Received: 30 July 2020 | Revised: 15 November 2020 | Accepted: 20 November 2020

DOI: 10.1002/spe.2940

SPECIAL ISSUE PAPER

WILEY

Dual-grained representation for hand, foot, and mouth disease prediction within public health cyber-physical systems

Zhijin Wang[1] | Yaohui Huang[2] | Bingyan He[1]

[1]Computer Engineering College, Jimei University, Xiamen, China

[2]Chengyi University College, Jimei University, Xiamen, China

Correspondence
Zhijin Wang, Computer Engineering College, Jimei University, Yinjiang Road 185, Xiamen 361021, China.
Email: zhijin@jmu.edu.cn

Funding information
Jimei University, Grant/Award Number: ZP2021013; Natural Science Foundation of Fujian Province, Grant/Award Numbers: 2019J01713, 2020J01700

Abstract

The prediction model is a major component within public health cyber-physical systems, which supports decisions on prevention and control of diseases. Hand, foot, and mouth disease (HFMD) is one of the most common global infectious diseases with the highest incidence rate. Previous HFMD prediction models are mainly based on the time series that counted in equal-grained time intervals. However, there are details in the time series counted in fine-grained time intervals. To benefit from both equal-grained and fine-grained data, we proposed a dual-grained representation (DGR) model. The DGR first represents inputted data to temporal patterns. Then, the represented patterns are consolidated to generate predictions. Experimental comparisons of the short-term prediction performance are figured out by using real outpatient collections in Xiamen, China.

KEYWORDS

dual-grained representation, HFMD prediction, public health cyber-physical systems, real data validation

1 | INTRODUCTION

In the past decade, cyber-physical applications for environment and people monitoring have been reported.[1] Cyber-physical systems (CPS)[2] integrate computational and physical processes, where devices interact with the physical world creating feedback control loops.[3] Recently, CPS are employed to support and promote public health policies,[4] since they enable the realization of these activities in a much more efficient and automatic manner.

Those CPS used for supporting public health management are called public health cyber-physical systems (PH-CPS). The architecture of the centralized PH-CPS is drawn in Figure 1. The PH-CPS consists of four layers: physical layer, communication layer, intelligent service layer, and data analysis layer. The physical layer is designed for devices to interact with patients. It receives information from health devices, and sends information to the communication layer. The communication layer links the physical layer and the intelligent service layer, and makes decisions in situations when intelligent service layer is unavailable. The intelligent service layer manages received information about every single patient, which includes personal information, current health state, diseases history, and health index tendencies. The data analysis layer collects information from different clinics, and uses the received information for further study. Such as, data analysis, data mining, disease discovery, and disease treatments.

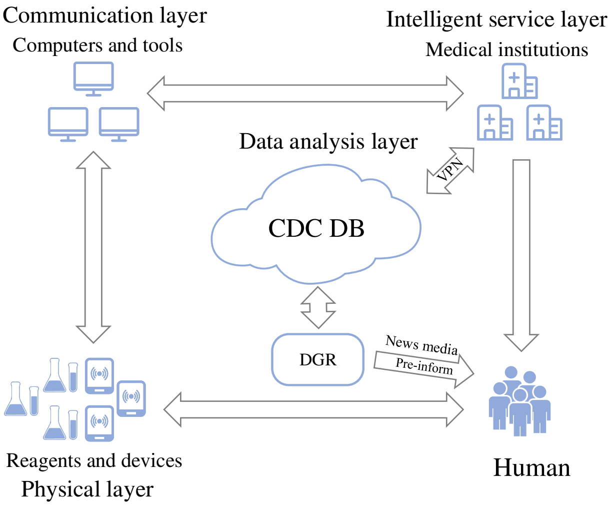


FIGURE 1 The architecture of the centralized cyber-physical systems for public health management [Color figure can be viewed at wileyonlinelibrary.com]

The prediction model is a major component within PH-CPS for forecasting the status of infectious disease. Those models generate predictions based on the outpatient records collected from physical devices in real time.[5] Hand, foot, and mouth disease (HFMD) is the most common global infectious disease.[6] Millions of infections were reported every year.[7] Especially in developing countries, children younger than 5 years is more significant, causing serious economic and social burden.[8] The disease is easy to cause fever, oral ulcers, blisters, and rashes on hands and feet, some serious and potentially fatal complications will lead to serious sequelae and even death.[9] The control and prevention of HFMD is a public health issue that receives attention from government agencies, medical institutions, and the public. China built up and updated the network of the national notifiable disease reporting system to control and prevent the outbreak of infectious diseases. The *Regulations on Preparedness for and Response to Public Health Emergency Events* (NHFPC) had been issued in 2003.[10] HFMD became a Class C notifiable disease in the reporting system on May 2, 2008.[11]

We aim to provide an efficient HFMD prediction model, which works well within PH-CPS. The HFMD prediction techniques have been developed to support the risk management of outbreaks. The HFMD prediction can be viewed as the problem of time series forecasting.[12] Numerous studies have examined the effects of exogenous data in improving the prediction performance.[13] The exogenous data can be climate data, water condition data, and air pollution data. However, the integration of exogenous data from external system into PH-CPS requires a relative large cost than collection of internal data.

We assume that big events in the future are hidden in the small details in the past. Technically, the prediction of disease outbreaks in future weeks can benefit from several past daily outpatient counts. For example, the number of outpatient visit in the upcoming week is predicted by several observed weekly outpatient visit counts and their daily outpatient visit counts. To distinguish the different granularities time series data , if the time interval length for each data point is equal to the interval length of the prediction target, we call them *equal-grained* data, such as weekly counts; if the time interval length for each data point is smaller than the interval length of the prediction target, we call them *fine-grained* data, such as daily counts.

There are two challenges when consolidating the two different granularities to get better predictions: (1) how to represent the temporal dynamics of fine-grained data and find key time intervals which related to the disease outbreak? (2) how to consolidate the temporal dynamics of fine-grained data and equal-grained data to reach better prediction?

To address the above challenges, we propose a dual-grained representation (DGR) model to benefit from fine-grained data and equal-grained data. The proposed method first extracts the temporal dynamics of fine-grained values and equal-grained values using two recurrent neural networks (RNN), and then fuse the extracted features to generate predictions.

The major contributions are summarized as follows:

1. To provide supports on disease control within PH-CPS, we develop a DGR model to generate outpatient cases prediction with better performance.
2. To avoid the cost in integrating exogenous data from external systems into PH-CPS, the DGR solely works with outpatient cases, which are count in different time intervals, that is, daily counts and weekly counts.
3. The extensive experimental results on real HFMD data collections show the effectiveness of the proposed DGR, when compared with state-of-the-art algorithms.

TABLE 1 Notations and semantics

Notation	Semantic
y_t	the outpatient count in equal-grained time interval t
$\boldsymbol{x}_t$	the outpatient count vector in equal-grained time interval t
$x_{t,i}$	the outpatient count in ith fine-grained time interval within equal-grained time interval t
$\hat{y}_{T+1}$	the predicted value in the upcoming week
y^i_{T+1}	the ith predicted value in test set
n	the number of fine-grained series
M	the length of a time series
N	the number of samples in test set
[]	the concatenated operation of elements or vectors
max(·)	the maximum value of a given vector
min(·)	the minimum value of a given vector

TABLE 2 A summary on recent studies with respect to time series prediction

Type of inputs	Category	Subcategory	Method
Univariate input	Stochastic	Linear function	AR, MA, ARMA, ARIMA, SARIMA
		Nonlinear function	Polynomial
	Learning	Linear mapping	MLR
		Nonlinear function	RFR, GBR, SVR, NN
	Decomposition		STL, RobustSTL, STR
Multivariate input	Stochastic	The dual-grained inputs are flatten and fed into UTS methods	
	Learning		
	Deep learning	Basic	CNN1d, RNN, LSTM, GRU
		Deep	Dilated CNN, Dilated RNN, HRHN, LSTNet
		Wide	Encoder–decoder, DA-RNN, TPA-LSTM

Abbreviations: AR, autoregression; ARFIMA, autoregressive fractionally integrated moving average; ARIMA, autoregressive integrated moving average; ARMA, autoregression with moving average; GBR, gradient boosting decision tree regression; GRU, gated recurrent unit; LSTM, long and short term memory; MA, moving average; MLR, multiple linear regression; RFR, random forest regression; RNN, recurrent neural networks; SARIMA, seasonal autoregressive integrated moving average; SVR, support vector machine regression.

