entropy

MDPI

Article

A Variational Bayesian Deep Network with Data Self-Screening Layer for Massive Time-Series Data Forecasting

Xue-Bo Jin [1,2], Wen-Tao Gong [1,2], Jian-Lei Kong [1,2,*], Yu-Ting Bai [1,2] and Ting-Li Su [1,2]

[1] Artificial Intelligence College, Beijing Technology and Business University, Beijing 100048, China; jinxuebo@btbu.edu.cn (X.-B.J.); gongwentao@st.btbu.edu.cn (W.-T.G.); baiyuting@btbu.edu.cn (Y.-T.B.); sutingli@btbu.edu.cn (T.-L.S.)

[2] China Light Industry Key Laboratory of Industrial Internet and Big Data, Beijing Technology and Business University, Beijing 100048, China

* Correspondence: kongjianlei@btbu.edu.cn

Abstract: Compared with mechanism-based modeling methods, data-driven modeling based on big data has become a popular research field in recent years because of its applicability. However, it is not always better to have more data when building a forecasting model in practical areas. Due to the noise and conflict, redundancy, and inconsistency of big time-series data, the forecasting accuracy may reduce on the contrary. This paper proposes a deep network by selecting and understanding data to improve performance. Firstly, a data self-screening layer (DSSL) with a maximal information distance coefficient (MIDC) is designed to filter input data with high correlation and low redundancy; then, a variational Bayesian gated recurrent unit (VBGRU) is used to improve the anti-noise ability and robustness of the model. Beijing's air quality and meteorological data are conducted in a verification experiment of 24 h PM2.5 concentration forecasting, proving that the proposed model is superior to other models in accuracy.

Keywords: time-series data forecast; data self-screening layer; variational inference; gated recurrent unit; maximal information distance coefficient

check for updates

Citation: Jin, X.-B.; Gong, W.-T.; Kong, J.-L.; Bai, Y.-T.; Su, T.-L. A Variational Bayesian Deep Network with Data Self-Screening Layer for Massive Time-Series Data Forecasting. *Entropy* **2022**, *24*, 335. https://doi.org/10.3390/e24030335

Academic Editor: Markus Pauly

Received: 24 January 2022
Accepted: 23 February 2022
Published: 25 February 2022

1. Introduction

With the development of sensors and computer storage technology, people have effectively acquired and stored massive amounts of time-series data. The amount of data has exploded, and data often contain multiple related data variables. For example, to accurately predict the PM2.5 of air pollution, the PM2.5 data of different regions, other air quality data, such as PM10, O_3, SO_2, CO, and meteorological factors, such as temperature and humidity, etc., can be collected and used. In general, researchers think it will be more accurate and effective to forecast based on multiple related data variables, which has become one of artificial intelligence's current hot research directions. Machine learning and deep learning methods have been mainly used in time-series data forecasting. Among them, traditional machine learning such as support vector regression (SVR) [1], the integrated moving average autoregressive (ARIMA) model [2], linear regression [3], the Markov prediction method [4], etc., have been widely used. However, due to the insufficient performance of nonlinear fitting, they cannot yet model the highly complex and nonlinear data.

Due to the ability of deep learning networks, they have been widely used in many artificial intelligence fields such as image recognition [5–7], image classification [8–10], and time-series data forecasting. Recurrent neural networks (RNNs) [11], long short-term memory networks (LSTM) [12], gated recurrent units (GRUs) [13], etc., have become effective ways to solve time-series data forecasting. Improved models such as CNN-LSTM [14] and ConvLSTM [15] with convolution operations can extract high-level spatiotemporal features. Although big data can make the prediction model obtain more input data, it will lead to a large amount of redundancy, conflicts, and inconsistency. Therefore, big data does not mean good data, and blindly considering using big data cannot achieve high forecasting accuracy. Another problem for forecasting is that the data collected by sensors was inevitably polluted by noise, which will degrade the model's learning accuracy or even overfit the training. Therefore, it is essential to enhance the anti-noise ability of the model. The time-series models are based on statistical data, and the model parameters can be estimated through some identification methods [16–20], such as recursive algorithms [21–25] and hierarchical algorithms [26–30].

In applications, understanding and selecting data can effectively improve model training performance and reduce computational costs. Many methods have been used to measure data correlation, such as the Granger causality analysis method [31], mutual information method [32], Spearman rank correlation coefficient [33], Pearson correlation coefficient [34], etc. However, these methods cannot analyze the redundancy between data. Input data with high redundancy cannot improve the modeling and prediction performance, but it will cost more model training time and even decrease the prediction performance.

Aiming at the problem that the prediction accuracy of neural networks is reduced due to a large number of redundant, conflicting, inconsistent, and noisy input data, the innovations of proposed deep learning networks include the following:

(1) The prediction network is constructed with the data self-screening layer. A maximal information distance coefficient (MIDC) with Bayesian hyperparameter optimization is designed to mine the correlation and redundancy of input data simultaneously, effectively extracting useful input information for deep learning networks and eliminating redundancy.

(2) The variational inference structure is introduced into the gated recurrent unit (GRU) to achieve a Gaussian distribution for the networks' weights and biases, which can enhance the anti-noise ability of the network and effectively improve forecasting accuracy and generalization performance.

The rest of this article is organized as follows: Section 2 introduces related research work in this field, Section 3 presents the proposed method and prediction model in detail, Section 4 gives experiments with the analysis of results to verify the proposed method, and Section 5 discusses the conclusions.

The abbreviations used in this article are shown in Table 1.

Table 1. List of abbreviations.

Full Name	Abbreviation
Data self-screening layer	DSSL
Variational Bayesian gated recurrent unit	VBGRU
Maximal information distance coefficient	MIDC
Maximal information coefficient	MIC
Distance entropy	DE
Gaussian process regression	GPR
Kullback–Leibler	KL
Long short-term memory network	LSTM
Gated recurrent unit	GRU
Convolutional long short-term memory network	ConvLSTM
Convolutional neural network-long short-term memory network	CNN-LSTM
Time convolutional network	TCN
Root mean square error	RMSE
Mean square error	MSE
Mean absolute error	MAE

2. Related Work

Because traditional machine learning is challenging to learn and fit big data due to its simple structure, researchers often apply deep learning methods with strong information mining capabilities. For example, Teng Mengfan et al. [35] combined LSTM and CNN with a core size of 1 × 1 to predict PM2.5 based on the data from different locations. Zhao et al. [36] proposed a data-driven model called the long short-term memory-fully connected (LSTM-FC) neural network using historical air quality data, weather data, weather forecast data, and the day of the week to predict PM2.5 pollution at a specific air quality monitoring station within 48 h. Yeo et al. [37] gave a deep learning model that combines CNN and GRU to predict the PM2.5 concentration of 25 sites in Seoul, South Korea. They used all weather and air quality data observed between 2015 and 2017 to train the deep learning model. Ding et al. [38] proposed deep transfer metric learning for kernel regression (DTMLKR) to combine deep learning and transfer learning (TL) to solve the problem of regression prediction. YongShi et al. [39] designed a neural network (CNN) with different kernel sizes as a strategy network to predict stock price trends and stock transactions. Jin et al. [40] aimed to deal with existing methods' limitations with poor stability and unsatisfactory forecast accuracy to propose an attention-based Bayesian hyperparameter optimization network for accurate short-term load prediction.

Researchers have to design more complex networks in the research mentioned above according to the complexity of big input data. Compared with traditional machine learning methods, deep learning networks can capture time-series data information due to their high fitting capabilities. In contrast, for big data, even with deep learning networks, two following factors still lead to a decrease in the forecasting performance:

(1) The data redundancy, conflict, and inconsistency will reduce the learning effect and forecasting accuracy. Therefore, we cannot blindly use big data as the network's input data. It is necessary to analyze their relationship and select the correct data to improve model training performances.

(2) The noise and uncertainty introduced in the process of sensor measurement will cause the classical neural network to overfit during the training process, which will reduce the forecasting performance. The deep learning network operation mechanism must be reformed to make it applicable and robust to noise and improve the anti-noise ability of the network.

In recent years, existing studies have gradually begun considering relationships between input data for forecasting time-series data to improve accuracy. Abdourahamane et al. [41] combined wavelet transformation and the Frank Copula function and proposed a mutual-information-based nonlinear rainfall forecast model by evaluating the relationship between rainfall series. Peng et al. [42] presented a primary and secondary fuzzy cognitive map (PS-FCM) model to explore the causal relationship of haze pollution data. Han et al. [43] proposed a long short-term memory network based on correlation graph attention, which nests the correlation attention mechanism in the graph attention mechanism to strengthen spatiotemporal correlation. Jin et al. [44] proposed a distributed predictor that can overcome irrelevant data and sensor noise with causality coefficients (SCC) by selecting high causality measures as input data.

