Hindawi
Scientific Programming
Volume 2020, Article ID 8814222, 12 pages
https://doi.org/10.1155/2020/8814222

Research Article

Short-Term Infectious Diarrhea Prediction Using Weather and Search Data in Xiamen, China

Zhijin Wang,[1] **Yaohui Huang**,[2] **Bingyan He**,[1] **Ting Luo**,[2] **Yongming Wang**,[3] **and Yonggang Fu**[1]

[1]*Computer Engineering College, Jimei University, Xiamen 361021, China*
[2]*Chengyi University College, Jimei University, Xiamen 361021, China*
[3]*China Electronics Technology Group Corporation, Shanghai 200001, China*

Correspondence should be addressed to Yonggang Fu; yonggangfu@jmu.edu.cn

Received 23 April 2020; Revised 13 August 2020; Accepted 27 August 2020; Published 30 September 2020

Academic Editor: Chenxi Huang

Infectious diarrhea has high morbidity and mortality around the world. For this reason, diarrhea prediction has emerged as an important problem to prevent and control outbreaks. Numerous studies have built disease prediction models using large-scale data. However, these methods perform poorly on diarrhea data. To address this issue, this paper proposes a parsimonious model (PM), which takes historical outpatient visit counts, meteorological factors (MFs) and Baidu search indices (BSIs) as inputs to perform prediction. An experimental evaluation was done to compare the short-term prediction performance of ten algorithms for four groups of inputs, using data collected in Xiamen, China. Results show that the proposed method is effective in improving the prediction accuracy.

1. Introduction

To keep up with the pace of income growth, urbanization, and globalization, risk management of infectious diseases in public has become a critical task [1]. Infectious diarrhea (ID) [2] is one of the most common infectious diseases in the world, which infects more than 1 billion persons. It became the 37th legally notifiable disease in the China's National Notifiable Disease Reporting and Surveillance System, and any new case must be reported within 24 hours of diagnosis [3].

Early warning techniques [4] have been developed to monitor the status of infectious diseases and the demand for healthcare and health services. These techniques can support decision making for medical intervention strategies [5], by preinforming people, health service providers, and the government.

The problem of predicting upcoming diarrhea outpatient visits can be viewed as time series prediction problem. In past decades, numerous studies have used autoregression (AR), autoregressive integrated moving average model (ARIMA), and machine learning methods to predict upcoming values based on past observations. The widely used machine learning methods are multiple linear regression (MLR), support vector regression (SVR), and random forest regression (RFR) [6–8]. A famous autoregression method is that of Box and Jenkins [9], which has been applied in many fields [10], such as for electricity load forecasting and stock price prediction. Another famous statistical method is spline interpolation [11], which learns and uses a cubic spline interpolation to predict future values. But the performance of these methods degrades when dealing with nonstationary and chaotic time series, such as those of diarrhea outpatients.

Recently, to alleviate the uncertainty of a time series, exogenous data have been collected and fused into machine learning methods to achieve better predictions [12–14]. Hereto, the methods using exogenous data are called NARX [15]. According to the structure of NARX, we categorize them into *wide models* and *deep models*.

A wide model commonly builds more than two components in a layer. To capture the sequential features of a time series, a recurrent neural network (RNN) [16] is adopted to process exogenous data. To discriminatively process exogenous data and historical observations, the encoder-decoder framework [17] is introduced to undertake predictions. Meanwhile, the gated recursive unit (GRU) [18] is used to replace RNN in the framework, which captures long- and short-term memory (LSTM) [19]. Moreover, an attention mechanism is designed to adjust the values of exogenous inputs and historical inputs in the stage of encoding and decoding, respectively [20–22].

The deep model consecutively connects components from inputs to outputs using more than two neural network layers. For example, DL4Epi in [23] consists of a CNN layer, a RNN layer, and a residual link layer. DilatedRNN in [24] dilated three RNN layers from input to predict glucose incidence rate. TPA-LSTM in [25] uses a temporal pattern attention layer to deepen a model as well.

However, not only wide models but also deep models suffer from the problem of needing numerous samples. Given a time series of weekly or monthly diarrhea outpatients, the number of training samples is usually in hundreds. If a model has lots of parameters and the training samples are few, the learned model has a poor generalization performance. In the following, we call a model having many parameters a *heavy model*. In reality, heavy models remember almost the training samples and thus perform poorly for predicting unknown inputs.

How to build a parsimonious model, which has few parameters and learns from hundreds of samples? Observations about from previous models reveal that models assign weights to input elements, and then weights are passed and transformed into the target. Can the weights of elements be reduced? Can the reduced weights be passed to the output?

To address the issues, we propose a parsimonious model (PM). The proposed model first assigns a vector to each input dimension, and then the vectors are connected to the target.

The rest of this paper is organized as follows. Section 2 briefly introduces the study area and data sources. Section 3 illustrates the defined problem and presents the proposed method. Section 4 describes the experimental setting and related benchmarks. Section 5 compares different methods and analyses their sensitiveness. Finally, a conclusion is drawn in Section 6.

2. Study Materials

This section first introduces the study area, i.e., Xiamen city. Then, it briefly describes meteorological factors and search behavior in this city using basic descriptive statistics.

The basic statistical variables of inputs and the target are listed in Table 1.

The reasons for choosing meteorological factor (MF) data and Baidu search index (BSI) data as exogenous data are that those data are significantly correlated to the infectious diseases. MF data have been proved to be associated with the incidence of infectious diseases [26, 27], such as ID and "hand, foot, and mouth disease" (HFMD). As an important entrance for users to obtain digital information resources on the Internet, the search engine provides lots of useful information. The search index of specific keywords directly reflects the social attention on infectious diseases [28–30]. Hence, MF data and BSI data are considered as exogenous inputs to models.

In order to determine the relationships between exogenous variables and the target variable, the Person correlation coefficient (PCC) values are figured out in the last column of Table 1. All of the exogenous factors are significantly correlated to weekly outpaitent counts except weekly maximum temperature. $H_{\max}$ is used as an important input, since humidity is considered as an important factor [27].

2.1. Study Areas. Xiamen is a developing city, which has a population of around 4 million permanent residents and around 4 million temporary residents as of 2018. It is an important special economic zone in China since 1980 and is located in southeast China. It covered a land area of 1,699.39 km^{2}and a sea area of over 390 km^2 in 2017. Xiamen has a monsoonal humid subtropical climate, characterised by long, hot, and humid summers (but moderate compared to much of the rest of the province) and short, mild, and dry winters, and the annual mean temperature is 20.7°C [31].

Herein, the detailed population from 2012 to 2016 is listed in Table 2 as a background for this study. The annual growth of its population is between 1.3% and 2.1%, which is relatively stable. Morbidity and mortality are influenced by disease trends within a period. Meanwhile, infectious diarrhea is the most common infectious disease. Hence, the number of infectious diarrhea outpatient visits is key to monitor the status of infectious diseases in a city.

2.2. Meteorological Factor (MF). The MF data from January 1, 2012, to December 31, 2016, were collected from Weather Underground (http://www.wunderground.com), which is a popular personal weather website. Those data from meteorological monitoring sites provide globally comprehensive, timely, and reliable meteorological data. At the moment, this website provides web-based application program interfaces for users to download data or develop third-party applications.

The provided weather information is formatted in days. We calculate 261 weekly weather information pairs by those daily records. The selected weather information in weeks consists of 12 factors, e.g., highest temperature (°C), average dew point (°C), lowest atmospheric pressure (hPa), and average relative humidity (%). Thus, we get 12 groups of exogenous data to describe weather conditions.