The remainder of this article are organized as follows. Section 2 introduces related work. Section 3illustrates our proposed approach. Section 4 gives experimental configurations and comparable methods. Section 5 gives experimental results and analyses. Finally, a conclusion is drawn in Section 6.

The main symbols used are listed at Table 1.

2 | LITERATURE REVIEW

This section introduces existing time series prediction techniques with respect to univariate input[14-34] and multivariate input.[35-51] Table 2 lists related studies on disease time series prediction.

2.1 | Methods based on univariate input

The methods in this category generate predictions, via learning equal-grained historical outpatient cases. These methods are divided into stochastic methods,[14-19] learning methods,[20-29] and decomposition methods.[30-34]

The stochastic method is said to be a linear or nonlinear function from historical observed values to upcoming values. These methods include but not limited to autoregression, moving average, autoregression with moving average,[14] autoregressive integrated moving average (ARIMA),[15] autoregressive fractionally integrated moving average,[16] and seasonal autoregressive integrated moving average. ARIMA model and its different variations are based on the famous Box–Jenkins principle, and so these methods are also broadly known as the Box–Jenkins models. There are also nonlinear methods, such as polynomial regression,[17] cubic spline.[18] In the real environment, most of the time series data are nonstationary and uncertain.[19] But these stochastic methods did not work well with nonstationary series.

Learning methods aim to learn a linear or nonlinear mapping from past observations to upcoming values. According to whether the temporal dynamics of historical observations are considered or not, these methods can be divided into traditional learning methods or temporal concerned learning methods. (a) The traditional learning methods regard a lagged past value as an input dimension, and learns from those values to find mappings. These methods include but not limited to multiple linear regression (MLR),[20] support vector machine regression (SVR),[21] gradient boosting decision tree regression (GBR),[22] random forest regression (RFR),[23] and neural networks.[24] (b) The temporal concerned learning methods consider the inherent temporal dynamics of past values in predicting upcoming values. RNN have been widely applied to infectious disease prediction,[25] for example, long and short term memory (LSTM),[26] and the gated recurrent unit (GRU).[27] In the past decade, due to the easy implementation of deep learning methods, the neural network method is stacked and dilated[28,29] to learning complicated temporal patterns. But, the complex models poorly perform on the small scale disease data. Those methods are easily remember all the trained samples, and lose their generalization abilities in predicting upcoming values.

Decomposition methods regard a time series is the composition of some components.[30,31] Commonly, a time series is decomposed into a trend sequence, a randomized sequence, and several periodic sequences. The STL[32] is a famous decomposition method for time series prediction, where a time series is factorized into a trend series, a seasonality series and a randomized series. The STR[33] explores the joint extraction of trend, seasonality, and residual without iteration. The RobustSTL[34] improves the prediction performance by decomposing long seasonality period and canceling high noises. In reality, not all the data can be well decomposed, since the data uncertainty and data missing problem of infectious disease data.

2.2 | Methods based on multivariate input

To alleviate the problem of data uncertainty, multiple exogenous data sources are collected and fused into the above stochastic methods[35-41] and learning methods.[42-51]

The stochastic method learns a linear or nonlinear function from historical observed target values and exogenous values to upcoming target values. These methods linearly combine past outpatient counts and exogenous data to generate predictions.[35-41] The main differences between them are the regression of target variables, functions on exogenous data and the composition of exogenous data. The climate factors,[35-38] air pollution,[39,40] and search engine query data[38,41] have been found effects on some infectious diseases, and HFMD is included.

In recent years, the learning methods using exogenous data have been intensively studied. These methods can be divided into three parts as below. (a) Traditional learning methods using exogenous inputs. For this subcategory, historical outpatient data and exogenous data are directly concatenated as the input of models. To achieve stable predictions, feature selection and model selection need to be further validated.[42,43] (b) Temporal learning methods using exogenous inputs. For this subcategory, the temporal dynamics of input data are captured using RNN structures, and a nonlinear mapping from inputs to the target is learned from training data. For example, NARX-RNN enhances vanilla RNN using additional exogenous features.[44] To differently treat exogenous inputs and target inputs, the encoder–decoder structure[45] is employed to do time series prediction task. The encoder–decoder framework consists of two RNN layers, and maps input sequence to output sequence.[46,47] (c) Temporal attention learning methods using exogenous inputs. Recently, the attention mechanism is fused into sequential models to forecast upcoming values, for example, TPA-LSTM,[48] DA-RNN,[49] HRHN,[50] and LSTNet.[51] These models have strong memory abilities in keeping numerous samples. Especially for small scale infection data, the training loss value would be very small, but the validation errors would be larger than simple machine learning methods.

However, the exogenous data has the same time span with target data, which misses particulars of historical observations. This article focus on predictions based on observed data and their fine-grained data. For example, using several past weekly observed data and several past daily observed data to predict upcoming HFMD outpatient counts, which refined the association of the information context of each week and reduce the loss of equal-grained data prediction accuracy.

3 | APPROACH

This section presents problem formulation, and illustrates the proposed DGR. The workflow of the proposed DGR is shown in Figure 2.

First, we normalize both fine-grained and equal-grained time series data, see the upper left part in Figure 2. The value range of each time series is quite different to others. Moreover, the RNN work well with data in range [0, 1]. Hence, the time series data are normalized, and this can speed up the training process as well.

Second, we use "one-step forward split" to transform time series data to supervised data, see the upper left part in Figure 2. The time series data cannot be directly fed into a machine learning model. Hence, we do this transformation, and use inputs and outputs of the supervised data to train models.

Third, the supervised data are fed into the proposed DGR, see light blue shade parts in Figure 2. To benefit from both fine-grained data and equal-grained data, we design two representation stages: (1) equal-grained representation stage, (2) fine-grained representation stage. In equal-grained representation stage, a GRU layer and a linear layer are exploited to extract temporal dynamics from equal-grained inputs. The fine-grained representation stage contains consecutive components as follows: a linear layer, a softmax layer, a GRU layer and a linear layer. To maximize the effects from fine-grained data, the DGR uses softmax function to extract key information from fine-grained inputs. The outputs of the model are generated by linearly combining equal-grained representation outputs and fine-grained representation outputs.

Finally, the denormalization from model outputs to predicted values, see the upper right part in Figure 2. To obtain the predictions, we denormalize the model outputs, since the model are trained using normalized data.

3.1 | Problem formulation

A time series is defined as a sequential observed values within consecutive identical interval lengths. In this article, we deal with two kinds of time intervals, for example, week and day. Given the interval length of a prediction target as a reference, if the time interval length for each data point is equal to the interval length of the prediction target, we call them