Although the above methods consider the selection between input data, these methods are all based on correlation analysis. It does not eliminate redundant information because it is often included in highly correlated data. Recently, variational inference has gradually been applied to the forecasting of the deep learning network of time-series data to improve the anti-noise ability of the network. For example, Zhou et al. [45] proposed a Bayesian framework of the variational graph cyclic attention neural network for robust traffic prediction. Similarly, a variational Bayesian network predicts solar radiation [46] and energy price [47]. These papers show that the variational Bayesian method can overcome the influence of uncertain data with noise, improving the prediction accuracy.

3. Data Self-Screening-Variational Bayesian GRU Model

This article proposes a maximal information distance coefficient considering the correlation and redundancy between multivariate data. Combined with Bayesian hyperparameter optimization, a self-screening layer with a self-learning optimization function based on different input data is constructed, which significantly improves the applicability of the prediction network. At the same time, we also built a Bayesian GRU deep prediction network combined with variational inference, which overcomes the difficulty of fitting noise in traditional deep learning and improves the model's prediction accuracy.

The framework of the proposed model with the data self-screening layer we designed is shown in Figure 1. The framework mainly includes two parts: data self-screening layer (DSSL) and variational Bayesian gated recurrent unit (VBGRU). The former has maximal information distance coefficient (MIDC) and Bayesian hyperparameter optimization, and the latter mainly combines variational inference and GRU.

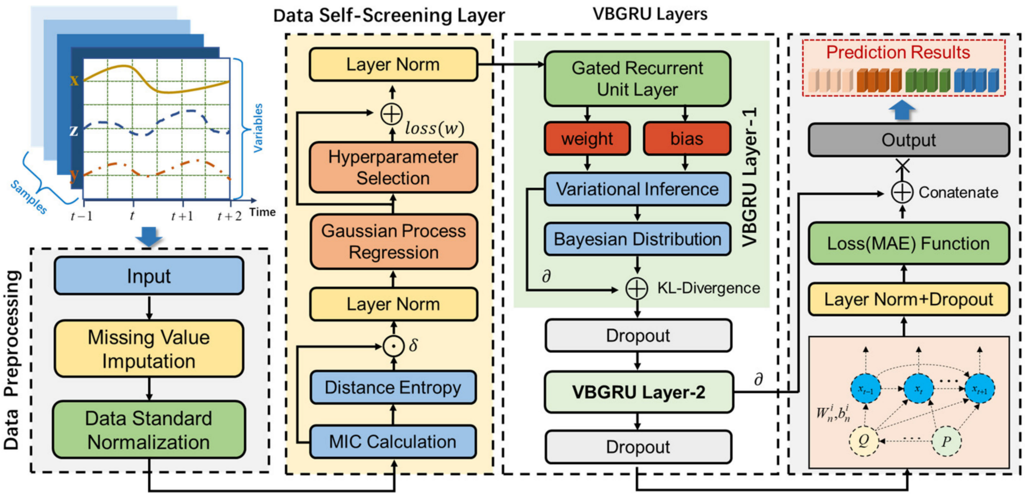

Figure 1. Deep Bayesian prediction network model framework with data self-screening layer.

The process of this forecasting framework is as the following:

(1) Collect time-series data of multidimensional variables and fill in missing values for the collected data.

(2) Input the processed time-series data into DSSL, which mainly screens variables with high correlation and low redundancy with the target variable and adaptively changes the relevant parameters of the data self-screening layer according to the different input data, then normalize the parameters by layer norm to enhance the suitability of the network.

(3) Input the variables selected by DSSL and target variables into the VBGRU network model for training; then, the dropout layer is used to randomly discard some neural network units to improve the robustness of the model, and finally obtain the prediction results of the target variables.

3.1. Data Self-Screening Layer

DSSL comprises two sections: MIDC and Bayesian hyperparameter optimization. MIDC consists of maximal information coefficient (MIC) and distance entropy (DE), and MIC screens the variables with high correlation but low redundancy with the target variable. Bayesian hyperparameter optimization adaptively learns the relevant parameters of MIDC according to the input data. The calculation flow chart of DSSL is shown in Figure 2.

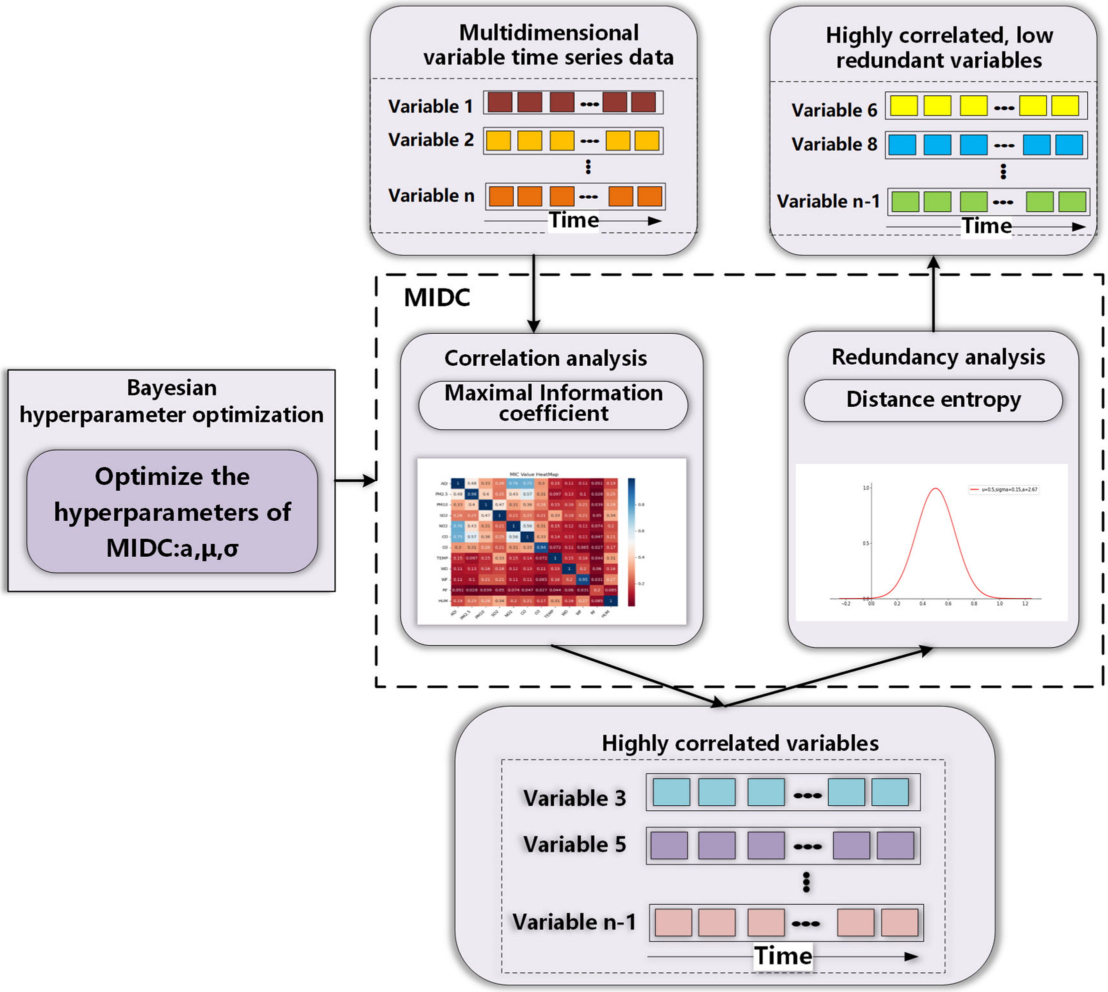

Figure 2. DSSL calculation flow chart.

As we know, the maximal information coefficient (MIC) has universality, fairness, and symmetry relative to sequence causality. The calculation of MIC about x and y is shown in Formula (1):

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} \tag{1}$$

However, highly correlated data will contain redundant information in many cases, which is not suitable for neural network training. To select the variables with high correlation but low redundancy, we propose MIDC as the following:

$$\delta(I(X;Y)) = \frac{\alpha}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(I(X;Y)-\mu)^2}{2\sigma^2}} \tag{2}$$

where α, σ, μ are the parameters. The distance coefficient diagram under different parameters is shown in Figure 3. We can see that the different parameters α, σ, μ can convert $I(X;Y)$ into different MIDC values. The Bayesian hyperparameter optimization method will be used to obtain the correct parameters.

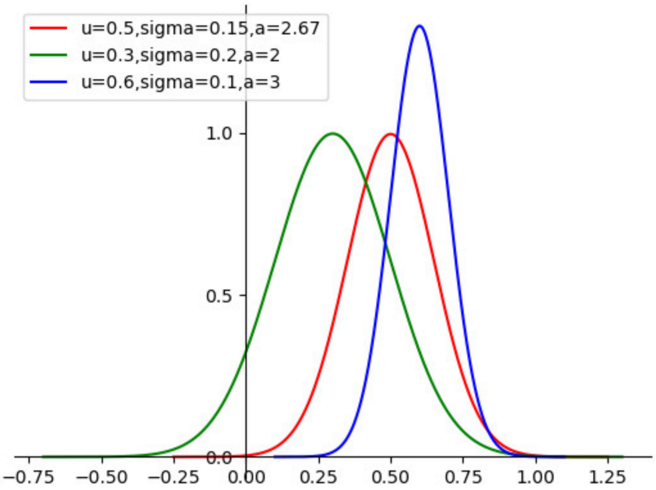

Figure 3. MIDC with different parameters.