2.3. Baidu Search Index (BSI). Baidu is the most widely used search engine (http://index.baidu.com) in China, and it is also the largest Chinese search platform. People in Xiamen, China, are accustomed to using that search engine as well. Baidu search index records the search volume of many terms, which are queried by users since January 2011, and it is publicly available. BSI has been used to monitor the

Table 1: Basic statistical characteristics of exogenous time series and infectious outpatient visits (261 samples).p 0.01

Symbols	Parameter (unit)	Range		Mean	Median	STD	PCC
		Min	Max				
$T_{\max}$	Maximum temperature (°C)	14	39	29.21	31	5.64	−0.342**
$T_{\min}$	Minimum temperature (°C)	1	27	16.9	18	6.54	−0.304**
T_{avg}	Average temperature (°C)	9.29	31.29	22.32	23.14	6.03	−0.317**
$D_{\max}$	Maximum dew temperature (°C)	9	29	22.32	23	4.94	−0.385**
$D_{\min}$	Minimum dew temperature (°C)	−12	24	10.57	11	9.32	−0.345**
D_{avg}	Average dew temperature (°C)	3.57	26.71	17.47	18.14	6.54	−0.393**
$H_{\max}$	Maximum relative humidity (%)	69	100	98.33	100	4.25	−0.092
$H_{\min}$	Minimum relative humidity (%)	9	70	39.78	42	13.59	−0.372**
H_{avg}	Average relative humidity (%)	52.14	94.71	75.18	75.43	9.5	−0.339**
$A_{\max}$	Maximum atmospheric pressure (hPa)	1004	1037	1018.84	1019	7.12	0.381**
$A_{\min}$	Minimum atmospheric pressure (hPa)	964	1021	1007.3	1008	7.46	0.392**
A_{avg}	Average atmospheric pressure (hPa)	998.1	1026.6	1013.56	1013.7	6.63	0.342**
$B_{p,c}$	BSI of “腹泻” in PC, China (count)	1953	6000	3804.34	3584	998.43	−0.181**
$B_{m,c}$	BSI of “腹泻” in mobile, China (count)	5764	9121	6415.20	6328	447.32	−0.241**
$B_{t,c}$	BSI of “腹泻” in total, China (count)	7971	12500	10219.53	10046	978.45	−0.325**
$B_{p,x}$	BSI of “腹泻” in PC, Xiamen (count)	0	440	311.03	303	85.07	−0.323**
$B_{m,x}$	BSI of “腹泻” in mobile, Xiamen (count)	0	515	307.31	430	205.63	−0.348**
$B_{t,x}$	BSI of “腹泻” in total, Xiamen (count)	0	925	618.34	683	194.09	−0.324**
C_d	Infectious diarrhea outpatient visits (case)	15	349	69.7	52	50.03	—

“STD” denotes standard deviation. ** indicates that p value is less than 0.01, which means significant correlation.

Table 2: Summary of the population in Xiamen [32].

Year	Population (million)	Population growth rate (%)
2012	3.67	
2013	3.73	1.6
2014	3.81	2.1
2015	3.86	1.3
2016	3.92	1.6

incidence of influenza epidemics [33], analyse regional infectious diseases, and perform real-time monitoring and prevention [28].

A major concern to acquire query data (search indices) is to find proper query words [29], which reflects user behavior about searching for infectious diarrhea. The engine returns daily counts of a given keyword, which can be conditioned by a region and a platform. Six time series of each query word are downloaded by choosing a region from {Xiamen, countrywide} and a platform from {mobile, PC, total}, respectively. Those data pairs are calculated in weeks. We have six groups of search indices, and they are used as exogenous data for each keyword. A correlation coefficient analysis is carried out on those input indices and target output (i.e., the number of cases). We found that the indices of the keyword “腹泻” (i.e., ID) have significant correlation, while others do not. Hence, we adopt the six group indices as inputs of models.

3. Approach

This section first defines the problem of time series prediction using exogenous data and then introduces data preprocessing and postprocessing. Finally, the proposed method is presented.

3.1. Problem Formulation. The epidemic prediction problem can be viewed as a time series forecasting task [10]. A time series is defined as a sequence of observation values with consecutive identical interval lengths.

Let $y_t \in \mathbb{R}^1$ denote the observation measured at time t and $\mathbf{x}_t \in \mathbb{R}^n$ denote the exogenous data measured at time t, where n is the number of input series. Furthermore, let T be a time window size. The known exogenous series with window size T is symbolized as $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$, and the historical target observations are denoted as $[y_1, y_2, \ldots, y_T]$.

The goal is to predict the value of a future time point $\hat{y}_{T+1}$, given historical observations and exogenous series. In detail, a nonlinear mapping $F(\cdot)$ is applied into the predictive formula:

$$\hat{y}_{T+1} = F(y_1, y_2, \ldots, y_T, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T). \tag{1}$$

Moreover, let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ denote the available training data in a time-span of size T and $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ be the recent historical outpatient visit counts.

3.2. One-Step Forward Split and Normalization

3.2.1. One-Step Forward Split. A time series cannot be directly used as input of regression models. Hence, we conduct one-step forward split to transform a time series to supervised data. The formula of the one-step split is shown as follows:

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_T & \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_T \\ y_2 & y_3 & \cdots & y_{T+1} & \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{T+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{M-T-1} & y_{M-T} & \cdots & y_{M-1} & \mathbf{x}_{M-T-1} & \mathbf{x}_{M-T} & \cdots & \mathbf{x}_{M-1} \end{bmatrix} \longrightarrow \begin{bmatrix} y_{T+1} \\ y_{T+2} \\ \vdots \\ y_M \end{bmatrix}, \tag{2}$$

where M is the length of inputted time series and T is the window size.

3.2.2. Normalization. It is a process of rescaling the data from the original range so that all values are within a given range [34], usually within 0 and 1 or 0.05 and 0.95. Both min-max normalization and standard (i.e., z-score) normalization methods are commonly applied to normalize time series data.

Min-max normalization scales data in the [0, 1] interval by using the bounds of the minimum and maximum values. Standardizing a dataset involves rescaling the distribution of values, so that the mean of observed values is 0 and the standard deviation is 1. The mean and standard deviation estimates of a dataset can be more robust to new data than the minimum and maximum.

We choose standard normalization to rescale inputs and the target, since we regard the outpatient visit counts as Gaussian distribution, and the standardization fits a Gaussian distribution well. The standard normalization of inputs is formulated and recovered as

$$\begin{aligned}\mathbf{d}' &= \frac{\mathbf{d} - \mu(\mathbf{d})}{\sigma(\mathbf{d})},\\ \mathbf{d} &= \mathbf{d}' \cdot \sigma(\mathbf{d}) + \mu(\mathbf{d}),\end{aligned} \tag{3}$$

where $\mathbf{d} \in \mathrm{R}^M$ denotes a feature of observed samples, M is the number of observed samples, $\mathbf{d}'$ is the normalized data, $\mu(\mathbf{d})$ is the mean value of $\mathbf{d}$, and $\sigma(\mathbf{d})$ is the standard variation of $\mathbf{d}$. The recovered formula is applied on model outputs in the postprocessing stage. Because normalization is only applied on exogenous data, the recovery action can be skipped while generating predictions.

3.3. Parsimony Model (PM). The diagram of the proposed PM is shown in Figure 1. This diagram consists of three stages: data preprocessing, PM, and postprocessing. In the preprocessing stage, the exogenous time series is normalized and combined with target time series to split the supervised data using one-step forward. In the PM processing stage, a parameter layer is exploited to extract pattern features from heterogeneous inputs. The outputs of the model are generated by linear summarization of the intermediate state result.