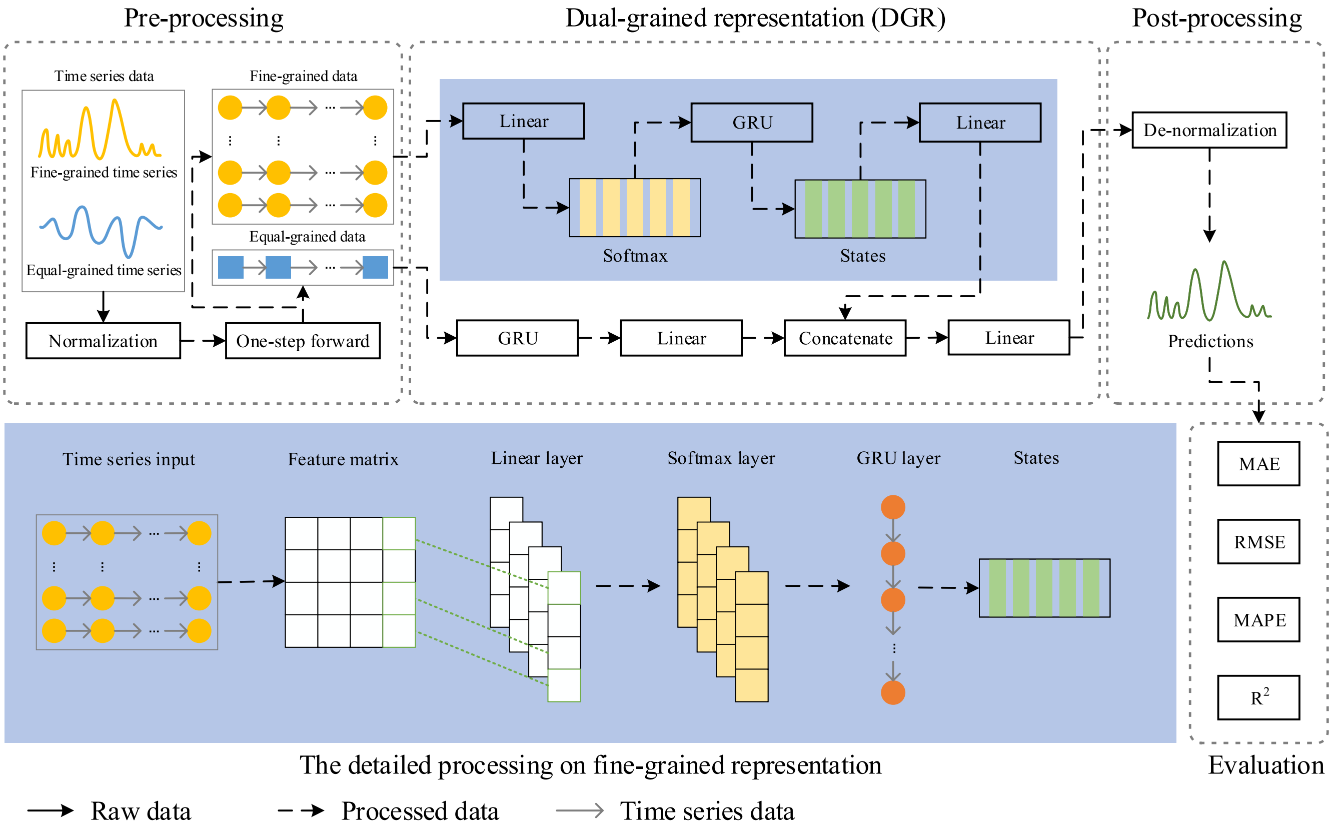


FIGURE 2 The workflow of dual-grained representation [Color figure can be viewed at wileyonlinelibrary.com]

equal-grained data; if the time interval length for each data point is smaller than the interval length of the prediction target, we call them *fine-grained* data. For example, if the prediction target is the outpatient count in the upcoming week, then the past weekly outpatient counts are called equal-grained data, and past daily outpatient counts are called fine-grained data.

Let $y_t \in \mathbb{R}$ denote the observation measured at time interval t, and let $\boldsymbol{x}_t \in \mathbb{R}^n$ denote the fine-grained data points measured within time interval t, where n is the number of fine-grained time intervals. Hence, the equal-grained time series is denoted by $[y_1, y_2, \dots, y_t, \dots, y_M]$, and the fine-grained time series is denoted by $[\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_t, \dots, \boldsymbol{x}_M]$. Furthermore, let T be a time window size. The known fine-grained series with window size T is symbolized as $[\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T]$, as well as the historical target observations as $[y_1, y_2, \dots, y_T]$.

The goal is to predict the value of a future time point $\hat{y}_{T+1}$, given equal-grained outpatient count time series and fine-grained outpatient count time series. Generally, a nonlinear mapping $F(\cdot)$ is applied into the predictive formula:

$$\begin{gathered}\hat{y}_{T+1} = F(y_1, y_2, \dots, y_T, \boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T), \\ \text{s.t.,} \quad y_t = \sum_{i=1}^{n} x_{t,i}.\end{gathered} \tag{1}$$

Moreover, let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T]$ denote the fine-grained data in a time-span of size T, and $\boldsymbol{y} = [y_1, y_2, \dots, y_T]$ be the equal-grained data in a time-span of size T.

3.2 | Dual-grained representation

The proposed DGR model is illustrated using three parts below.

3.2.1 | Data processing

Normalization and denormalization. We normalize the data to range $[0, 1]$. This can also speed up the training process of models. The min–max normalization and zero-score standardization are two most commonly used normalization methods.[52]

The min–max normalization is chosen to compress all the variables into the range $[0, 1]$. Because zero-score standardization requires that the input variable belongs to Gaussian distribution, while min–max normalization is not. The normalization formula and its denormalization formula are as follows:

$$\boldsymbol{d}' = \frac{\boldsymbol{d} - \min(\boldsymbol{d})}{\max(\boldsymbol{d}) - \min(\boldsymbol{d})}, \tag{2}$$

$$\boldsymbol{d} = \boldsymbol{d}' * (\max(\boldsymbol{d}) - \min(\boldsymbol{d})) + \min(\boldsymbol{d}), \tag{3}$$

where $\boldsymbol{d} \in \mathbb{R}^M$ denotes a feature of all the observed samples, M is the number of observed samples, $\boldsymbol{d}'$ is the normalized data, $\max(\boldsymbol{d})$ is the maximum value of $\boldsymbol{d}$, and $\min(\boldsymbol{d})$ is the minimum value of $\boldsymbol{d}$. The denormalization formula is applied for outputs of models in postprocessing stage.

One-step forward split. The time series data cannot be directly fed into a model. Hence, we use one-step forward to split those time series data into inputs and outputs, and use the obtained inputs and outputs data to train models. Given an equal-grained time series $[y_1, y_2, \dots, y_M]$, and several fine-grained time series $[\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M]$, the one-step forward split is formulated as:

$$\begin{bmatrix} y_1 & y_2 & \dots & y_T & \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_T \\ y_2 & y_3 & \dots & y_{T+1} & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \dots & \boldsymbol{x}_{T+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ y_{M-T-1} & y_{M-T} & \dots & y_{M-1} & \boldsymbol{x}_{M-T-1} & \boldsymbol{x}_{M-T} & \dots & \boldsymbol{x}_{M-1} \end{bmatrix} \rightarrow \begin{bmatrix} y_{T+1} \\ y_{T+2} \\ \dots \\ y_M \end{bmatrix}, \tag{4}$$

where the left part is inputs of a model, and the right part is outputs of a model.

3.2.2 | Equal-grained representation

The process of equal-grained representation is shown in the middle-bottom part of DGR within Figure 2. The RNN encodes the input sequences into a features representation in machine translation. Here, we use RNN to encode windowed time series into a feature representation. Given the equal-grained input sequence $\boldsymbol{y} = [y_1, y_2, \dots, y_T]$, the RNN can be applied to learn a mapping from $\boldsymbol{y}$ to $\boldsymbol{h}_k$ with:

$$\boldsymbol{h}_k = f_1(\boldsymbol{h}_{k-1}, \boldsymbol{y}), \tag{5}$$

where $\boldsymbol{h}_k \in \mathbb{R}^m$ is the hidden state of RNN at time k, m is the size of the hidden state, and f_1 is a nonlinear activation function that could be a LSTM or a GRU.

Since the GRU has few parameters and the similar usage as LSTM, we exploit a GRU as f_1 to capture the periodicity and latency of infectious diseases in the long-term trend of the sequence. Each GRU unit has a memory cell with the state $\boldsymbol{s}_t$ at time t. The access of the memory cell will be controlled by two sigmoid gates: reset gate $\boldsymbol{r}_t$ and update gate $\boldsymbol{z}_t$. The update of a GRU unit can be summarized as follows:

$$\boldsymbol{r}_k = \sigma(\boldsymbol{W}_r[\boldsymbol{h}_{k-1}; \boldsymbol{y}]), \tag{6}$$

$$\boldsymbol{z}_k = \sigma(\boldsymbol{W}_z[\boldsymbol{h}_{k-1}; \boldsymbol{y}]), \tag{7}$$

$$\tilde{\boldsymbol{h}}_k = \tanh(\boldsymbol{W}_{\tilde{h}}[\boldsymbol{r}_k * \boldsymbol{h}_{k-1}; \boldsymbol{y}]), \tag{8}$$

$$\boldsymbol{h}_k = (1 - \boldsymbol{z}_k) * \boldsymbol{h}_{k-1} + \boldsymbol{z}_k * \tilde{\boldsymbol{h}}_k, \tag{9}$$

$$\boldsymbol{o}_k = \sigma(\boldsymbol{W}_o \boldsymbol{h}_k), \tag{10}$$

where [;] indicates a concatenated vector, * indicates the element-wise multiplication, and σ, tanh are the activation functions used in this structure in order to keep the information flowing through the GRU within a specific range. $\boldsymbol{W}_r$, $\boldsymbol{W}_z$, $\boldsymbol{W}_{\tilde{h}}$, $\boldsymbol{W}_o \in \mathbb{R}^{m \times (m+n)}$ is the weight that is optimized during training, and $\boldsymbol{r}_k$, $\boldsymbol{z}_k \in \mathbb{R}^n$ are parameters to learn. The feature space is represented by $[\boldsymbol{h}_{k-1}; \boldsymbol{a}_k]$ and the prediction is represented by $\boldsymbol{o}_k$. The key reason for using GRU is that it overcomes the problem of vanishing gradients, better capture temporal dynamics of equal-grained time series, and few learned parameters.