The root mean square error is used as the objective function for optimizing hyperparameters:

$$loss(w) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \tag{3}$$

where $w = \alpha, \sigma, \mu$ is the hyperparameter that MIDC needs to optimize, T is the number of input samples, y_t is the actual value, $\hat{y}_t$ is the predicted value, and t is the time index.

The functional relationship between the hyperparameters and the loss function to be optimized solves the hyperparameter set that minimizes $loss(w)$. The process can be expressed as:

$$w* = \underset{w \in W}{\text{argmin}}\, loss(w) \tag{4}$$

where $w*$ is the optimal parameter determined by Bayesian hyperparameter optimization, w is a set of input hyperparameters, and W is the parameter space of multidimensional hyperparameters.

Bayesian hyperparameter optimization is divided into two steps: Gaussian process regression (GPR) and hyperparameter selection [48]. The pseudocode of Bayesian optimization is shown in Algorithm 1.

Algorithm 1: Bayesian Hyperparameter Optimization

```
Input:
    Initial observation set $D_n = \{x_n, y_n\}$
    Bounds for the search space $\chi$
Output: $\{x_n, y_n\}_{n=1}^{t}$
for $n = 1, 2, \ldots,$ t do
    Fit the current data sample $D_n$ to get the GPR model $G(w)_n$
    Solve the extreme points of the objective function $loss(w)$:$w_{n+1} = \underset{w \in W}{\text{argmin}} loss(w, G(w)_n)$
    Obtain new samples $(w_{n+1}, loss(w)_{n+1})$
    Update data sample $D_{n+1} = \{D_n(w_{n+1}, loss(w)_{n+1})\}$.
    Update data self-screening layer parameters
end for
```

3.2. Variational Bayesian GRU

The training process of the deep network model is the process of optimizing the network weights through the sample and target data pairs and finally reaching the desired indicators and obtaining the optimal model weights. Researchers have found that the performance of the deterministic deep network model is affected by the training data itself, resulting in its limited generalization ability. At the same time, the noise in the data source will also affect the network's performance. For example, when the model is fully trained, it may cause overfitting the noise, resulting in a decrease in prediction accuracy. To solve this problem, researchers have proposed a weight calculation method based on variational inference and applied this method to the training process of deep learning networks to realize the learning of the feature distribution of the original data. The primary theoretical basis of this method is to change the traditional fixed-weight neural network to a distributed-weight neural network. When predicting, the weight distribution is sampled to obtain the prediction result. The network structure of VBGRU is shown in Figure 4.

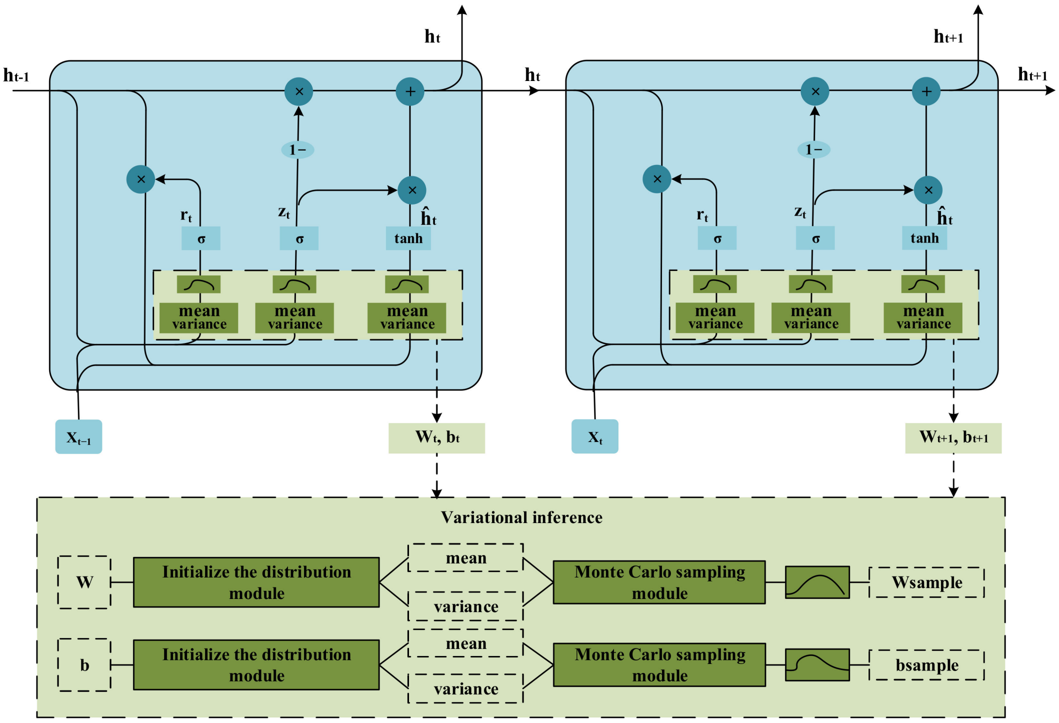

Figure 4. VBGRU structure.

For deep deterministic networks, the trainable parameters of each network layer correspond to linear or nonlinear changes in input and output, as shown in Formula (5).

$$y^{(i+1)} = W^{(i+1)} \cdot y^{(i)} + b^{(i+1)} \tag{5}$$

Among them $y^{(i)}$ represents the activation output corresponding to the i-th layer. In order to realize the flow of information between layers, the weight $W^{(i)}$ and bias $b^{(i)}$ of each layer of the trained deep network are composed of specific values. Therefore, a deterministic result will be obtained when using new samples for network testing.

Different from the deterministic network, VBGRU selects the weights by sampling from the parameter distribution of the trainable variables calculated in each feedforward calculation, thereby introducing the uncertainty of the weights. As shown in Figure 4, the weights and biases of VBGRU are converted into distribution by the variational inference method. The specific process is: First, initialize the weight W and bias b of the model by initializing the distribution module to obtain the corresponding mean and variance; Then, use the Monte Carlo sampling module to sample the new weight W_{sample} and bias b_{sample} from their corresponding mean and variance. This enables the Bayesian layer to optimize the performance indicators of the model but also learns the uncertainty of network predictions on specific data points. In Figure 5, we show the schematic diagrams of the weights of the deep Bayesian network and the deep deterministic network. VBGRU can obtain multiple model outputs by adding sampling points to calculate the uncertainty of the model at a specific point.

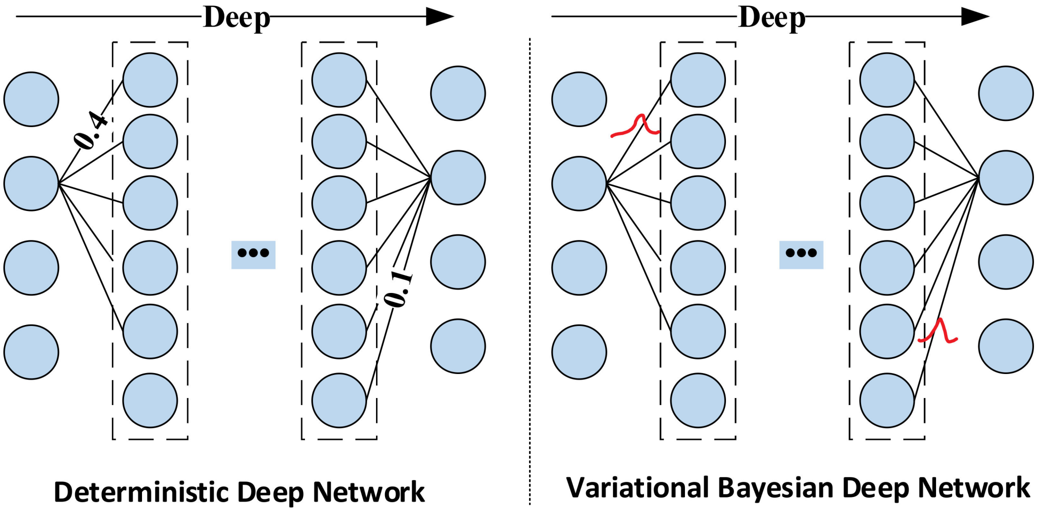

Figure 5. Schematic diagram of the weights of the variational Bayesian network.