The left bottom part of Figure 1 gives toy examples of the structure of the parsimonious method with respect to inputs and weights. The yellow solid circle denotes exogenous inputs, such as meteorological factor or Baidu search index. The blue solid rectangle denotes historical observed outpatient visits, and red solid rectangle is the target. The yellow solid rectangles are weights to be learned. The goal is to reduce these complex models and find a parsimonious model to learn and predict well on the diarrhea data collections.

Therefore, the weights of the first layer in a neural network are exploited to reconsider model structures. The left bottom part of Figure 1 gives inputs of exogenous data (i.e., MF, BSI, or MF + BSI) and historical cases, and the right part indicates the weights and connections among them (arrows between circles and rectangles). The two following definitions are introduced to provide clearer explanations.

Definition 1. A weight is assigned to each input element.

Definition 1 is formulated as follows:

$$F_1(\mathbf{W}, [\mathbf{X}; \mathbf{y}]) \longrightarrow \hat{y}_{T+1}, \tag{4}$$

where $[\mathbf{X}; \mathbf{y}] \in \mathbb{R}^{(n+1)\times T}$ is the input of a given sample (i.e., input matrix), $\hat{y}_{T+1}$ is the target of this sample, which means the outpatient visit counts of the upcoming week, $\mathbf{W} \in \mathbb{R}^{(n+1)\times T}$ is the weight corresponding to the inputs (i.e., weight matrix), and F_1 is a linear or nonlinear mapping from input and weight matrices to target.

Both wide models and deep models in the research field of epidemic prediction are based on this definition. A wide model processes $\mathbf{X}$ and $\mathbf{y}$ in different components. For example, the encoder-decoder structure encodes $\mathbf{X}$, passes a code to the decoder, and the decoder processes the code and y to make predictions. A deep model adds layers on the sequence of $\mathbf{W}$, such as CNN and RNN.

When learning with a small number of samples, the number of samples is usually in hundreds, and these models have a burden of parameter training. In reality, the number of parameters is greater than the number of samples times the number of exogenous factors. Hence, these trained models would remember all the training samples and have a value of loss close to 0, but they poorly predict upcoming events.

Therefore, the wide model and the deep model are condensed into a light neural network by setting F_1 as a linear function or nonlinear function, such as sigmoid function. But the condensed light neural network has many parameters as well. Thus, we try to use two vectors to restore weight $\mathbf{W}$ in equation (4). The aim is not only to have fewer parameters during the training stage but also to obtain better performance at the predicting stage.

Definition 2. A weight is assigned to each input dimension.

Definition 2 is formulated as follows:

$$F_2(\mathbf{w}_r, \mathbf{w}_c, [\mathbf{X}; \mathbf{y}]) \longrightarrow \hat{y}_{T+1}, \tag{5}$$

where $[\mathbf{X}; \mathbf{y}] \in \mathbb{R}^{(n+1)\times T}$ is the input matrix, and $\hat{y}_{T+1}$ is the prediction target (i.e., the upcoming week outpatient counts), $\mathbf{w}_r \in \mathbb{R}^{n+1}$ and $\mathbf{w}_c \in \mathbb{R}^T$ are the weights corresponding to two input dimensions (i.e., weight vectors), and F_2 is a linear function or a nonlinear mapping from input and weight matrices to the target.

There are limitations in accurately capturing the input matrix of a sample solely on the basis of using two vectors, whose lengths are the same as the size of input matrix. Hence, we exploit two ways to recover $\mathbf{W}$ in equation (4) using $\mathbf{w}_r$ and $\mathbf{w}_c$: the addition method and the multiplication method.

For each element w_{ij} in $\mathbf{W}$, it can be recovered using the addition method. The addition recovery formula is

$$\widehat{w}_{ij} = w_{ri} + w_{cj}, \tag{6}$$

where $\widehat{w}_{ij}$ is the recovered value, $w_{ri} \in \mathbf{w}_r$ denotes the i-th element of $\mathbf{w}_r$, and $w_{cj} \in \mathbf{w}_c$ denotes j-th element of vector

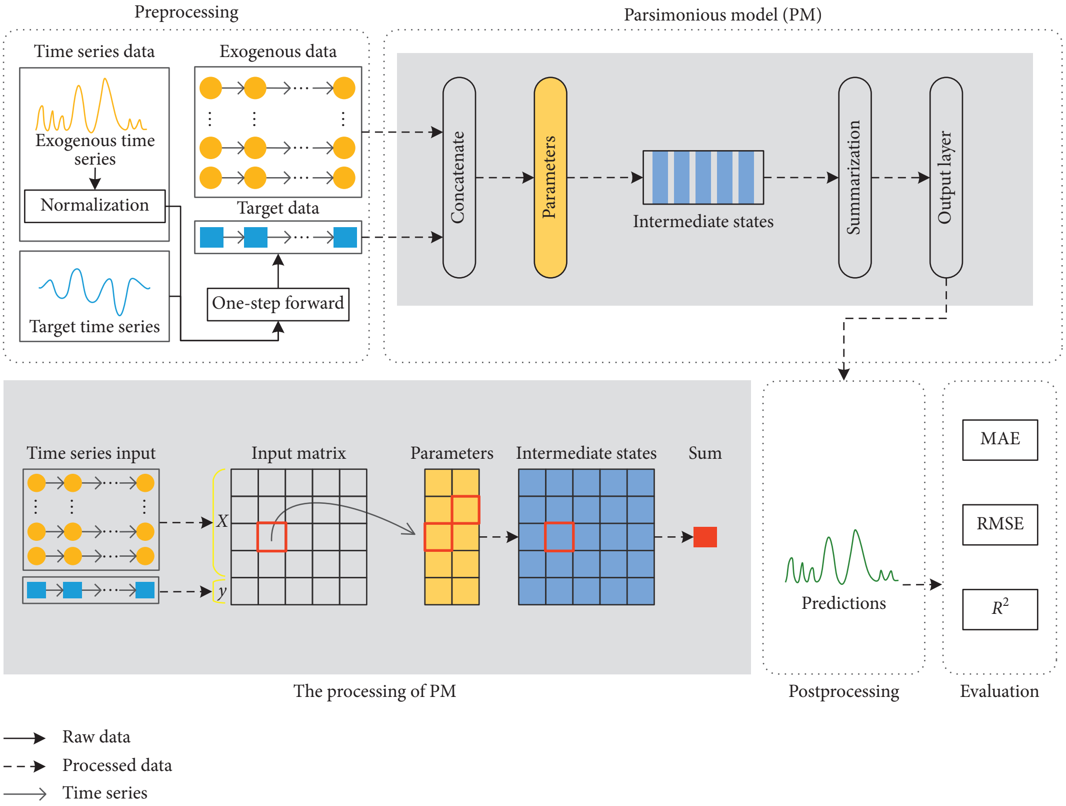


Figure 1: The diagram of the proposed PM.

$\mathbf{w}_c$, respectively. Equation (6) shows that a matrix can be recovered by two vectors, which would reduce the training parameters.

Another recovery method is the multiplication method. For each element w_{ij} in $\mathbf{W}$, it can be recovered using the multiplication method. The multiplication recovery formula is

$$\widehat{w}_{ij} = w_{ri} \cdot w_{cj}, \tag{7}$$

where $\widehat{w}_{ij}$ is the recovered value, $w_{ri} \in \mathbf{w}_r$ denotes the i-th element of $\mathbf{w}_r$, and $w_{cj} \in \mathbf{w}_c$ denotes j-th element of vector $\mathbf{w}_c$, respectively. Equation (7) shows that a matrix can be recovered by two vectors, which would reduce the training parameters as well.