After the feature representation of equal-grained data, the hidden state is linearly combined to generate outputs. The generated value from represented equal-grained data is formulated as:

$$c = \boldsymbol{w}_c \boldsymbol{h}_k + b_c, \tag{11}$$

where c is the consolidated output of fine-grained data side, $\boldsymbol{w}_c \in \mathbb{R}^m$ is a weight parameter needs to be learned, and b_c is a bias term, and $\boldsymbol{h}_k$ is the hidden state which is represented from equal-grained data.

3.2.3 | Fine-grained representation

The fine-grained inputs include much sequential fluctuation details and short-term incidence patterns, which are significant to short-term prediction. The process of fine-grained representation is shown in the shadow part within Figure 2.

To distinguish the fine-grained inputs, we adopt a linear layer to process the input of fine-grained data, and get the transformed outputs from the linear layer. The transformation is formulated as follows:

$$\boldsymbol{F} = \boldsymbol{W}_f * \boldsymbol{X} + b_f, \tag{12}$$

where $\boldsymbol{F} \in \mathbb{R}^{n \times T}$ is the output of linear transformation, * indicates the element-wise multiplication, $\boldsymbol{W}_f \in \mathbb{R}^{n \times T}$ is the weight corresponding to the input dimension, and b_f is a bias term.

Based on the transformed fine-grained inputs, the softmax function is exploited to highlight the periodic events, for example, every Saturday of a week. The softmax layer is formulated as:

$$
\begin{aligned}
\boldsymbol{P} &= \text{softmax}(F), \\
\text{softmax}(p_{i,t}) &= \frac{e^{f_{i,t}}}{\sum_{j=1}^{T} e^{f_{i,j}}}, \\
\text{s.t.,} \quad & p_{i,t} \in P, f_{i,t} \in F,
\end{aligned} \tag{13}
$$

where P is the output of the softmax layer. The key reason for using the softmax layer is that the periodic events can be highlighted. Once the recent periodic trends of a time series is precisely captured, the accuracy of short-term prediction be improved.

We adopt another RNN layer to represent the temporal dynamic of the highlighted outputs, and get states from the RNN outputs. The mapping is formulated as follows:

$$\boldsymbol{h'}_k = f_2(\boldsymbol{h'}_{k-1}, \boldsymbol{P}), \tag{14}$$

where $\boldsymbol{h'}_k \in \mathbb{R}^m$ is the hidden state of RNN at time k, m is the size of the hidden state, and f_2 is a nonlinear activation function that could be a LSTM or a GRU. To make the model simpler, another GRU is selected to f_2, and is update the same as Equations (6)–(10).

After the feature representation of fine-grained data, the hidden state is linearly combined to generate output. The generated value from represented fine-grained data is formulated as:

$$g = \boldsymbol{w}_g \boldsymbol{h'}_k + b_g, \tag{15}$$

where g is the consolidated output of fine-grained data side, $\boldsymbol{w}_g \in \mathbb{R}^m$ is learned parameters, b_g is a bias term, and $\boldsymbol{h'}_k$ is the hidden state which is represented from equal-grained data.

Finally, the outputs from two stages (i.e., Equations (11) and (15)) are combined using a fully connected layer, the formula is as follows:

$$\hat{y}_{T+1} = \sigma(\boldsymbol{w}_m[c; g] + b_m), \tag{16}$$

where $[c_t; c_y] \in \mathbb{R}^2$ is the concatenated vector of outputs, $\boldsymbol{w}_m \in \mathbb{R}^2$ is the weight of those outputs, $\hat{y}_{T+1}$ is the predicted value of the number of outpatients in the next week, and b_m is a bias term.

4 | EXPERIMENTAL SETUP

This section gives data collections, evaluation metrics, and comparable methods.

4.1 | Data collection and configuration

As Figure 3 plotted, a total of 261 weeks data were collected to develop the time series models. The HFMD outpatient visit counts are obtained from the Chinese Center for Disease Control and Prevention, All individual-level data are anonymized.

The numbers of HFMD cases from January, 2012 to December, 2015 are used to train models, and the rest data are used to evaluate the predictive performance of models. Reminder that the number of HFMD outpatient counts are in a time sequence. Hence, the front 80% data is used to train models and the rest as test set. The windows size T is varied from 1 to 20, to find the best performance of models.

Basic statistical characteristics of fine-grained data and HFMD outpatient counts are listed at Table 3.

All neural models are trained using the Adam optimizer.[53] 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 $\{32, 64\}$.

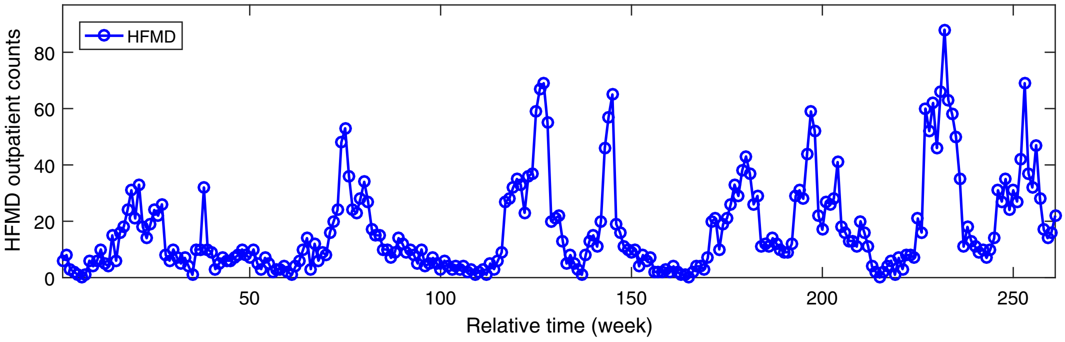


FIGURE 3 The distribution of weekly outpatient visit counts from January 1, 2012 to December 31, 2016 [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Basic statistical characteristics of fine-grained time series and HFMD counts (261 samples)

		Range				
Symbols	**Parameter (Unit)**	**Min**	**Max**	**Mean**	**Median**	**STD**
H_7	the daily number of outpatients on Sunday (count)	0	88	16.98	10	16.47
H_1	the daily number of outpatients on Monday (count)	0	82	17.48	12	16.07
H_2	the daily number of outpatients on Tuesday (count)	0	80	16.59	11	15.54
H_3	the daily number of outpatients on Wednesday (count)	0	73	15.73	10	15.15
H_4	the daily number of outpatients on Thursday (count)	0	69	15.25	10	14.89
H_5	the daily number of outpatients on Friday (count)	0	68	15.20	10	13.87
H_6	the daily number of outpatients on Saturday (count)	0	72	15.81	11	14.57
C	the weekly number of outpatients (count)	2	435	113.04	79	100.76

Abbreviation: HFMD, hand, foot, and mouth disease; STD, standard deviation.

4.2 | Evaluation metrics

The performance evaluation metrics are used to observe the overall efficiency of the prediction of time series data. For forecasting task or analysis task of time series data, there are many error metrics, such as mean absolute error (MAE), normalized mean absolute error, mean absolute percentage error (MAPE), MSE, root mean square error (RMSE), and correlation coefficient (R^2). We choose some metrics, since some of them are logically equivalent. For example, the RMSE value can be obtained once the MSE value is figured out, and vice versa.