Let $W_{(n)}^{(i)}$ denote the n-th sampling weight of the i-th layer and $b_{(n)}^{(i)}$ distinguish the bias. In the deep Bayesian network, the weight and bias are not specific numbers but the result obtained by sampling on a distribution. Like the deep deterministic network, the deep Bayesian network is also based on the sample and target data pairs and the weight $W_{(n)}^{(i)}$ and bias $b_{(n)}^{(i)}$ obtained through training. The difference is that training conveys is not their definite values but their distribution parameters $\rho^{(i)}$ $\nu^{(i)}$. The relationship between them and the weight $W_{(n)}^{(i)}$ and bias $b_{(n)}^{(i)}$ are shown in formulas (6) and (7).

$$W_{(n)}^{(i)} = N(0,1) * \log(1+\rho^{(i)}) + \nu^{(i)} \tag{6}$$

$$b_{(n)}^{(i)} = N(0,1) * \log(1+\rho^{(i)}) + \nu^{(i)} \tag{7}$$

Generally, a loss function that can be differentiated is required in a deep learning model, and the average absolute error (MAE) is usually used as the loss function. Additionally, the deep Bayesian network's optimization goal needs to be determined by the loss function. The deep Bayesian network calculates the Kullback–Leibler (KL) divergence between a complex and simple distribution. The loss function is defined as follows:

Let $P(\omega)$ be a manually set low-entropy prior distribution,$Q(\omega|\theta)$ be the posterior distribution of a given parameter. For each scalar weight W obtained for sampling, the KL divergence of the low-entropy prior distribution $P(\omega)$ and the posterior distribution $Q(\omega|\theta)$ follows the formula (8).

$$D_{KL}(Q(\omega|\theta)||P(\omega)) = \lim_{n\to\infty} 1/n \sum_{i=0}^{n} Q\left(\omega^{(i)}\middle|\theta\right) * \left(\log Q\left(\omega^{(i)}\middle|\theta\right) - \log P\left(\omega^{(i)}\right)\right) \tag{8}$$

When n is infinite, we have:

$$D_{KL}(Q(\omega|\theta)||P(\omega)) = 1/n \sum_{i=0}^{n} Q\left(\omega^{(i)}\middle|\theta\right) * \left(\log Q\left(\omega^{(i)}\middle|\theta\right) - \log P\left(\omega^{(i)}\right)\right) \tag{9}$$

Since the value of a given parameter $Q(\omega|\theta)$ can be calculated directly, the KL divergence is calculated by the formula (10).

$$D_{KL}(Q(\omega|\theta)||P(\omega)) = \nu_Q * \sum_{i=0}^{n}\left(\log Q\left(\omega^{(i)}\middle|\theta\right) - \log P\left(\omega^{(i)}\right)\right) \tag{10}$$

where ν_Q is $\frac{1}{n}\sum\limits_{i=0}^{n} Q\left(\omega^{(i)}\middle|\theta\right)$, represents the mean value of the posterior distribution, which can be obtained by sampling the posterior distribution multiple times and averaging. Without involving error calculation, it can be excluded from the loss function, and the final loss of each sample can be expressed in the form of Formula (11).

$$Loss = \log\left(Q\left(\omega^{(n)}\middle|\theta\right)\right) - \log\left(P\left(\omega^{(n)}\right)\right) \tag{11}$$

In addition, using the KL divergence as the loss error between the prediction result and the network output is not enough because this can only learn the distribution characteristics of the data. Therefore, when training Bayesian GRU, it is usually necessary to combine MAE error to form a connection error, as shown in the formula (12). Among them ∂ represents an error weight parameter, which is generally set as the reciprocal of the number N of all training samples, that is, $\partial = 1/N$.

$$Loss = Loss_{MAE} + \partial \cdot \left[\log\left(Q\left(\omega^{(n)}\middle|\theta\right)\right) - \log\left(P\left(\omega^{(n)}\right)\right)\right] \tag{12}$$

Through the above analysis, we initialize all weights and biases in the GRU to a standard normal distribution and update the weight parameters of the network model through the Adam optimizer to obtain the best network parameters. The optimal mean and variance of the weight distribution and bias distribution are obtained. Using the trained model, the weight distribution and bias distribution are sampled multiple times by sampling, and multiple sets of prediction results are obtained. Finally, the multiple sets of prediction results are averaged, which is the predicted value output by the network. Based on the above analysis, the iterative calculation process of the VBGRU model is shown below:

(1) VBGRU initialization: set initialization weight distribution w_{random}= N$(0, 0.1)$, bias distribution b_{random}= N$(0, 0.01)$. Setting parameters $\theta = (\rho, \nu)$, The weight w_{sample}, and bias b_{sample} after sampling are obtained by Monte Carlo sampling;

(2) Given a total of m samples for each batch: $D(X, Y)$. Among them X represents the network input data, Y represents the expected output of the network, the network output is $\hat{Y}$;

(3) Use variational inference to sample the network weights and biases n times and calculate the average loss:

$$Loss = \frac{1}{n}\left\{Loss_{MAE} + \partial\cdot\left[\log\left(Q\left(\omega^{(n)}\middle|\theta\right)\right) - \log\left(P\left(\omega^{(n)}\right)\right)\right]\right\} \tag{13}$$

(4) Use Adam optimizer according to *Loss* to update the weight and bias parameters: $\theta(\rho, \nu)$;

(5) Repeat the second to fourth steps of network convergence, that is, *Loss* no longer drops;

(6) Use the test set to evaluate the trained network model.

4. Experiment and Analysis

4.1. Data Set Description and Preprocessing

This article uses PM2.5 in the air quality data of Guanyuan in Xicheng District, Beijing, from 1 January 2017, to 31 December 2021, as the target variable. The data sets are the air quality data of neighboring areas and the meteorological data of Haidian, the adjacent area. We performed two types of preprocessing operations: missing value padding and normalization. Each data set contains 43,760 data points, with 90% for the training and 10% for the test. The sampling frequency of all data is 1 h. The air quality data in this data set has a strong temporal and spatial correlation, as shown in Figure 6.

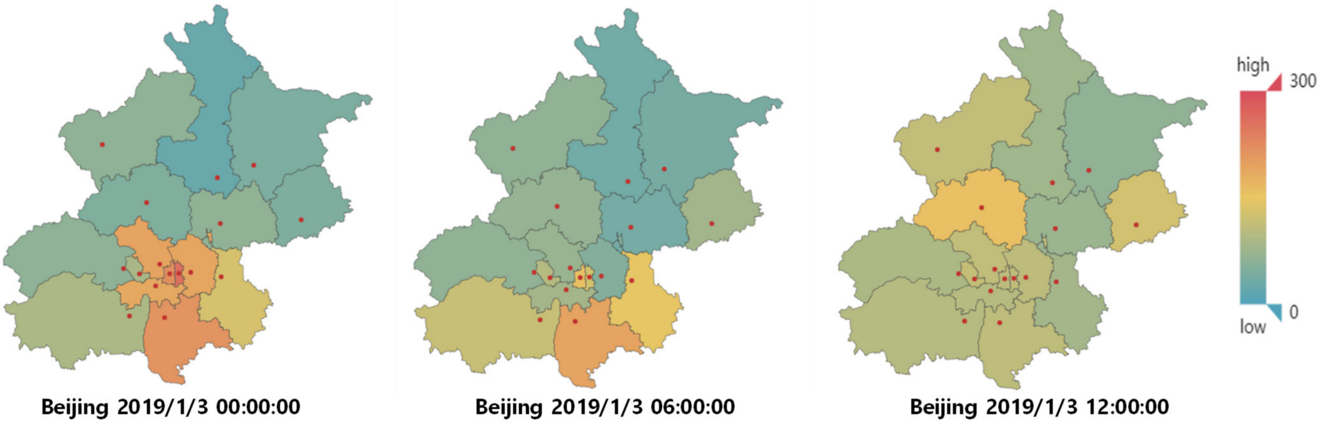
Beijing 2019/1/3 00:00:00
Beijing 2019/1/3 06:00:00
Beijing 2019/1/3 12:00:00
high
300
0
low

Figure 6. Distribution of PM2.5 in Beijing.

It can be seen that PM2.5 has changed little within six-hour. The reason is that PM2.5 particles have a dissipation process under climatic conditions. In addition, there is a similar process in neighboring areas, and the PM2.5 in heavy industrial development zones is relatively high.

Figure 7 shows the solid spatial correlation in the air quality data. The red area represents Yongding Gate, and the green area represents West Wanshou Palace. It can be seen that these two points are very close. The right figure shows the PM2.5 in these two areas at 480 samplings. The high degree of coincidence proves a redundant relationship between them.