To benefit the recovery effects from the both methods, we add them together and get

$$\widehat{w}_{ij} = \left(w_{ri} + w_{cj}\right) + w_{ri} \cdot w_{cj}. \tag{8}$$

According to equations (8) and (5), we have

$$F_2\left(\widehat{\mathbf{W}}, [\mathbf{X}; \mathbf{y}]\right) \longrightarrow \widehat{y}_{T+1}, \tag{9}$$

where $\widehat{w}_{ij} \in \widehat{\mathbf{W}}$. Lots of methods applied to equation (5) can be used to equation (6) as well, whereas the goal of this paper is trying to find a parsimony method to fit well with small-scale datasets. Hence, F_2 in equation (9) should be as simple as possible.

We adopt the summarization and sigmoid activate function to replace F_2 and make it be simple. The prediction function is formulated as

$$\widehat{y}_{T+1} \leftarrow \sum_{j=1}^{T} \sum_{i=1}^{n+1} \left(\left(w_{ri} + w_{cj}\right) + w_{ri} w_{cj}\right) s_{ij}, \tag{10}$$

where $s_{ij} \in [\mathbf{X}; \mathbf{y}]$ denotes the element in the i-th row and j-th column of input matrix $[\mathbf{X}; \mathbf{y}]$. Equation (10) is called the prediction function of the proposed PM.

4. Experimental Settings

This section first describes the exogenous feature selection, experimental settings, and relevant related work, which are compared with our proposed method. Finally, an analysis of the comparable results is provided.

4.1. Experimental Configuration. The disease data collection is divided into two subsets: the first part, from the 1st week of 2012 to the 52nd week of 2015, is used to build and train model; the remaining part, from the 1st to the 52nd week of 2016, is utilized to assess the learned models.

All neural models are trained using the Adam optimizer [35]. The batch size is set to 32. Their learning rate is set to 0.001, and mean squared error (MSE) is chosen as the loss function. For RNN and LSTM, the number of hidden neurons is set to {64, 128}.

We ran each method five times and reported the median. The optimal RMSE values of models at given T and input data are in bold types. DA-RNN [20] and DL4Epi [23] have been applied on the four groups of inputs, and they are not converged in training stage. Hence, their results are not listed and compared.

4.2. Evaluation Metrics. A number of performance evaluation criterion have been employed to evaluate and compare the performance of models, but there is a no uniform standard. Therefore, we evaluate models based on error metrics, which are commonly adopted in regression performance evaluation.

The evaluation criterion consists of the mean absolute error (MAE), the root mean square error (RMSE), and the coefficient of determination (R^2). These metrics are expressed as the following mathematical expressions:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left(\left|y_{T+1}^{i} - \hat{y}_{T+1}^{i}\right|\right),$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{T+1}^{i} - \hat{y}_{T+1}^{i}\right)^{2}},$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_{T+1}^{i} - \hat{y}_{T+1}^{i}\right)^{2}}{\sum_{i=1}^{N} y_i^2}, \tag{11}$$

where y_{T+1}^{i} is an actual value at week (relative time) i in the test set, $\hat{y}_{T+1}^{i}$is a predicted value at week (relative time) i, $\overline{y}_{T+1}$ is the mean value in the test set, and N is the number of weeks in the test period. The model providing the smallest MAE and RMSE and the largest R^2 is considered to have the best performance.

4.3. Comparable Methods. According to the type of input variables, the prediction methods are categorized into univariate models and multivariate models. We choose models which can work with multiple inputs and unstable series and keep ARIMA as a baseline.

4.3.1. Multiple Linear Regression (MLR). It is widely adopted to model the connection between several independent variables and dependent variables. The prediction model of MLR based on generalized MLR analysis formula is given below:

$$\begin{aligned}\hat{y}_{T+1} &= \mathbf{w}_y' \cdot \mathbf{y} + b + \varepsilon \\ &= \sum\nolimits_{i=1}^{T} w_{y,i} \cdot y_i + b + \varepsilon,\end{aligned} \tag{12}$$

where $\mathbf{w}_\mathrm{y} \in \mathbb{R}^T$ and b are weights and bias to be learned, T is the size of the historical data, ε represents the random error term, and $E(\varepsilon) = 0$.

We extend MLR to model exogenous inputs and target as follows:

$$\hat{y}_{T+1} = \mathbf{w}_{xy}' \cdot \left[x_1; x_2; \ldots; x_T; y\right] + b + \varepsilon, \tag{13}$$

where $\mathbf{w}_{xy}' \in \mathbb{R}^{(n+1)T\times 1}$ denotes weights of exogenous inputs and historical data at past T time intervals.

4.3.2. Random Forest Regression (RFR). It is a CART [36] method. It has been used to predict the death characteristics of patients, postoperative prognosis of hepatocellular carcinoma, and other data [37]. Random forest regression (RFR) consists of a collection of unpruned regression trees using different bootstrap samples of the training data. In each bootstrap sample, a random sample with replacement and with the same length, some of the data are repeated, and the left-out samples are called out of bag (OOB). In practice, the number of trees n_trees and the size of the variable subset n_subsets should be optimized to reach the ideal forest by minimizing the OOB error. The values of the parameters n_trees and n_subsets were optimized simultaneously by using the grid-search method ranging from 10 to 1000 (with step size 10) and from 1 to 9 (with step size 1), respectively. The parameter values which give the lowest RMSE of the OOB data were selected as an indicator for the performance.

4.3.3. Support Vector Regression (SVR). It is a nonlinear kernel-based regression method, which tries to find the best regression hyperplane with smallest structural risk in a high dimensional feature space. It has been used in many medical related applications, such as diagnosis of the incidence of infectious diarrhea [26]. The relationship between input (or inputs) and output is formulated as

$$\begin{aligned}\hat{y}_{T+1} &= \mathbf{w}_{xy}' \cdot \varphi(\mathbf{y}) + b, \\ \hat{y}_{T+1} &= \mathbf{w}_{xy}' \cdot \varphi([\mathbf{X}; \mathbf{y}]) + b,\end{aligned} \tag{14}$$

where $\phi(\cdot)$ denotes kernel function, $\mathbf{x}_t$ is a vector of inputs at time t, and b is a bias term. Radial basis function expressed as $K(a, b) = \exp(-|a - b|^2/\sigma^2)$ was used as a kernel function because of its advantages and simple implementation only one tuning parameter.

4.3.4. Gradient Boosting Regression (GBR). It is a powerful regression model that enhances the decision tree model using gradient boosting, which produces a regression model by combining a series of weak prediction models [38].

GBR iteratively constructs decision trees, and the newly added decision tree is trained according to the negative gradient information of the loss function from the current model. The goal of GBR is to learn an optimal model $F(x)$ that minimizes $\sum_{i=1}^{n} L(y_i, F(x_i))$ for a specified loss function $L(y, F(x))$. The optimal model $F^*(x)$ of GBR can be calculated as follows:

$$F^*([\mathbf{X}; \mathbf{y}]) = F_0([\mathbf{X}; \mathbf{y}]) + \nu \sum_{i=1}^{m} \rho_i h_i([\mathbf{X}; \mathbf{y}]), \tag{15}$$

where $F_0(\cdot)$ is the first built decision tree, m is the number of iterations, ν represents the shrinkage parameter that controls the learning rate of GBR, $h_i(\cdot)$ denotes the tree trained in the

i-th iteration, and ρ_i is the weight of $h_i(\cdot)$. The generated optimal model is used for testing.