The evaluation criteria are combined with MAE, RMSE, MAPE, and R^2. These criteria can be expressed in the following mathematical expressions:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}(|y_{T+1}^{i} - \hat{y}_{T+1}^{i}|), \tag{17}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_{T+1}^{i} - \hat{y}_{T+1}^{i})^{2}}, \tag{18}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{(|y_{T+1}^{i} - \hat{y}_{T+1}^{i}|)}{|y_{T+1}^{i}|} \times 100\%, \tag{19}$$

TABLE 4 The comparable methods used for the competitions on two kinds of inputs

Comparable method	Name	Weekly data	Daily + weekly data
MLR[47]	Multiple linear regression	Yes	Yes
SVR[21]	Support vector machine regression	Yes	Yes
GBR[22]	Gradient boosting regression	Yes	Yes
RFR[23]	Random forest regression	Yes	Yes
Xgboost[54]	Extreme Gradient Boosting regression	Yes	Yes
RNN[46]	Recurrent neural network	Yes	Yes
LSTM[26]	Long and short term memory	Yes	Yes
DA-RNN[49]	Dual-attention RNN		Yes

Abbreviations: GBR, gradient boosting decision tree regression; LSTM, long and short term memory; MLR, multiple linear regression; RFR, random forest regression; RNN, recurrent neural networks; SVR, support vector machine regression.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_{T+1}^{i} - \hat{y}_{T+1}^{i}\right)^2}{\sum_{i=1}^{N} {y_{T+1}^{i}}^2}. \tag{20}$$

In the above equations, y_{T+1}^{i} is the ith actual value in the test period, $\hat{y}_{T+1}^{i}$ is the ith predicted value, N is the length of the testing period. The performance with the smallest MAE, RMSE, and MAPE and the largest R^2 are considered to be the best model.

4.3 | Comparable methods

The comparable methods are listed at Table 4. The descriptions on these methods are given as follows.

MLR[47] is widely used for modeling the linear relationship between the input variables and the target variable. The advantage is that the parameters do not need to be tuned.

SVR[21] can efficiently perform a nonlinear regression using the kernel functions, which implicitly mapping inputs into high-dimensional feature spaces. Specifies the kernel type and penalty parameter to build the prediction model.

GBR[22] produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

RFR[23] is an ensemble learning method for regression via learning decision trees. The number of trees and the minimum number of samples are tuned, to adjust the model.

Xgboost[54] has gained much popularity and attention recently, which append regularization term into loss function, and it is an improvement on the gradient boosting. Models can be adjusted by the number of trees, learning rate, and maximum tree depth.

RNN[46] exhibits temporal dynamic states of the sequence by their internal state. The hidden neurons are used to control the complexity of RNN.

LSTM[26] is a kind of RNN, which is composed of a cell, an input gate, an output gate and a forget gate.

DA-RNN[49] is based on the encoder–decoder framework, which exploits input attention and temporal attention mechanisms to process exogenous data.

5 | RESULTS AND ANALYSES

This section conducts several experiments to investigates the effects on prediction performance, and visualizes the correlations between fine-grained inputs and equal-inputs. These experiments and analyses focus on the following questions:

1. How parameter T affects the prediction performance? Intuitively, how the past observed values affect the prediction on the upcoming value?

2. What is the degree of performance improvement via fine-grained representation? In detail, could the DGR outperform other comparable methods? Could the prediction performance of DGR be improved by consolidating the fine-grained representation? Could the performance of comparable methods be improved by directly input fine-grained data and equal-grained data?
3. How the hidden size of GRU affects the prediction performance?
4. Why the prediction performance can benefit from fine-grained data? In other words, what is the insights between fine-grained data and equal-grain data?

5.1 | Effects on parameter *T*

To investigate the effects on parameter T of the proposed DGR, we measure the performance of DGR by varying parameter T. To investigate the effects on the hidden size of GRU component, we measure the performance of DGR by varying the hidden size in 32, 64. The performance in terms of MAE, RMSE, MAPE, and R^2 is plotted Figure 4. We summarize the major observation from these results as follows:

1. The optimal values of MAE, RMSE, and R^2 are found $T = 13$, respectively.
2. The optimal MAPE value are found at $T = 14$.
3. The trend of performance are degraded when T is greater or less than 13.
4. The DGR(32) is more robust than DGR(64) when varies window size T. The best performance and worst performance are found at DGR(64).

As plotted in Figure 4(A,B,D), the DGR(32) and DGR(64) obtain the optimal values of MAE, RMSE, R^2, and second best MAPE values at $T = 13$, respectively. When the window size T is set to 14, the MAPE value achieves the optimal value. In reality, $T = 13$ means that the upcoming values have relations with past 13 weeks values. The duration of 13 weeks is equivalent to one quarter. This suggests that the outbreak period of HFMD infection is one quarter. Moreover, spring and autumn are the peak periods of outbreaks.

As shown in Figure 4(C), the optimal MAPE values are found at $T = 14$. Compared with MAE, MAPE is the percentage of the error to the real values, and its value is the average absolute error over the testing set. MAPE is insensitive to prediction errors when the actual value is small.

It can be observed that the performance is degraded when the T is set smaller or greater than 13. The major reason is that the past observed values is strongly related to the future values. Few inputs of past observations are easy to cause insufficient information while training a model. Numerous inputs of past observations would interference the mapping from past values to future values.

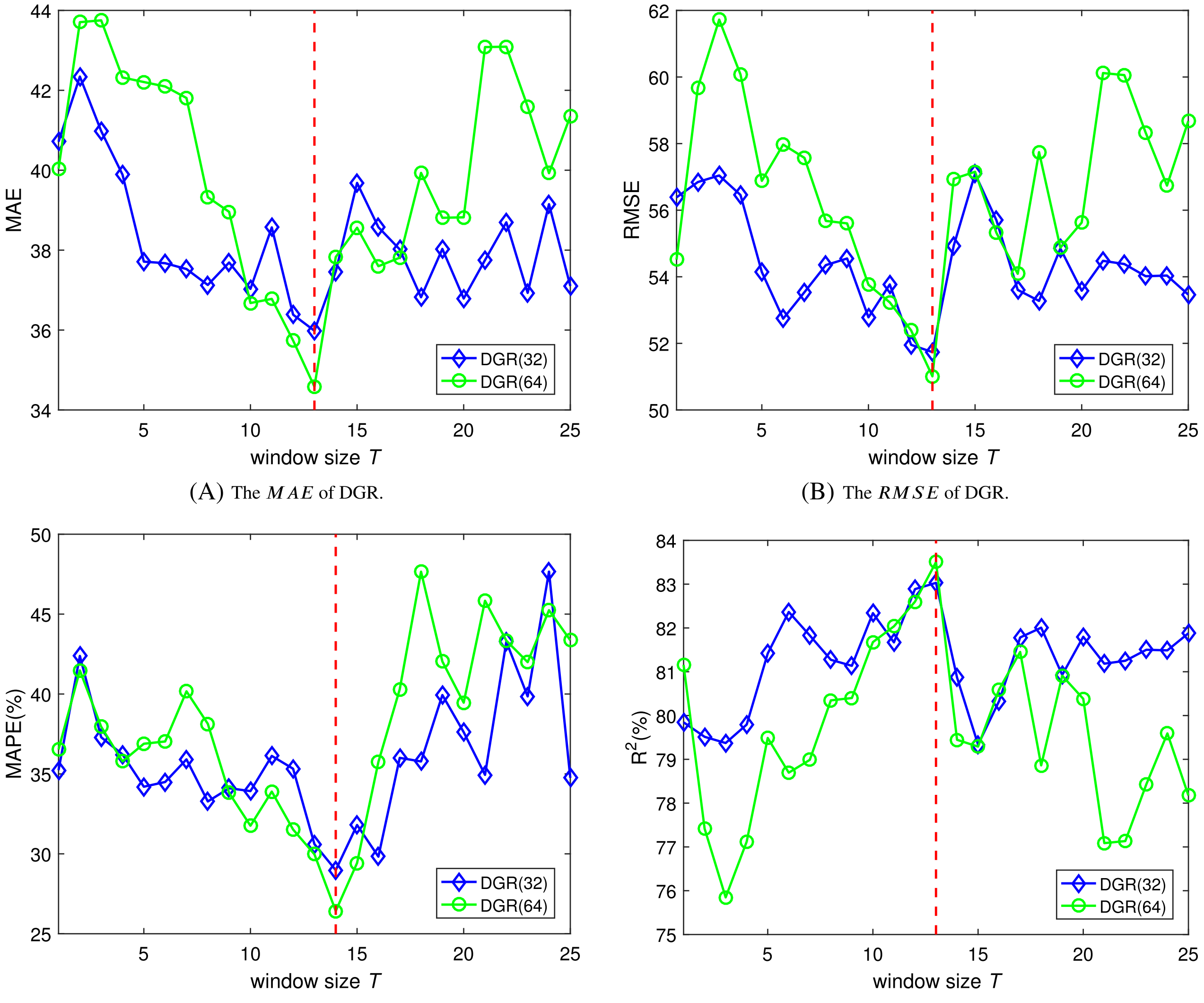
The DGR with 64 hidden neurons gets the best result at $T = 13$. The DGR with 32 hidden neurons outperforms the DGR with 64 hidden neurons in terms of model stability and overall performance. The models with many neurons are susceptible to random disturbances, and require more training samples to reach a converged status. Therefore, the prediction performance shows unstable oscillations as T gradually increases.