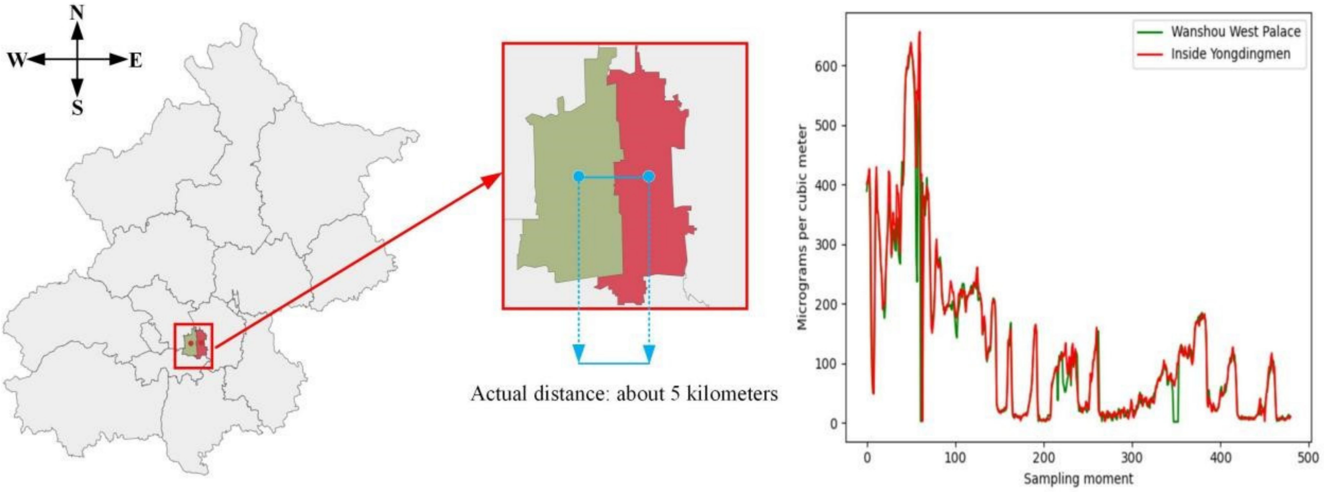

Figure 7. PM2.5 changes in Beijing Wanshou West Palace and Yongdingmen Observatory.

4.2. *Experiment Establishment and Evaluation Function*

Based on our data set, the following two experiments are designed:

(1) DSSL mainly consists of MIDC and Bayesian hyperparameter optimization. MIDC is an essential part of DSSL to analyze and quantify the relationship between different variables. Therefore, the first experiment used MIDC to quantitatively analyze the relationship between the target variable and other air quality factors, select variables with high correlation and low redundancy, and then superimpose these variables in

turn and input them into VBGRU to verify the impact of correlation and redundancy between data on prediction performance (see Section 4.3 for details).

(2) Based on the variables filtered out using DSSL, we compare the prediction performance of VBGRU with LSTM, GRU, convolutional long short-term memory network (ConvLSTM), convolutional neural network-long short-term memory network (CNN-LSTM), and time convolutional network (TCN), and evaluate the pros and cons of the model's predictive ability through the evaluation function (see Section 4.4 for details).

Our experiment used the open-source deep learning library Pytorch to build a deep learning network model. Specifically, our prediction model consists of the DSSL layer, VBGRU layer, and linear layer, and the implicit neuron size of the prediction model is set to 24. The prediction steps are 24. Twenty-four air quality data and weather data on day i are used as input objects, and 24 PM2.5 concentration data on day (i + 1) are used as expected values. The model is made to perform supervised learning using the Adam optimization algorithm. The learning rate of the Adam optimization algorithm was set to 0.001, and 100 epochs were trained.

Our experiments are conducted on a desktop computer equipped with an AMD R7-5800 processor, 4.0 GHz, and 16GB of RAM. At the same time, we use three evaluation functions to evaluate the prediction performance of the model, namely: root mean square error (RMSE), mean square error (MSE), and mean absolute error (MAE). They are calculated by Formulas (14)–(16), respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{15}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(\hat{y}_i - y_i)^2| \tag{16}$$

where n represents the total number of samples in the data set, y_i representing the actual value of PM2.5, $\hat{y}_i$ representing the prediction of PM2.5 obtained through experiments. The smaller RMSE, MSE, and MAE, the better the model's prediction performance.

4.3. Performance Verification of MIDC

We first analyzed the correlation between the other six air quality factors, such as AQI, CO, NO_2, O_3, PM10, SO_2, as well as the predicted PM2.5 in the Guanyuan area. The thermodynamic diagram of each variable MIC is shown in Figure 8.

It can be seen that the MICs of PM2.5 and AQI and CO are 0.75 and 0.57, respectively. However, the MIDCs are 0.26 and 0.91, respectively (as shown in Table 2). Table 2 also gives the prediction results of Guanyuan PM2.5 combined with AQI or CO, respectively. It can be seen that when using PM2.5 and AQI as input data, the RMSE is 28.87, and the training time is 48.44 s, while when using PM2.5 and CO as input data, the RMSE is 28.66, and the training time is reduced to 44.81 s.

Table 2. Forecast results of PM2.5 in the Guanyuan combined with air quality factors.

The Input Data	MIC	MIDC	RMSE	MSE	MAE	Train_Time
PM2.5, AQI	0.76	0.26	28.87	833.48	20.29	48.44 s
PM2.5, CO	0.57	0.91	28.66	821.18	20.02	44.81 s

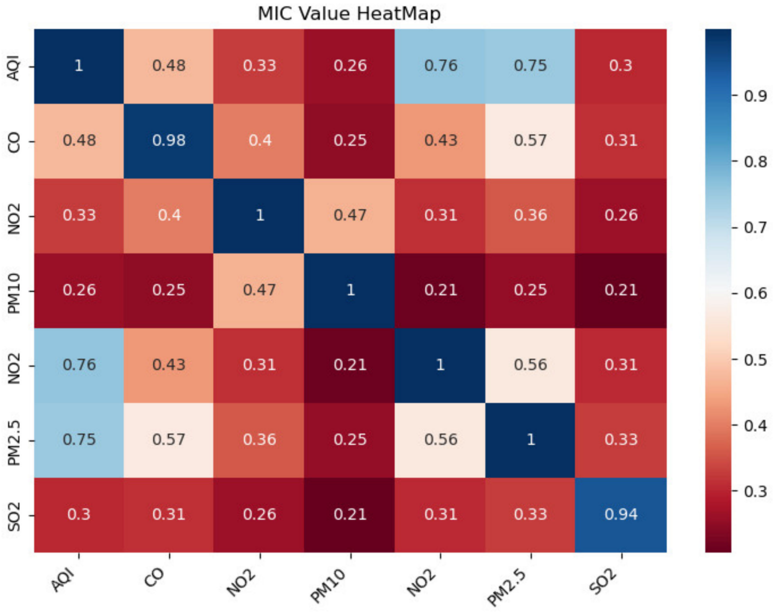

Figure 8. MIC results between different air quality variables in the Guanyuan area.

This result verifies that MIC cannot be used to select input data. The highest MIC indicates high redundancy between AQI and PM2.5, which increases the number of useless computations on the network and reduces the network convergence speed. On the contrary, the proposed MIDC shows a high correlation and can exclude high redundancy.

4.4. Compared with Other Models

In this section, we want to compare the performance of the VBGRU model with models such as LSTM [12], GRU [13], CNN-LSTM [14], ConvLSTM [15], and TCN [49] in predicting the hourly PM2.5 concentration in the next 24 h. To better compare the prediction performance of each model, we consider the use of cross-validation methods for performance validation. The commonly used cross-validation methods in machine learning are Monte Carlo simulation [50] and K-fold cross-validation [51]. However, the Monte Carlo simulation method, in which all data samples are randomly sampled after defining the sizes of the training and test machine sets, is not suitable for predicting temporal data with backward and forward dependencies. K-fold cross-validation, also known as rolling cross-validation, is a method that splits the temporal data set into training and test sets according to the temporal order. The results of the K-fold cross-validation runs are averaging can yield an (almost) unbiased estimate of the algorithm performance, avoiding prediction results obtained by chance based on a single iteration for each model, and ensuring accurate comparison of prediction methods.

We use 10-fold cross-validation, which divides the dataset into 10 parts in chronological order, and use the first 9 parts of the data in turn as training data in chronological order and the last 1 part of the training dataset as test data to obtain the prediction results of each dataset in turn and carry out the average to obtain the final prediction results. The prediction results of each model are shown in Figures 9 and 10, respectively.

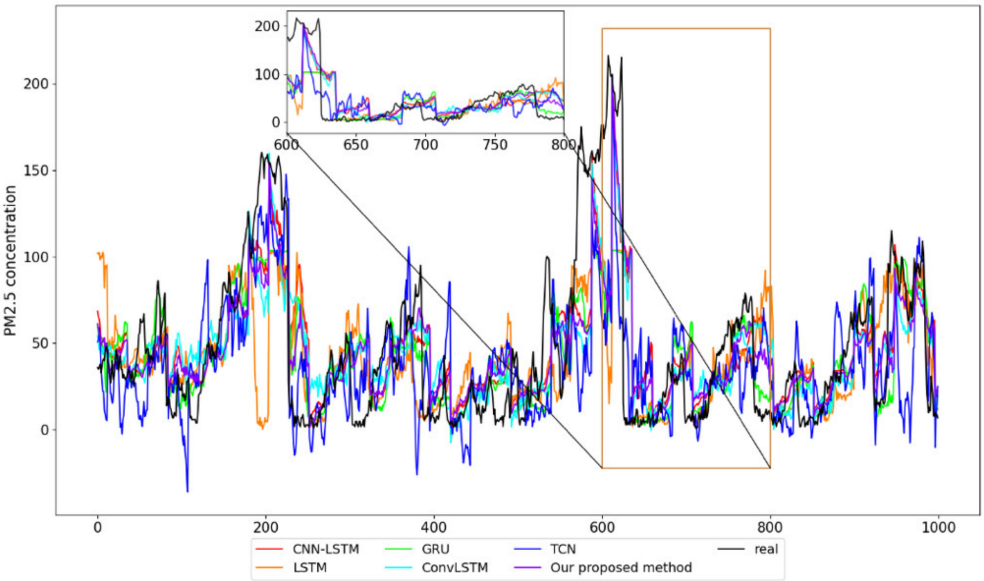

Figure 9. Part of the prediction results of each model.