4.3.5. Extreme Gradient Boosting Regression (XGBoost). It is an improvement on the gradient boosting algorithm [39]. XGBoost adds the regularization terms during the decision tree construction phase. The loss function is

$$\mathscr{L}^{(t)} = \sum_{t}^{T} l(y_t, \hat{y}_t) + \sum_{k} \Omega(f_k), \quad (16)$$

where $\hat{y}_t$ represents the prediction of the t-th training sample and Ω is the regularization term. The regularization term is calculated as follows:

$$\Omega(f) = \gamma N + \frac{1}{2}\lambda|\mathbf{w}|^2, \quad (17)$$

where N is the number of leaf nodes, $\mathbf{w}$ is the vector formed by all leaf node values of the decision tree, and γ and λ are manually set parameters. Similar to GBR, the goal of XGBoost is also to minimize the loss function. Besides, XGBoost uses weight shrinkage and column sampling techniques to resist overfitting.

4.3.6. Convolutional Neural Networks (CNN1d). CNN [40] consists of convolutional layers that are based on the convolutional operation. Filtering with kernel window function gives an advantage of image or series processing to CNN architectures with fewer parameters, which are beneficial for computing and storage [41–43]. The basic convolution operation is shown as follows:

$$s = \text{relu}([x_1; x_2; \ldots; x_T]^* \mathbf{W}), \quad (18)$$

where s denotes mapped space and $\mathbf{W}$ denotes kernel. CNN has been verified to have good accuracy results when applied to pattern recognition [44].

4.3.7. Neural Network Regression (NNR). On account of the neural network (NN) approximation and generalization property, NN-based prediction is widely used. Recurrent neural network is a type of deep neural network specially designed for sequence modelling [16]. The main idea of RNN is to provide a weighted feedback connection between layers of neurons and add time significance to the entire artificial NN. But RNN may face the problem of vanishing gradient or gradient explosion during backpropagation. LSTM was proposed by Hochreiter and Schmidhuber in [45].

We adopt a lite RNN model like NARX in [15], which consists of an input layer, an RNN layer, and an output layer, to test its performance. The nonlinear sequential layer can be a LSTM layer or a gated recurrent unit (GRU) [18] layer as well.

5. Experimental Results and Analyses

This section presents the results of intensive experiments to evaluate algorithms over four kinds of inputs and analyses them. Figure 2 shows the performance of PM with varying window size T. Figure 3 shows the evaluation results of all the methods over all the inputs for all metrics. Figure 4 gives the visual comparison of PM methods over four groups of inputs.

5.1. The Impact of Window Size T. Parameter sensitivity is a significant part of experimental analysis. In time series prediction study, T is the most interesting parameter of time delay. Different window sizes will affect the parameter complexity and the overall prediction performance of the methods. In addition, window size is an important indicator to reveal the incubation period of infectious diseases. Therefore, this study visualized the influence of different window sizes on the proposed model. The performance of PM with varying window size T in terms of MAE, RMSE, and R^2 is plotted in Figure 2.

According to the results, when we vary T and keep other parameters fixed, it is easily observed that the performance of PM degraded when T increases from 4 to 20, and the performance tends to get worse as the length of delay time steps continues to increase. A possible reason is that the longer lagged time delay leads to larger feature dimension of the inputs, which causes an increase in the complexity of the regression function in the training process and directly influences the weighted parameter representation for small-scale sample, resulting in the regression function unable to accurately fit the input. By setting $T = 4$, we notice that PM usually achieves the best performance, which possibly suggests that the diarrhea infection cycle pattern lags about four days in Xiamen. This is an important artificial parameter required by the predictive model.

5.2. The Performance of Time Series Prediction. The optimal performance of the benchmark and proposed methods is found at $T = 4$. Hence, the evaluation results of all the comparable methods are shown in Figure 2 when T is fixed at 4.

Several important observations are made about these results:

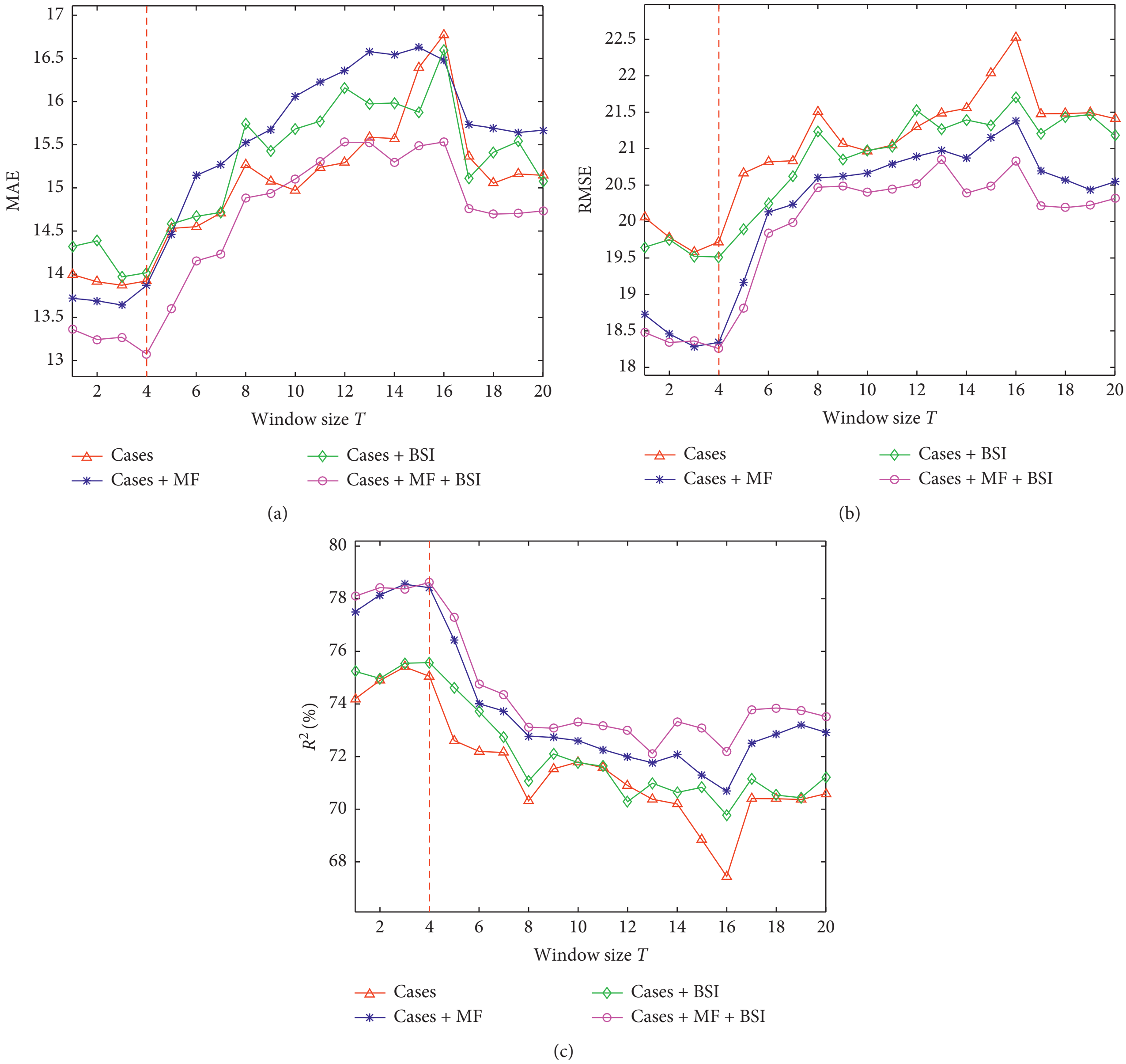
(1) By benefiting from MF and BSI data, the PM model obtains the best performance in terms of three metrics

(2) The performances of MLR and GBR are better than some complex RNN methods, and ARIMA cannot perform well under unstable time series

(3) CNN1d and RNN methods can achieve stable performance when processing each type of inputs

(4) Both MF and BSI data can improve the prediction accuracy of diarrhea outpatient counts in upcoming weeks.