5.2 | Performance comparison

The optimal MAE, RMSE, MAPE, and R^2 values for other models are found around $T = 13$. For fair comparison, we compare all the models by fixing $T = 13$. As presented in Figure 5, all the comparable methods are well-tuned, and their performance is measured in terms of MAE, RMSE, MAPE, and R^2.

According to the prediction results solely based on weekly data, see the blue bars in Figure 5, DGR and RNN methods have the best prediction results, and SVR and MLR have the second best performance. Actually, the DGR method can be regarded as a simple GRU method by removing the fine-grained representation component. This reveals that the RNN methods and the linear methods are effective in forecasting, when the data scale of training set is small, such as hundreds of samples. Noted that all tree-based methods get unstable performance. When training a small scale data, the tree-based method is difficult to eliminate the random disturbance of a time series. Hence, the prediction accuracy of tree-based methods are degraded.

According to the results of predictions based on weekly and daily data, see green bars in Figure 5, the performance of traditional machine learning methods generally declines when compared with predictions based on weekly data.



(A) The MAE of DGR.

(B) The $RMSE$ of DGR.

(C) The $MAPE$ of DGR.

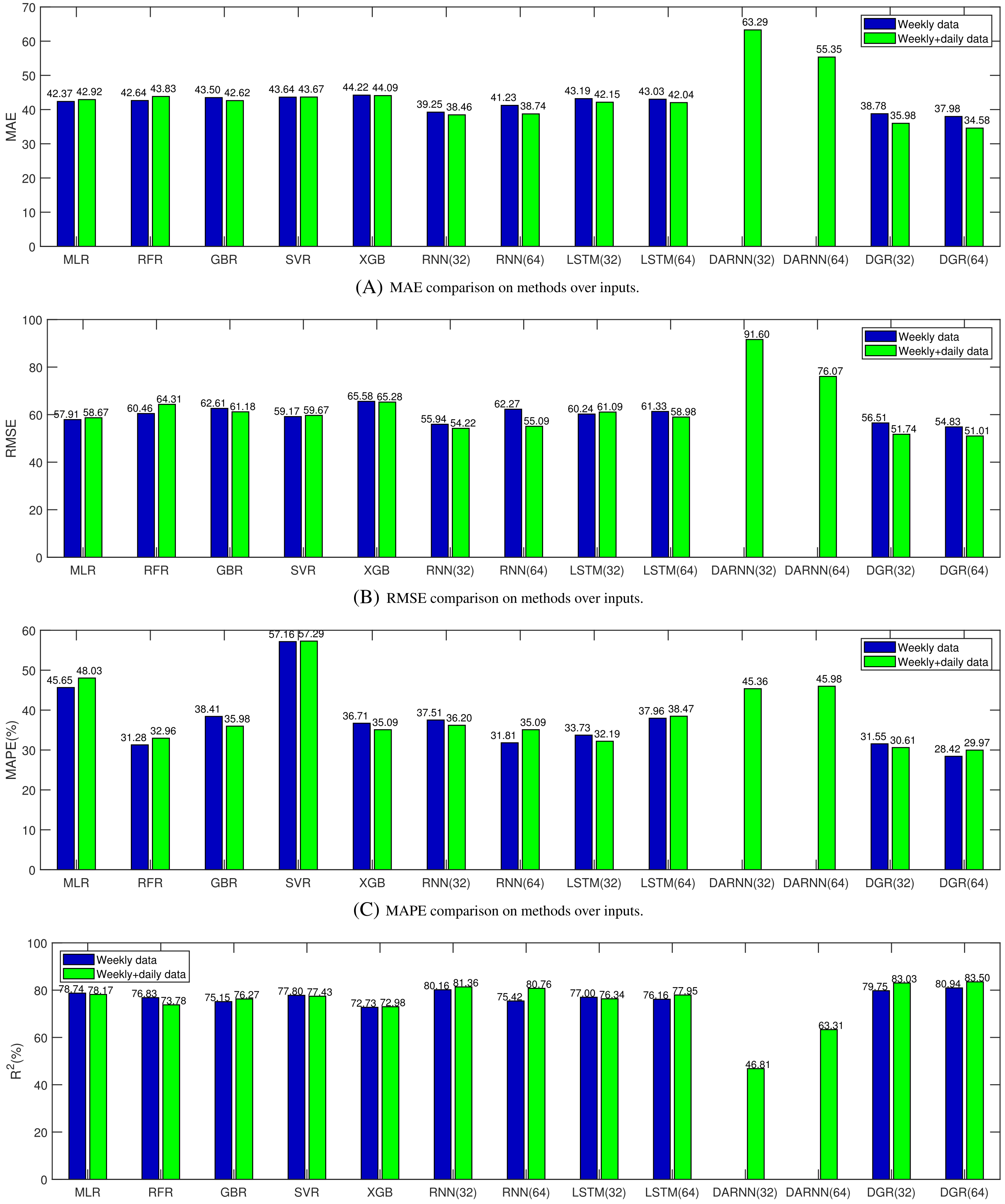
(D) The R^2 of DGR with varying T.

FIGURE 4 The performance of DGR with varying window size T in terms of MAE, RMSE, MAPE, and R^2. For each metric, the optimal values are found at red dash line. DGR, dual-grained representation; MAE, mean absolute error; MAPE, mean absolute percentage error; RMSE, root mean square error [Color figure can be viewed at wileyonlinelibrary.com]

However, the prediction performance based on deep learning methods, especially the RNN method has been significantly improved, which means that fine-grained data can effectively improve the model performance. Traditional machine learning methods cannot extract sufficient input information or feature details from small-scale input data to predict the target variable. The DA-RNN method shows the worst prediction performance. The potential reason is the complex hybrid neural network remembers too much trained information, which would lead to a poor performance in dealing with unknown inputs.

Compared with LSTM, the RNN and GRU methods have fewer parameters and therefore are easier to converge. In the case of the small scale inputs, the predict performance of RNN and GRU is better, and the effect is better than LSTM. Vanilla RNN cannot effectively extract long-sequence information and solve the problem of sequence dependence. The GRU model achieves effective control of historical time information through the gating unit, thereby better capturing the long-term dependence of the sequence. It should be noted that the GRU model has the best MAPE value. The main reason is that the MAPE index is sensitive to the magnitude of the real value, and the GRU model has a good fitting performance for the time period with a large real value, but the overall prediction result of total samples still fluctuates and oscillates, so MAE and RMSE values are not ideal.

The proposed DGR achieves the best results under both input conditions. There are two potential reasons for the excellent performance of the DGR method. First of all, this article adds a GRU unit to extract the time dynamics of the



(A) MAE comparison on methods over inputs.

(B) RMSE comparison on methods over inputs.

(C) MAPE comparison on methods over inputs.

(D) R^2 comparison on methods over inputs.