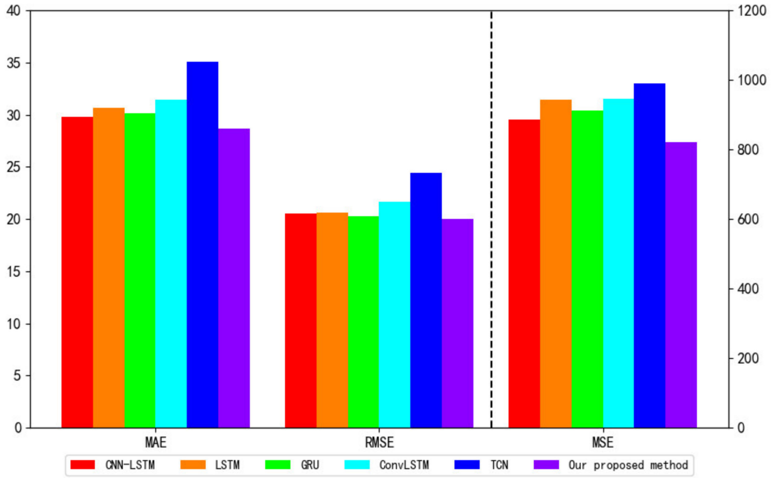

Figure 10. MAE, RMSE, and MSE of different models.

It can be seen from Table 3 that the RMSE, MSE, and MAE of the proposed VBGRU model are 28.59, 817.12, and 19.78, respectively.

Table 3. Evaluation results of different models on the same data set.

Models	RMSE	MSE	MAE	Train_Time
CNN-LSTM [14]	29.76	886.45	20.51	69.11 s
LSTM [12]	30.66	942.24	20.60	36.73 s
GRU [13]	30.13	911.26	20.28	39.97 s
ConvLSTM [15]	31.45	990.17	21.61	78.26 s
TCN [49]	35.05	1233.42	24.43	119.54 s
Our proposed VBGRU	28.59	817.12	19.78	44.81 s

Compared with the prediction performance of the other five models, the RMSE of VBGRU decreased by 3.9%, 6.8%, 5.1%, 9.2%, and 18.4% compared to CNN-LSTM, LSTM, GRU, ConvLSTM and TCN, respectively; the MSE decreased by 7.8%, 14.9%, 10.3%, 17.5% and 33.8% respectively; and the MAE of CNN-LSTM, LSTM, GRU, ConvLSTM and TCN decreased by 3.6%, 4.0%, 2.5%, 8.5% and 19.0%, respectively. The method we propose has the smallest prediction error, the best fit to the true value, and the smallest deviation from the true value.

In addition, the training speed of our proposed VBGRU model is 47.71, and the training times of CNN-LSTM, ConvLSTM, and TCN are 69.11, 78.26, and 119.54, respectively, which are much higher than the training speed of the VBGRU model. The training time of GRU and LSTM is 36.73 and 39.97, which is faster than our proposed model, but the prediction accuracy of the two is far lower than our proposed model.

To fully demonstrate and compare the prediction performance of each model, we plotted the RMSE of the 10-time cross-validation results of each model into a violin diagram to comprehensively compare the prediction accuracy and robustness of the models. The statistical results are shown in Figure 11.

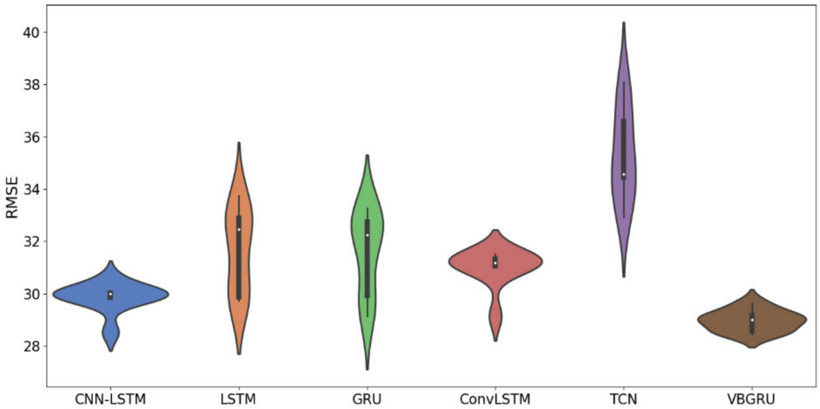

Figure 11. Violin plot of 10 cross-validation results for different models.

As shown in Figure 11, our proposed VBGRU has the smallest prediction error range, the most uniform and concentrated distribution, and the smallest average prediction error of the model. It has the highest prediction accuracy and the best stability compared with other models.

Based on the above data analysis, we can conclude that the proposed VBGRU model can achieve better performance at a little computational cost.

5. Conclusions and Future Work

Aiming at the problem that a large amount of noise and data conflict, redundancy, or inconsistency reduces the prediction accuracy, this paper proposes a variational Bayesian deep prediction network with a self-screening layer. The model used the self-screening layer to high mine correlation and low redundancy between multiple time-series input variables, reducing unnecessary input information of the model. It gives full play to the powerful feature extraction and anti-noise capabilities of the variational Bayesian GRU for modeling time-series data. It improves the prediction accuracy and robustness effectively.

The prediction and verification experiment of Beijing air quality data, considering the indicators such as RMSE, MSE, and MAE, shows that the model is superior to other models in terms of prediction accuracy and calculation speed. The proposed prediction approaches of time-series models in the paper can combine other parameter estimation algorithms [52–58] with studying the parameter identification problems of linear and nonlinear systems with different disturbances [59–64] to build soft sensor models and prediction models based on time-series data that can be applied to other fields [65–70] such as signal processing and engineering application systems [71–78].