To demonstrate the effectiveness of PM, we compare the performance of PM against some common methods for the same prediction tasks. The comparable experiments choose MLR, GBR, RFR, SVR, and XGBoost as the representative of the traditional methods and three other deep learning



(a)

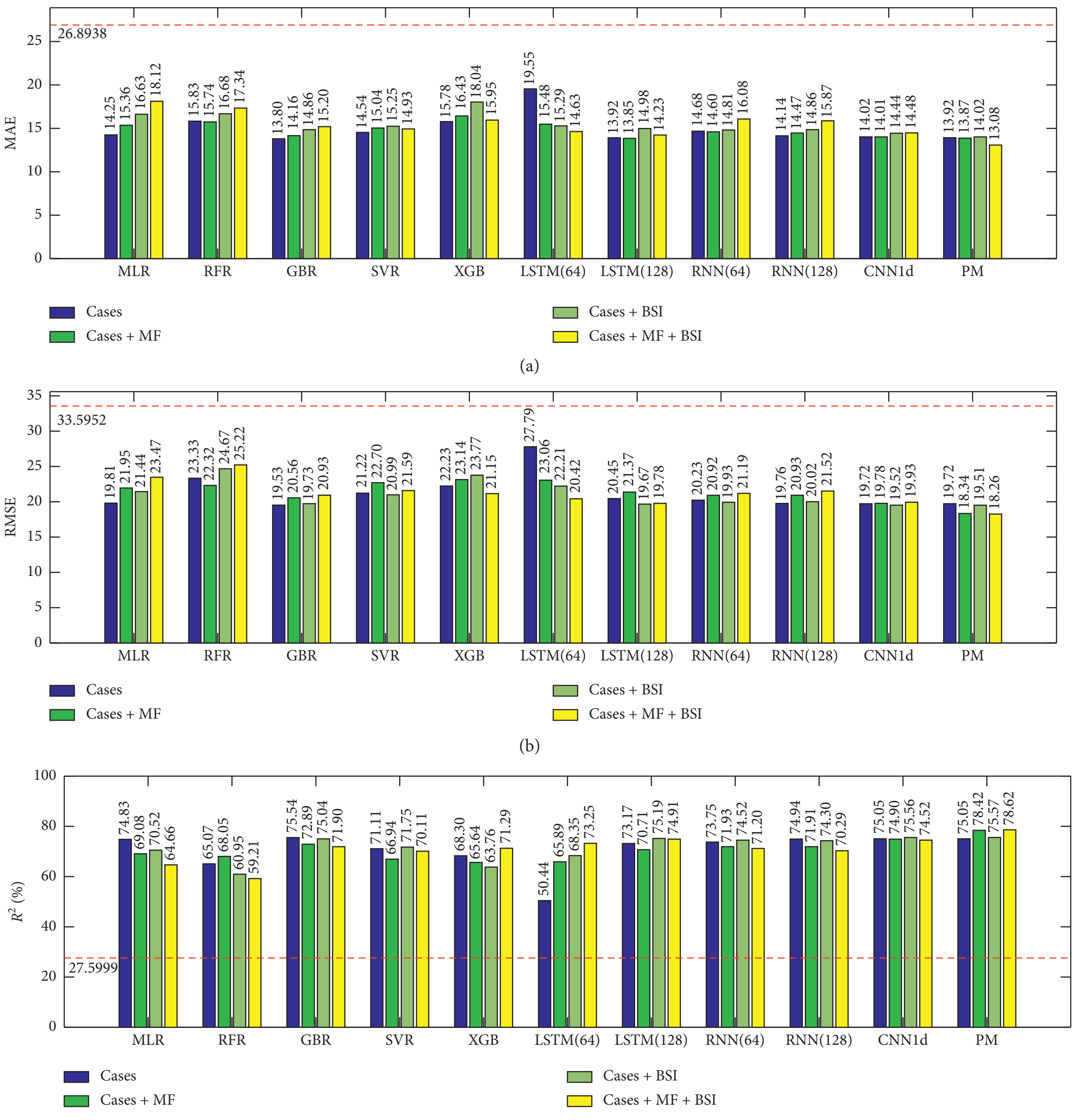
(b)

(c)

Figure 2: The performance of PM with varying window size T in terms of MAE, RMSE, and R^2. (a) The MAE of PM with varying T. The minimum value is found at $T = 4$ (see the red dashed line). (b) The RMSE of PM with varying T. The minimum value is found at $T = 4$ (see the red dashed line). (c) R^2 of PM with varying T. The maximum value is found at $T = 4$ (see the red dashed line).

methods. The deep learning methods include CNN, RNN and LSTM with {64, 128} hidden nodes.

Comparing the performances of all the methods over the four inputs, according to the results of predictions based on historical cases, we can observe that the PM method shows the best performance, and CNN1d, RNN, GBR, and MLR also have good results. The performance of MLR is better than the performance of deep learning methods when the inputs are solely on the basis of historical cases. A possible reason is that the small-scale data collections cannot train deep learning methods well, especially the LSTM method, either overfitting or underfitting. According to the results of predictions based on cases+BSI, we can observe that the traditional m`ts that common learning methods cannot extract sufficient features from high dimensional inputs to predict the upcoming values. According to the results of predictions based on historical cases and BSI, we can observe that the performance of all the methods has degraded, which suggests that too many missing values will greatly affect the prediction performance of the model. However, PM has ability to adapt to input and to deal with missing values, which is significantly stronger than other models. According to the the inputs are solely on the basis of predictions based on cases+MF+BSI, we can observe that PM obtains the best results. Compared to the CNN1d method, it is difficult for traditional learning models to give stable performance with exogenous inputs. The prediction accuracy of the LSTM method has a slight improvment when compared with MLR.



(a)