FIGURE 5 Comparison of 13 methods on two inputs in terms of four metrics. The windows size T is fixed at 13. For MAE, RMSE, and MAPE, the lower value of is the better performance. For R^2, the higher value is the better performance. MAE, mean absolute error; MAPE, mean absolute percentage error; RMSE, root mean square error [Color figure can be viewed at wileyonlinelibrary.com]

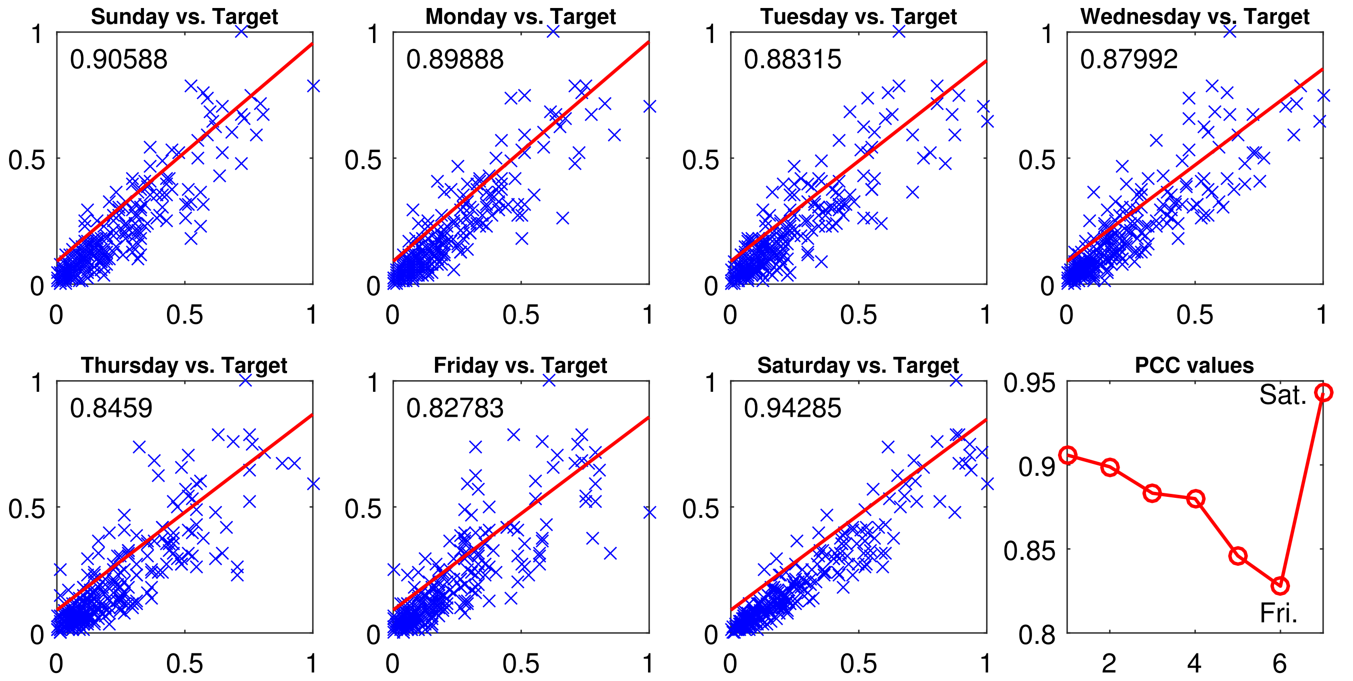


FIGURE 6 The visualization of correlations between the target variable and the fine-grained variables [Color figure can be viewed at wileyonlinelibrary.com]

time series of outpatients, which is effective for establishing connections between different time points. Second, for the input of daily data, this article uses another GRU structure to establish the relationship between the target variable and the fine-grained inputs, mapping the fine-grained dynamic changes to the equal-grained dynamic changes is a new method to reduce the loss of equal-grained prediction accuracy.

5.3 | Correlation analysis

To investigate the impact of fine-grained data, we visualize the correlations between equal-grained inputs and fine-grained inputs.

As shown in Figure 6, the target variable (equal-grained data) is significantly correlated with fine-grained variables. Their Pearson correlation coefficient values are measured and plotted in the Northwest of first seven subfigures, respectively. The comparison of those seven correlations is plotted in the last subfigure. The infected outpatient counts in Friday are weakest correlated to outpatient counts in the whole week, but the counts in Saturday are strongest correlated to the whole week's counts. The potential reasons are as below: (a) social activities are mostly on Friday; (b) people tend to check their health statuses on weekends, even if they get sick on weekdays, especially on Saturday.

Nevertheless, the significant correlated fine-grained data is overlooked. This article discovered the possibility of optimizing the time series prediction problem by leveraging fine-grained characteristics.

6 | CONCLUSIONS

This article focuses on the prediction of HFMD outpatient visit counts in upcoming week, by leveraging past weekly counts and daily counts within PH-CPS. We proposed DGR to benefit from both weekly data and daily data. The DGR first extracts temporal patterns from dual-grained time series data, by exploiting two designed time series processing components. Intensive experiments on the real HFMD data collection reveals the effectiveness of our proposed method. Technically, DGR discovers and consolidates fine-grained data into equal-grained data in predicting future values.

In the future, the multihorizon prediction will be further studied, and the simultaneous prediction of multiple infectious diseases as well.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of Fujian Province of China (nos. 2019J01713 and 2020J01700) and the National Fund Cultivation Program of Jimei University (no. ZP2021013). We would like to thank Xiamen Center for Disease Control and Prevention for sharing the data. The authors would like to thank the editor and anonymous reviewers for their helpful comments in improving the quality of this article.

ORCID

Zhijin Wang https://orcid.org/0000-0002-7962-2827
Yaohui Huang https://orcid.org/0000-0002-3437-1120

REFERENCES

1. Bordel B, Alcarria R, Sánchez-Picot Á, Rivera DDS. Cyber-physical systems for environment and people monitoring in large facilities: a study case in public health. Paper presented at: Proceedings of the International Conference on Information Technology and Systems, Quito, Ecuador; 2019:406-416.
2. Schirner G, Erdogmus D, Chowdhury KR, Padir T. The future of human-in-the-loop cyber-physical systems. *IEEE Comput*. 2013;46(1):36-45.
3. Zhou J, Sun J, Zhang M, Ma Y. Dependable scheduling for real-time workflows on cyber-physical cloud systems. *IEEE Trans Ind Inform*. 2020. https://doi.org/10.1109/TII.2020.3011506. 1–1. https://ieeexplore.ieee.org/abstract/document/9146769.
4. Bordel B, Alcarria R, Robles T, Martín D. Cyber-physical systems: extending pervasive sensing from control theory to the Internet of Things. *Pervasive Mob Comput*. 2017;40:156-184. https://doi.org/10.1016/j.pmcj.2017.06.011.
5. Zhou J, Sun J, Cong P, et al. Security-critical energy-aware task scheduling for heterogeneous real-time MPSoCs in IoT. *IEEE Trans Serv Comput*. 2019;13(4):745-758. https://doi.org/10.1109/TSC.2019.2963301.
6. Jun Sun B, Jie Chen H, Chen Y, Dong An X, Sen Zhou B. The risk factors of acquiring severe hand, foot, and mouth disease: a meta-analysis. *Canad J Infect Dis Med Microbiol*. 2018;2018:1-12. https://www.hindawi.com/journals/cjidmm/2018/2751457/.
7. Ji T, Han T, Tan X, et al. Surveillance, epidemiology, and pathogen spectrum of hand, foot, and mouth disease in mainland of China from 2008 to 2017. *Biosafety Health*. 2019.1(1):32–40.
8. Yang S, Wu J, Ding C, et al. Epidemiological features of and changes in incidence of infectious diseases in China in the first decade after the SARS outbreak: an observational trend study. *Lancet Infect Dis*. 2017;17(7):716-725.
9. Ooi MH, Wong SC, Mohan A, et al. Identification and validation of clinical predictors for the risk of neurological involvement in children with hand, foot, and mouth disease in Sarawak. *BMC Infect Dis*. 2009;9(1):3.
10. Public health emergency events emergency regulations; 2005. http://www.nhfpc.gov.cn/yjb/s3580/200804/b41369aac27847dba3e6aebccc72e2%f8.shtml/chn.
11. WHO representative office China; 2008. http://www.wpro.who.int/china/mediacentre/factsheets/hfmd/en/.
12. Mahalakshmi G, Sridevi S, Rajaram S. A survey on forecasting of time series data. Paper presented: Proceedings of the International Conference on Computing Technologies and Intelligent Data Engineering. Kovilpatti, Tamilnadu, India: IEEE; 2016:1-8.
13. Wang Z, Huang Y, He B, et al. Short-term infectious diarrhea prediction using weather and search data in Xiamen, China. *Sci Program*. 2020;2020:1-12. https://doi.org/10.1155/2020/8814222.
14. Zhang B, Chan JC, Cross JL. Stochastic volatility models with ARMA innovations: an application to G7 inflation forecasts. *Int J Forecast*. 2020;36(4):1318-1328.
15. George EP, Box GCR, Ljung GM. Time series analysis: forecasting and control. *J Time Ser Anal*. 2015;37:712.
16. Adhikari R, Agrawal RK. *An Introductory Study on Time Series Modeling and Forecasting*. arXiv preprint. 2013. arXiv:1302.6613.
17. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Paper presented at: Proceedings of the 33rd International Conference on Advances in Neural Information Processing Systems. Barcelona, Spain: Curran Associates, Inc; 2016:3844-3852.
18. Balestriero R, Cosentino R, Glotin H, Baraniuk R. Spline filters for end-to-end deep learning. Paper presented at: Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden; 2018:364-373.
19. Safaei M, Ismail AS, Chizari H, et al. Standalone noise and anomaly detection in wireless sensor networks: a novel time-series and adaptive Bayesian-network-based approach. *Softw Pract Exp*. 2020;50(4):428-446. https://doi.org/10.1002/spe.2785.
20. Bai J, Perron P. Estimating and testing linear models with multiple structural changes. *Econometrica*. 1998;66:47-78. https://doi.org/10.2307/2998540.
21. Sady CC, Ribeiro ALP. Symbolic features and classification via support vector machine for predicting death in patients with Chagas disease. *Comput Biol Med*. 2016;70:220-227.
22. Friedman J. Greedy function approximation: a gradient boosting machine. *Ann Stat*. 2001;29:1189–1232.
23. Yuchi W, Gombojav E, Boldbaatar B, et al. Evaluation of random forest regression and multiple linear regression for predicting indoor fine particulate matter concentrations in a highly polluted city. *Environ Pollut*. 2019;245:746-753.
24. Wang Y, Gu J. Comparative study among three different artificial neural networks to infectious diarrhea forecasting. Paper presented at: Proceedings of International Conference on Bioinformatics and Biomedicine, Belfast,UK; 2014:40-46.