Author Contributions: Conceptualization, X.-B.J. and W.-T.G.; methodology, X.-B.J. and W.-T.G.; software, W.-T.G.; validation, X.-B.J. and W.-T.G.; formal analysis, J.-L.K.; investigation, Y.-T.B. and T.-L.S.; resources, X.-B.J. and J.-L.K.; data curation, W.-T.G., Y.-T.B. and T.-L.S.; writing—original draft preparation, X.-B.J. and W.-T.G.; writing—review and editing, X.-B.J. and W.-T.G.; visualization, W.-T.G.; supervision, J.-L.K., Y.-T.B. and T.-L.S.; project administration, X.-B.J. and J.-L.K.; funding acquisition, X.-B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China No., 62006008, 62173007, 61903009.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, W.; Liang, S.; Yu, Q. PM2.5 concentration prediction based on pseudo-F statistic feature selection algorithm and support vector regression. In *Earth and Environmental Science, Proceedings of the Third International Workshop on Environment and Geoscience, Chengdu, China, 18–20 July 2020*; IOP Conference Series; IOP Publishing: Bristol, UK, 2020; Volume 569, p. 012039. [CrossRef]
2. Guo, N.; Chen, W.; Wang, M.; Tian, Z.; Jin, H. Appling an Improved Method Based on ARIMA Model to Predict the Short-Term Electricity Consumption Transmitted by the Internet of Things (IoT). *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6610273. [CrossRef]
3. Cholianawati, N.; Cahyono, W.E.; Indrawati, A.; Indrajad, A. Linear Regression Model for Predicting Daily PM2. 5 Using VIIRS-SNPP and MODIS-Aqua AOT. In *Earth and Environmental Science, Proceedings of the International Conference On Tropical Meteorology And Atmospheric Sciences, Bandung, Indonesia, 19–20 September 2018*; IOP Conference Series; IOP Publishing: Bristol, UK, 2019; Volume 303, p. 012039. [CrossRef]
4. Sun, W.; Zhang, H.; Palazoglu, A.; Singh, A.; Zhang, W.; Liu, S. Prediction of 24-hour-average PM2.5 concentrations using a hidden Markov model with different emission distributions in Northern California. *Sci. Total Environ.* **2013**, *443*, 93–103. [CrossRef] [PubMed]
5. Kong, J.L.; Wang, H.; Wang, X.; Jin, X.B.; Fang, X.; Lin, S. Multi-stream hybrid architecture based on cross-level fusion strategy for fine-grained crop species recognition in precision agriculture. *Comput. Electron. Agric.* **2021**, *185*, 106134. [CrossRef]
6. Kong, J.; Yang, C.; Wang, J.; Wang, X.; Zuo, M.; Jin, X.; Lin, S. Deep-stacking network approach by multisource data mining for hazardous risk iden-tification in IoT-based intelligent food management systems. *Comput. Intell. Neurosci.* **2021**, *2021*, 1194565. [CrossRef]
7. Zhen, T.; Kong, J.L.; Yan, L. Hybrid Deep-Learning Framework Based on Gaussian Fusion of Multiple Spatiotemporal Networks for Walking Gait Phase Recognition. *Complexity* **2020**, *2020*, 8672431. [CrossRef]
8. Zheng, Y.-Y.; Kong, J.-L.; Jin, X.-B.; Wang, X.-Y.; Su, T.-L.; Zuo, M. CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture. *Sensors* **2019**, *19*, 1058. [CrossRef]
9. Zheng, Y.Y.; Kong, J.L.; Jin, X.B.; Wang, X.-Y.; Su, T.-L.; Wang, J.-L. Probability Fusion Decision Framework of Multiple Deep Neural Networks for Fine-grained Visual Classification. *IEEE Access* **2019**, *7*, 122740–122757. [CrossRef]
10. Kong, J.L.; Wang, Z.N.; Ji, X.B.; Wang, X.-Y.; Su, T.-L.; Wang, J.-L. Semi-supervised segmentation framework based on spot-divergence super voxelization of multi-sensor fusion data for autonomous forest machine applications. *Sensors* **2018**, *18*, 3061. [CrossRef]
11. Connor, J.T.; Martin, R.D.; Atlas, L.E. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **1994**, *5*, 240–254. [CrossRef]
12. Qadeer, K.; Rehman, W.U.; Sheri, A.M.; Park, I.; Kim, H.K.; Jeon, M. A long short-term memory (LSTM) network for hourly estimation of PM2. 5 con-centration in two cities of South Korea. *Appl. Sci.* **2020**, *10*, 3984. [CrossRef]
13. Becerra-Rico, J.; Aceves-Fernández, M.A.; Esquivel-Escalante, K.; Pedraza-Ortega, J.C. Airborne particle pollution predictive model using Gated Recurrent Unit (GRU) deep neural networks. *Earth Sci. Inform.* **2020**, *13*, 821–834. [CrossRef]
14. Li, W.; Gao, X.; Hao, Z.; Sun, R. Using deep learning for precipitation forecasting based on spatio-temporal information: A case study. *Clim. Dyn.* **2021**, *58*, 443–457. [CrossRef]
15. Wang, W.; Mao, W.; Tong, X.; Xu, G. A Novel Recursive Model Based on a Convolutional Long Short-Term Memory Neural Network for Air Pollution Prediction. *Remote Sens.* **2021**, *13*, 1284. [CrossRef]
16. Ding, F.; Chen, T. Combined parameter and output estimation of dual-rate systems using an auxiliary model. *Automatica* **2004**, *40*, 1739–1748. [CrossRef]
17. Ding, F.; Chen, T. Parameter estimation of dual-rate stochastic systems by using an output error method. *IEEE Trans. Autom. Control* **2005**, *50*, 1436–1441. [CrossRef]
18. Ding, F.; Shi, Y.; Chen, T. Auxiliary model-based least-squares identification methods for Hammerstein output-error systems. *Syst. Control Lett.* **2007**, *56*, 373–380. [CrossRef]
19. Zhou, Y.; Ding, F. Modeling Nonlinear Processes Using the Radial Basis Function-Based State-Dependent Autoregressive Models. *IEEE Signal Process. Lett.* **2020**, *27*, 1600–1604. [CrossRef]
20. Zhou, Y.H.; Zhang, X.; Ding, F. Partially-coupled nonlinear parameter optimization algorithm for a class of multivariate hybrid models. *Appl. Math. Comput.* **2022**, *414*, 126663. [CrossRef]
21. Zhou, Y.H.; Zhang, X.; Ding, F. Hierarchical Estimation Approach for RBF-AR Models with Regression Weights Based on the Increasing Data Length. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3597–3601. [CrossRef]

22. Ding, F.; Zhang, X.; Xu, L. The innovation algorithms for multivariable state-space models. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1601–1618. [CrossRef]
23. Ding, F.; Liu, G.; Liu, X.P. Parameter estimation with scarce measurements. *Automatica* **2011**, *47*, 1646–1655. [CrossRef]
24. Liu, Y.J.; Ding, F.; Shi, Y. An efficient hierarchical identification method for general dual-rate sampled-data systems. *Automatica* **2014**, *50*, 962–970. [CrossRef]
25. Zhang, X.; Ding, F. Optimal Adaptive Filtering Algorithm by Using the Fractional-Order Derivative. *IEEE Signal Process. Lett.* **2022**, *29*, 399–403. [CrossRef]
26. Li, M.H.; Liu, X.M.; Ding, F. The filtering-based maximum likelihood iterative estimation algorithms for a special class of nonlinear systems with autoregressive moving average noise using the hierarchical identification principle. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1189–1211. [CrossRef]
27. Ding, J.; Ding, F.; Liu, X.P.; Liu, G. Hierarchical Least Squares Identification for Linear SISO Systems with Dual-Rate Sampled-Data. *IEEE Trans. Autom. Control* **2011**, *56*, 2677–2683. [CrossRef]
28. Ding, F.; Liu, Y.; Bao, B. Gradient-based and least-squares-based iterative estimation algorithms for multi-input multi-output systems. *Proc. Inst. Mech. Eng. Part I J. Syst. Control. Eng.* **2012**, *226*, 43–55. [CrossRef]
29. Xu, L.; Chen, F.Y.; Hayat, T. Hierarchical recursive signal modeling for multi-frequency signals based on discrete measured data. *Int. J. Adapt. Control Signal Process.* **2021**, *35*, 676–693. [CrossRef]
30. Wang, Y.; Ding, F. Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model. *Automatica* **2016**, *71*, 308–313. [CrossRef]
31. Zhang, K.; Zheng, L.; Liu, Z.; Jia, N. A deep learning based multitask model for network-wide traffic speed prediction. *Neurocomputing* **2020**, *396*, 438–450. [CrossRef]
32. Wang, G.; Awad, O.I.; Liu, S.; Shuai, S.; Wang, Z. NOx emissions prediction based on mutual information and back propagation neural network using correlation quantitative analysis. *Energy* **2020**, *198*, 117286. [CrossRef]
33. Song, H.Y.; Park, S. An Analysis of Correlation between Personality and Visiting Place using Spearman's Rank Correlation Coefficient. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 1951–1966. [CrossRef]
34. Liu, Y.; Mu, Y.; Chen, K.; Li, Y.; Guo, J. Daily Activity Feature Selection in Smart Homes Based on Pearson Correlation Coefficient. *Neural Process. Lett.* **2020**, *51*, 1771–1787. [CrossRef]
35. Mengfan, T.; Siwei, L.; Ge, S.; Jie, Y.; Lechao, D.; Hao, L.; Senlin, H. Including the feature of appropriate adjacent sites improves the PM2.5 concentration prediction with long short-term memory neural network model. *Sustain. Cities Soc.* **2021**, *76*, 103427. [CrossRef]
36. Zhao, J.; Deng, F.; Cai, Y.; Chen, J. Long short-term memory-Fully connected (LSTM-FC) neural network for PM2. 5 concentration prediction. *Chemosphere* **2019**, *220*, 486–492. [CrossRef]
37. Yeo, I.; Choi, Y.; Lops, Y.; Sayeed, A. Efficient PM2.5 forecasting using geographical correlation based on integrated deep learning algorithms. *Neural Comput. Appl.* **2021**, *33*, 15073–15089. [CrossRef]
38. Ding, Y.; Jia, M.; Miao, Q.; Huang, P. Remaining useful life estimation using deep metric transfer learning for kernel regression. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107583. [CrossRef]
39. Shi, Y.; Li, W.; Zhu, L.; Guo, K.; Cambria, E. Stock trading rule discovery with double deep Q-network. *Appl. Soft Comput.* **2021**, *107*, 107320. [CrossRef]
40. Jin, X.-B.; Zheng, W.-Z.; Kong, J.-L.; Wang, X.-Y.; Bai, Y.-T.; Su, T.-L.; Lin, S. Deep-Learning Forecasting Method for Electric Power Load via Attention-Based Encoder-Decoder with Bayesian Optimization. *Energies* **2021**, *14*, 1596. [CrossRef]
41. Abdourahamane, Z.S.; Acar, R.; Serkan, Ş. Wavelet–copula-based mutual information for rainfall forecasting applications. *Hydrol. Process.* **2019**, *33*, 1127–1142. [CrossRef]
42. Peng, Z.; Liu, W.Q.; An, S.J. Haze pollution causality mining and prediction based on multi-dimensional time series with PS-FCM. *Inf. Sci.* **2020**, *523*, 307–317. [CrossRef]
43. Han, S.; Dong, H.; Teng, X.; Li, X.; Wang, X. Correlational graph attention-based Long Short-Term Memory network for multivariate time series prediction. *Appl. Soft Comput.* **2021**, *106*, 107377. [CrossRef]
44. Jin, X.-B.; Yu, X.-H.; Su, T.-L.; Yang, D.-N.; Bai, Y.-T.; Kong, J.-L.; Wang, L. Distributed Deep Fusion Predictor for aMulti-Sensor System Based on Causality Entropy. *Entropy* **2021**, *23*, 219. [CrossRef]
45. Zhou, F.; Yang, Q.; Zhong, T.; Chen, D.; Zhang, N. Variational Graph Neural Networks for Road Traffic Prediction in Intelligent Transportation Systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2802–2812. [CrossRef]
46. Liu, Y.; Qin, H.; Zhang, Z.; Pei, S.; Wang, C.; Yu, X.; Jiang, Z.; Zhou, J. Ensemble spatiotemporal forecasting of solar irradiation using variational Bayesian convolutional gate recurrent unit network. *Appl. Energy* **2019**, *253*, 113596. [CrossRef]
47. Brusaferri, A.; Matteucci, M.; Portolani, P.; Vitali, A. Bayesian deep learning based method for probabilistic forecast of day-ahead electricity prices. *Appl. Energy* **2019**, *250*, 1158–1175. [CrossRef]
48. Jin, X.B.; Wang, H.X.; Wang, X.Y.; Bai, Y.-T.; Su, T.-L.; Kong, J.-L. Deep-learning prediction model with serial two-level decomposition based on bayesian optimization. *Complexity* **2020**, *2020*, 4346803. [CrossRef]
49. Luo, X.; Gan, W.; Wang, L.; Chen, Y.; Ma, E. A Deep Learning Prediction Model for Structural Deformation Based on Temporal Convolutional Networks. *Comput. Intell. Neurosci.* **2021**, *2021*, 8829639. [CrossRef] [PubMed]
50. Bokde, N.D.; Yaseen, Z.M.; Andersen, G.B. ForecastTB—An R package as a test-bench for time series forecasting—Application of wind speed and solar radiation modeling. *Energies* **2020**, *13*, 2578. [CrossRef]