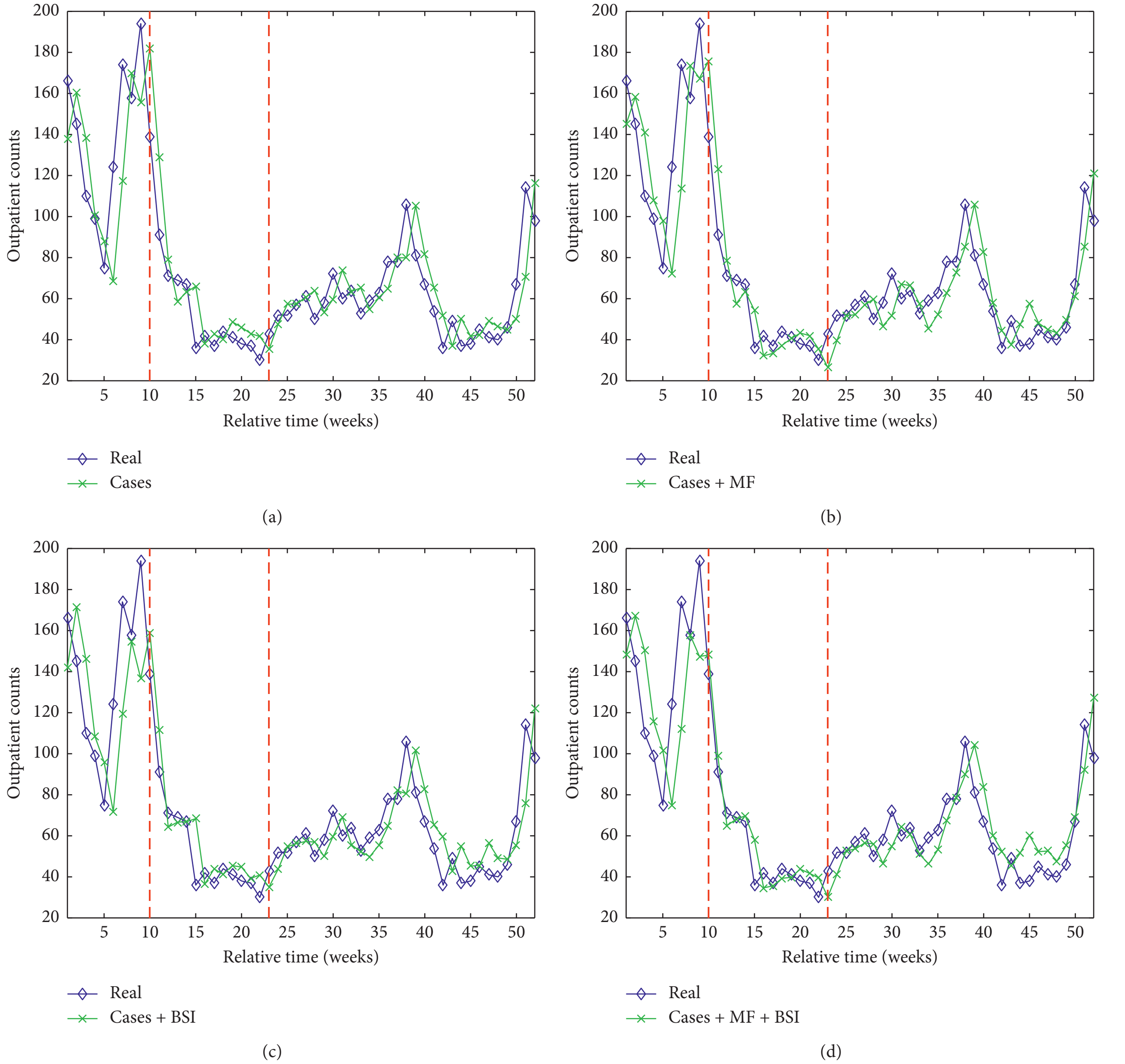
(b)

(c)

Figure 3: Comparisons of eleven methods on four inputs in terms of three metrics. T is fixed at 4. Lower MAE and RMSE values and higher R^2 value indicate better performance. (a) MAE comparison on methods over inputs (the red dashed line denotes ARIMA). (b) RMSE comparison on methods over inputs (the red dashed line denotes ARIMA). (c) R^2 comparison on methods over inputs (the red dashed line denotes ARIMA).

The prediction effect of the RNN method fluctuates significantly. It is suggested that exogenous inputs can effectively improve model performance, but it also related to the adaptability of the model's structure to exogenous inputs.

There are two potential reasons for the improvements in the PM method. On the one hand, the RNN models only consider the temporal dynamics of exogenous inputs. With small-scale samples, these models cannot extract sufficient features to predict the target. The traditional machine learning methods are easily converged and have worse performance in predicting the upcoming values with exogenous inputs. A possible reason is that these methods cannot capture the correlations among different components of the inputs. The proposed method has ability to learn the interactions of different exogenous factors by introducing the weighted layers, which can effectively improve model performance in small-scale data collections through more concise feature representation. On the other hand, and



(a)

(b)

(c)

(d)

Figure 4: The visualizations of real values and PM predictions. The window size T is set to 4, where the best performance is found. The red dashed lines mark the significant differences, whose values follow the highest and lowest real values, respectively. (a) Real values vs. PM predictions based on cases. (b) Real values vs. PM predictions based on cases + MF. (c) Real values vs. PM predictions based on cases + BSI. (d) Real values vs. PM predictions based on cases + MF + BSI.

two weights for each input element greatly decreases the need of parameters, which can be updated by the training process, and the two weighted vectors are adopted to learn temporal weights and factor weights.

When compared with CNN1d, the proposed method exploits the multiplication and addition of two vectors to generate predictions instead of using a kernel matrix to calculate the inputted 2d data to get output values. When T is small, the convolution kernel of CNN has poor performance, since the kernel size should smaller than input size. Moreover, not only wide or deep models but also machine learning models have abilities in overcoming overfitting using decay weights (i.e., regularization items) in their loss functions. The key concern is that more parameters will lead to hard training on small-scale data collections. Therefore, PM is more concise and effective, which has better results and better adaption on broad inputs.

Another considerable result is that the performance differences in results among the methods are enlarged with complicated inputs. For example, when only considering the outpatient cases with T = 4, PM's performance is similar to other models. But the performance differences become larger when fusing exogenous inputs, since the mined patterns from heterogeneous data are more complex than those found in only outpatient case data. PM has a better ability to handle small-scale heterogeneous input data. The results of correlation experiments also show that it is completely feasible to use exogenous factors to predict the incidence of infectious diarrhea in Xiamen.

5.3. The Visualization Analysis of Predictions. In time series prediction, it is sometimes more interesting to compare the performance on capturing the so-called extreme events, e.g., an oscillation after a stable growth or decay, or a huge sudden change during oscillations. Thus, a visualization comparing real values and PM predictions is presented in Figure 4. The red dashed lines mark the 10th and 23rd week, whose values follow the highest and lowest real values, respectively.

The prediction of time series extreme events has been proposed by previous studies [46]. We can see that the number of outpatients peaked at 9th week and 10th week, and their outpatient counts significantly dropped. In this situation, PM models with BSI + MF fit the real value better than other methods, which illustrates the ability of weighted layers in summarizing the interactions of exogenous inputs, and the heterogeneous data fusion is an effective method to improve prediction accuracy. The advantage is also retained in the method with MF and BSI. It also should be noted that at 9th week, the number of cases reaches the peak values, and MF effectively improves the accuracy in this timestamp, which makes up for the loss of accuracy of fusion of BSI and historical data in the forecast at that time. Historical cases and MF have slightly fluctuated at 13th week. By incorporating BSI data, the influence of random disturbance in this timestamp is obviously reduced for prediction results. When the value at 23rd week followed the lowest real value, the performance of the PM model with all heterogeneous data fusion is not stable.

Moreover, the predicted value using MF or BSI of 45th week is obviously higher, and the situation is reflected in the fusion of MF and BSI. A possible reason is that the internal random disturbance of exogenous data has not been completely eliminated in the preprocessing stage. Therefore, the extreme weather conditions or special events and degrade the prediction performance.

6. Conclusions

In this paper, we focus on predicting the number of infectious diarrhea outpatient visits in the upcoming week. A parsimonious model (PM) is proposed by condensing previous prediction models. The benchmarks of ten algorithms with four groups of inputs show the advantage of our method. It achieves better prediction performance by consolidating MF data and BSI data.

In the future, we will try to investigate the prediction of other cities and improve the robustness of PM in predicting those cities. Moreover, the simultaneous forecast of multiple cities would be our future research direction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Xiamen Centre for Disease Control and Prevention for sharing the data. This study was supported in part by the Natural Science Foundation of Fujian Province of China (nos. 2018J01539 and 2019J01713) and the Science Project of Xiamen City (no. 2019SH400060).

References

[1] T. Wu, C. Perrings, A. Kinzig, J. P. Collins, B. A. Minteer, and P. Daszak, “Economic growth, urbanization, globalization, and the risks of emerging infectious diseases in China: a review,” *Ambio*, vol. 46, no. 1, pp. 18–29, 2017.