25. Wu Y, Yang Y, Nishiura H, Saitoh M. Deep learning for epidemiological predictions. Paper presented at: Proceedings of the 41st International Conference on Research Development in Information Retrieval, Ann Arbor, MI; 2018:1085-1088.
26. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780.
27. Cho K, Merrienboer VB, Gülçehre Ç, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Paper presented at: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar; 2014:1724-1734.
28. Zhang Y, Yang X, Ivy JS, Chi M. ATTAIN: attention-based time-aware LSTM networks for disease progression modeling. Paper presented at: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, SAR, China; 2019:4369-4375.
29. Ke H, Chen D, Shah T, et al. Cloud-aided online EEG classification system for brain healthcare: a case study of depression evaluation with a lightweight CNN. *Softw Pract Exp*. 2020;50(5):596-610. https://doi.org/10.1002/spe.2668.
30. Wang Y, Gu J, Zhou Z, Wang Z. Diarrhoea outpatient visits prediction based on time series decomposition and multi-local predictor fusion. *Knowl-Based Syst*. 2015;88:12-23.
31. Wang Z, He L. User identification for enhancing IP-TV recommendation. *Knowl-Based Syst*. 2016;98:68-75.
32. Cleveland R, Cleveland W, McRae J, Terpenning I. STL: a seasonal-trend decomposition procedure based on loess. *J Off Stat*. 1990;6:3-33.
33. Dokumentov A, Hyndman RJ. STR: a seasonal-trend decomposition procedure based on regression. 2020; arXiv preprint. arXiv:2009.05894
34. Wen Q, Gao J, Song X, Sun L, Xu H, Zhu S. RobustSTL: a robust seasonal-trend decomposition algorithm for long time series. Paper presented at: Proceedings of the 33rd International Conference on Artificial Intelligence, Honolulu, Hawaii; 2019:5409-5416.
35. Song C, Shi X, Bo Y, Wang J, Wang Y, Huang D. Exploring spatiotemporal nonstationary effects of climate factors on hand, foot, and mouth disease using Bayesian spatiotemporally varying coefficients (STVC) model in Sichuan, China. *Sci Total Environ*. 2019;648:550-560.
36. Yang Y, You E, Wu J, et al. Effects of relative humidity on childhood hand, foot, and mouth disease reinfection in Hefei, China. *Sci Total Environ*. 2018;630:820-826.
37. Wang J, Xiao Y, Cheke RA. Modelling the effects of contaminated environments on HFMD infections in mainland China. *Biosystems*. 2016;140:1-7.
38. Huang DC, Wang JF. Monitoring hand, foot and mouth disease by combining search engine query data and meteorological factors. *Sci Total Environ*. 2018;612:1293-1299. https://doi.org/10.1016/j.scitotenv.2017.09.017.
39. Du Z, Lawrence WR, Zhang W, Zhang D, Yu S, Hao Y. Interactions between climate factors and air pollution on daily HFMD cases: a time series study in Guangdong, China. *Sci Total Environ*. 2019;656:1358-1364.
40. Yang Z, Hao J, Huang S, et al. Acute effects of air pollution on the incidence of hand, foot, and mouth disease in Wuhan, China. *Atmos Environ*. 2020;225:r117358.
41. Chen S, Liu X, Wu Y, 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. *Sci Total Environ*. 2019;652:1013-1021.
42. Wang Z, Huang Y, He B, Luo T, Wang Y, Lin Y. TDDF: HFMD outpatients prediction based on time series decomposition and heterogenous data fusion in Xiamen, China. Paper presented at: Proceedings of the 15th International Conference Advanced Data Mining and Applications, Dalian, China; 2019:658-667.
43. Liu K, Wang T, Yang Z, et al. Using baidu search index to predict dengue outbreak in China. *Sci Rep*. 2016;6:38040.
44. Diaconescu E. The use of NARX neural networks to predict chaotic time series. *WSEAS Trans Comput Res*. 2008;3:182–191.
45. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Paper presented at: Proceedings of Annual Conference on Neural Information Processing Systems, Montreal, Quebec, Canada; 2014:3104-3112.
46. Lipton ZC. A critical review of recurrent neural networks for sequence learning; 2015. arXiv preprint arXiv:1506.00019.
47. Cho K, Merrienboer VB, Bahdanau D, Bengio Y. On the properties of neural machine translation: encoder-decoder approaches. Paper presented at: Proceedings of 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar.
48. Shih S, Sun F, Lee H. Temporal pattern attention for multivariate time series forecasting. *Mach Learn*. 2019;108(8-9):1421-1441.
49. Qin Y, Song D, Chen H, Cheng W, Jiang G, Cottrell GW. A dual-stage attention-based recurrent neural network for time series prediction. Paper presented at: Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, VIC, Australia; 2017:2627-2633.
50. Tao Y, Ma L, Zhang W, Liu J, Liu W, Du Q. Hierarchical attention-based recurrent highway networks for time series prediction; 2018. arXiv preprint arXiv:1806.00685.
51. Lai G, Chang W, Yang Y, Liu H. Modeling long- and short-term temporal patterns with deep neural networks. Paper presented at: Proceedings of the 41st International Conference on Research Development in Information Retrieval, Ann Arbor, MI; 2018:95-104.
52. Jain S, Shukla S, Wadhvani R. Dynamic selection of normalization techniques using data complexity measures. *Expert Syst Appl*. 2018;106:252-262.
53. Kingma DP, Ba J. Adam: a method for stochastic optimization; 2015. arXiv preprint arXiv:1412.6980.
54. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. Paper presented at: Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA; 2016:785-794.

How to cite this article: Wang Z, Huang Y, He B. Dual-grained representation for hand, foot, and mouth disease prediction within public health cyber-physical systems. *Softw Pract Exper*. 2020;1–16. https://doi.org/10.1002/spe.2940