51. Wong, T.T.; Yeh, P.Y. Reliable accuracy estimates from k-fold cross validation. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 1586–1594. [CrossRef]
52. Ding, F.; Lv, L.; Pan, J.; Wan, X.; Jin, X.-B. Two-stage Gradient-based Iterative Estimation Methods for Controlled Autoregressive Systems Using the Measurement Data. *Int. J. Control. Autom. Syst.* **2020**, *18*, 886–896. [CrossRef]
53. Zhang, X.; Ding, F. Hierarchical parameter and state estimation for bilinear systems. *Int. J. Syst. Sci.* **2020**, *51*, 275–290. [CrossRef]
54. Xu, L.; Ding, F.; Zhu, Q. Decomposition strategy-based hierarchical least mean square algorithm for control systems from the impulse responses. *Int. J. Syst. Sci.* **2021**, *52*, 1806–1821. [CrossRef]
55. Zhang, X.; Xu, L.; Ding, F.; Hayat, T. Combined state and parameter estimation for a bilinear state space system with moving average noise. *J. Frankl. Inst.* **2018**, *355*, 3079–3103. [CrossRef]
56. Pan, J.; Jiang, X.; Wan, X.; Ding, W. A filtering based multi-innovation extended stochastic gradient algorithm for multivariable control systems. *Int. J. Control Autom. Syst.* **2017**, *15*, 1189–1197. [CrossRef]
57. Zhang, X.; Yang, E.F. State estimation for bilinear systems through minimizing the covariance matrix of the state estimation errors. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1157–1173. [CrossRef]
58. Pan, J.; Ma, H.; Zhang, X.; Liu, Q.; Ding, F.; Chang, Y.; Sheng, J. Recursive coupled projection algorithms for multivariable output-error-like systems with coloured noises. *IET Signal Process.* **2020**, *14*, 455–466. [CrossRef]
59. Ding, F.; Liu, G.; Liu, X.P. Partially Coupled Stochastic Gradient Identification Methods for Non-Uniformly Sampled Systems. *IEEE Trans. Autom. Control* **2010**, *55*, 1976–1981. [CrossRef]
60. Ding, F.; Shi, Y.; Chen, T. Performance analysis of estimation algorithms of non-stationary ARMA processes. *IEEE Trans. Signal Process.* **2006**, *54*, 1041–1053. [CrossRef]
61. Wang, Y.J.; Ding, F.; Wu, M.H. Recursive parameter estimation algorithm for multivariate output-error systems. *J. Frankl. Inst.* **2018**, *355*, 5163–5181. [CrossRef]
62. Xu, L. Separable Multi-innovation Newton Iterative Modeling Algorithm for Multi-frequency Signals Based on the Sliding Measurement Window. *Circuits Syst. Signal Process.* **2022**, *41*, 805–830. [CrossRef]
63. Xu, L. Separable Newton Recursive Estimation Method Through System Responses Based on Dynamically Discrete Measurements with Increasing Data Length. *Int. J. Control Autom. Syst.* **2022**, *20*, 432–443. [CrossRef]
64. Zhang, X.; Ding, F. Adaptive parameter estimation for a general dynamical system with unknown states. *Int. J. Robust Nonlinear Control* **2020**, *30*, 1351–1372. [CrossRef]
65. Zhang, X.; Ding, F.; Xu, L. Recursive parameter estimation methods and convergence analysis for a special class of nonlinear systems. *Int. J. Robust Nonlinear Control* **2020**, *30*, 1373–1393. [CrossRef]
66. Zhang, X.; Ding, F. Recursive parameter estimation and its convergence for bilinear systems. *IET Control Theory Appl.* **2020**, *14*, 677–688. [CrossRef]
67. Liu, S.Y.; Ding, F.; Xu, L.; Hayat, T. Hierarchical Principle-Based Iterative Parameter Estimation Algorithm for Dual-Frequency Signals. *Circuits Syst. Signal Process.* **2019**, *38*, 3251–3268. [CrossRef]
68. Wan, L.J. Decomposition- and gradient-based iterative identification algorithms for multivariable systems using the mul-ti-innovation theory. *Circuits Syst. Signal Process.* **2019**, *38*, 2971–2991. [CrossRef]
69. Pan, J.; Li, W.; Zhang, H.P. Control Algorithms of Magnetic Suspension Systems Based on the Improved Double Exponential Reaching Law of Sliding Mode Control. *Int. J. Control Autom. Syst.* **2018**, *16*, 2878–2887. [CrossRef]
70. Ma, H.; Pan, J.; Ding, W. Partially-coupled least squares based iterative parameter estimation for multi-variable out-put-error-like autoregressive moving average systems. *IET Control Theory Appl.* **2019**, *13*, 3040–3051. [CrossRef]
71. Ding, F.; Liu, P.X.; Yang, H.Z. Parameter Identification and Intersample Output Estimation for Dual-Rate Systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2008**, *38*, 966–975. [CrossRef]
72. Ding, F.; Liu, X.P.; Liu, G. Multiinnovation least squares identification for linear and pseudo-linear regression models. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2010**, *40*, 767–778. [CrossRef] [PubMed]
73. Xu, L.; Ding, F.; Yang, E.F. Auxiliary model multiinnovation stochastic gradient parameter estimation methods for nonlinear sandwich systems. *Int. J. Robust Nonlinear Control* **2020**, *31*, 148–165. [CrossRef]
74. Xu, L.; Ding, F.; Wan, L.; Sheng, J. Separable multi-innovation stochastic gradient estimation algorithm for the nonlinear dynamic responses of systems. *Int. J. Adapt. Control Signal Process.* **2020**, *34*, 937–954. [CrossRef]
75. Zhao, Z.Y.; Zhou, Y.Q.; Wang, X.Y.; Wang, Z.; Bai, Y. Water quality evolution mechanism modeling and health risk assessment based on stochastic hybrid dynamic systems. *Expert Syst. Appl.* **2022**, *193*, 116404. [CrossRef]
76. Chen, Q.; Zhao, Z.Y.; Wang, X.Y.; Xiong, K.; Shi, C. Microbiological predictive modeling and risk analysis based on the one-step kinetic inte-grated Wiener process. *Innov. Food Sci. Emerg. Technol.* **2022**, *75*, 102912. [CrossRef]
77. Jin, X.-B.; Gong, W.-T.; Kong, J.-L.; Bai, Y.-T.; Su, T.-L. PFVAE: A Planar Flow-Based Variational Auto-Encoder Prediction Model for Time Series Data. *Mathematics* **2022**, *10*, 610. [CrossRef]
78. Jin, X.-B.; Zheng, W.-Z.; Kong, J.-L.; Wang, X.-Y.; Zuo, M.; Zhang, Q.-C.; Lin, S. Deep-Learning Temporal Predictor via Bi-directional Self-attentive Encoder-decoder framework for IOT-based Environmental Sensing in Intelligent Greenhouse. *Agriculture* **2021**, *11*, 802. [CrossRef]