[2] P. Kelly, “Infectious diarrhoea,” *Medicine*, vol. 47, no. 6, pp. 341–346, 2019.

[3] L. Wang, L. Jin, and W. Xiong, “Chapter 2–nfectious diseasesurveillance in China,” in *Early Warning for Infectious DiseaseOutbreak*, W. Yang, Ed., pp. 23–33, Academic Press, Cambridge, MA, USA, 2017.

[4] Y. Lan, D. Zhou, H. Zhang, and S. Lai, “Chapter 3-developmentof early warning models,” in *Early Warning for Infectious Disease Outbreak*, W. Yang, Ed., pp. 35–74, Academic Press, Cambridge, MA, USA, 2017.

[5] I. N. Soyiri and D. D. Reidpath, “An overview of health forecasting,” *Environmental Health and Preventive Medicine*, vol. 18, no. 1, pp. 1–9, 2013.

[6] L. Yu, L. Zhou, L. Tan et al., “Application of multiple seasonal arima model in forecasting incidence of hfmd in Wuhan, China,” *International Journal of Infectious Diseases*, vol. 21, p. 271, 2014.

[7] P. Zhang, S. Shu, and M. Zhou, “An online fault detection model and strategies based on SVM-grid in clouds,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 445–456, 2018.

[8] J. Cai, K. Xu, Y. Zhu, F. Hu, and L. Li, “Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest,” *Applied Energy*, vol. 262, Article ID 114566, 2020.

[9] R. Oppenheim, “Forecasting via the box-jenkins method,” *Journal of the Academy of Marketing Science*, vol. 6, no. 3, pp. 206–221, 1978.

[10] G. Mahalakshmi, S. Sridevi, and S. Rajaram, “A survey on forecasting of time series data,” in *Proceedings of the 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE’16)*, pp. 1–8, Kovilpatti, India, January 2016.

[11] S. Chen, X. Liu, Y. Wu et al., “The application of meteorological data and search index data in improving the prediction of HFMD: a study of two cities in Guangdong province, China,” *Science of The Total Environment*, vol. 652, pp. 1013–1021, 2019.

[12] R. Adhikari and R. K. Agrawal, “An introductory study on time series modeling and forecasting,” 2013, https://arxiv.org/abs/1302.6613.

[13] H. Liu, M. Zhou, and Q. Liu, “An embedded feature selection method for imbalanced data classification,” *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, 2019.

[14] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2019.

[15] E. Diaconescu, “The use of NARX neural networks to predict chaotic time series,” *WSEAS Transactions on Computer Research*, vol. 3, 2008.

[16] R. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural Comput*, vol. 1, pp. 270–280, 1998.

[17] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” 2014, https://arxiv.org/abs/1409.3215.

[18] K. Cho, B. Merrienboer, Y. Bengio et al., “On the properties of neural machine translation: encoder-decoder approaches,” 2014, https://arxiv.org/abs/1409.1259.

[19] Z. C. Lipton, “A critical review of recurrent neural networks for sequence learning,” 2015, https://arxiv.org/abs/1506.00019.

[20] Y. Qin, D. Song, H. Chen et al., “A dual-stage attention-based recurrent neural network for time series prediction,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence IJCAI2017*, pp. 2627–2633, Melbourne, Australia, August 2017.

[21] G. Lai, W. Chang, and Y. Yang, “Modeling long- and short-term temporal patterns with deep neural networks,” in *Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

[22] Y. Chang, F. Sun, Y. H. Wu et al., “A memory-network based solution for multivariate time-series forecasting,” 2018, https://arxiv.org/abs/1809.02105.

[23] Y. Wu, Y. Yang, H. Nishiura et al., “Deep learning for epidemiological predictions,” in *Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval SIGIR2018*, pp. 1085–1088, Ann Arbor, MI, USA, June 2018.

[24] J. Chen, K. Li, P. Herrero et al., “Dilated recurrent neural network for short-time prediction of glucose concentration,” in *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval SIGIR2018*, pp. 95–103, Ann Arbor, MI, USA, June 2018.

[25] S. Shih, F. Sun, and H. Lee, “Temporal pattern attention for multivariate time series forecasting,” *Machine Learning*, vol. 108, no. 8-9, pp. 1421–1441, 2019.

[26] Y. Wang, J. Gu, Z. Zhou, and Z. Wang, “Diarrhoea outpatient visits prediction based on time series decomposition and multi-local predictor fusion,” *Knowledge-Based Systems*, vol. 88, pp. 12–23, 2015.

[27] Y. Wang, J. Li, J. Gu, Z. Zhou, and Z. Wang, “Artificial neural networks for infectious diarrhea prediction using meteorological factors in shanghai (China),” *Applied Soft Computing*, vol. 35, pp. 280–290, 2015.

[28] S. Li, T. Chen, L. Wang, and C. Ming, “Effective tourist volume forecasting supported by PCA and improved BPNN using Baidu index,” *Tourism Management*, vol. 68, pp. 116–126, 2018.

[29] Z. Li, T. Liu, G. Zhu et al., “Dengue Baidu search index data can improve the prediction of local dengue epidemic: a case study in Guangzhou, China,” *PLoS Neglected Tropical Diseases*, vol. 11, no. 3, pp. 1–13, 2017.

[30] Z. Wang, Y. Huang, B. He et al., “TDDF.: HFMD outpatients prediction based on time series decomposition and heterogenous data fusion in Xiamen, China,” in *Proceedings of the 15th International Conference, ADMA 2019*, pp. 658–667, Dalian, China, November 2019.

[31] “Xiamen” from Wikipedia, 2020, https://en.wikipedia.org/wiki/Xiamen.

[32] Xiamen Statistics Bureau, 2018, http://tjj.xm.gov.cn/tjnj/2018/.

[33] K. Liu, T. Wang, Z. Yang et al., “Using Baidu search index to predict dengue outbreak in China,” *Scientific Reports*, vol. 6, p. 38040, 2016.

[34] J. Brownlee, “Long short-term memory networks with python: develop sequence prediction models with deep learning,” Machine Learning Mastery, 2017, https://machinelearningmastery.com.

[35] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2015, https://arxiv.org/abs/1412.6980.

[36] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[37] C. C. R. Sady and A. L. P. Ribeiro, “Symbolic features and classification via support vector machine for predicting death in patients with Chagas disease,” *Computers in Biology and Medicine*, vol. 70, pp. 220–227, 2016.

[38] H. F. Jerome, “Greedy function approximation: a gradient boosting machine,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[39] T. Chen and C. Guestrin, “Xgboost: a scalable tree boosting system,”in *Proceedings of the Proceedings of the 22nd ACM-SIGKDD International Conference on Knowledge Discoveryand Data Mining SIGKDD2016,* San Francisco, CA, USA, August 2016.

[40] Y. Lecun, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[41] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018, https://arxiv.org/abs/1803.01271.

[42] S. Barra, S. M. Carta, A. Corriga, A. S. Podda, and D. R. Recupero, “Deep learning and time series-to-image encoding for financial forecasting,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 683–692, 2020.

[43] M. M. Hassan, A. Gumaei, G. Aloi, G. Fortino, and M. Zhou, “A smartphone-enabled fall detection framework for elderly people in connected home healthcare,” *IEEE Network*, vol. 33, no. 6, pp. 58–63, 2019.

[44] D. Freire-Obregón, F. Narducci, S. Barra, and M. Castrillón-Santana, “Deep learning for source camera identification on mobile devices,” *Pattern Recognition Letters*, vol. 126, pp. 86–91, 2019.

[45] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[46] D. Ding, M. Zhang, X. Pan et al., “Modeling extreme events in time series prediction,” in *Proceedings of the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1114–1122, July 2019.