Applied Intelligence
https://doi.org/10.1007/s10489-021-02391-6

COVID-19 cases prediction in multiple areas via shapelet learning

Zhijin Wang[1] · **Bing Cai**[1]

Accepted: 25 March 2021

Abstract
Predicting the number of COVID-19 cases in a geographical area is important for the management of health resources and decision making. Several methods have been proposed for COVID-19 case predictions but they have important limitations in terms of model interpretability, related to COVID-19's incubation period and major trends of disease transmission. To be able to explain prediction results in terms of incubation period and transmission trends, this paper presents the Multivariate Shapelet Learning (MSL) model to learn shapelets from historical observations in multiple areas. An experimental evaluation was done to compare the prediction performance of eleven algorithms, using the data collected from 50 US provinces/states. Results show that the proposed method is effective and efficient. The learned shapelets explain increasing and decreasing trends of new confirmed cases, and reveal that the COVID-19 incubation period in the USA is around 28 days.

Keywords COVID-19 · Prediction · Multivariate · Shapelet learning · Interpretability

1 Introduction

The Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) caused a pandemic around the world, and became a global threat in the past months [1]. It's also known as the Corona Virus Disease 2019, and in short of COVID-19. The COVID-19 has caused 7,720,830 active cases, and produced 1,013,992 deaths till September 31, 2020. China reported the pandemic in Wuhan city, in late December 2019. World Health Organization reported the pandemic on June 7, 2020.

Governments and authorities have been struggling to make critical decisions. The COVID-19 prediction is a novel area, and an important tool for predicting future events or situations, such as, allocation of medical supplies and dispatch of medical staff. The prediction tools generate predictions of the spread of the virus, and support decisions, such as preventive medicine and healthcare intervention strategies.

Several methods have been done to predict the number of COVID-19 cases. These methods build a model for each area, and ignore the interconnections among areas. Meanwhile, their interpretability is poor, such as COVID-19's incubation period [2] and transmission trends.

Our goal is to develop a high interpretable model [3] to predict upcoming COVID-19 cases in multiple areas. The model should tackle the two issues as follows: (1) the determination of COVID-19's incubation period among geographically connected areas, such as provinces/states in a country. (2) the obtainment of key transmission trends in multiple connected areas.

Inspired by the interpretability of shapelets [4], we exploit shapelets to represent key trends of the COVID-19 time series in multiple areas, and use the shapelet length to denote the incubation period length. The shapelets are introduced to enhance classifiers [5]. They are defined as discriminative subsequences of several time series, which belong to a common class. In the past decade, shapelet studies mainly focus on improving the classifier performance in terms of accuracy and speed. Can shapelets be representative subsequences of several interrelated time series? Can shapelets predict the upcoming values of those time series as well?

We propose a model named "Multivariate Shapelet Learning (MSL)" to efficiently solve the two questions. The MSL consists of four procedures as follows. (1) To degrade

This article belongs to the Topical Collection: *Artificial Intelligence Applications for COVID-19, Detection, Control, Prediction, and Diagnosis*

✉ Zhijin Wang
zhijin@jmu.edu.cn

Bing Cai
ccailearning@gmail.com

[1] Computer Engineering College, Jimei University, Yinjiang Road 185, Xiamen 361021, China

Published online: 07 May 2021

the search complexity, we adopt the one-step-forward split to window multivariate time series. Another benefit of the one-step-forward split is the transformation from time series data to supervised data, which means the MSL can be trained. (2) We design a shapelet layer to store the shapelets. (3) To learn and keep shapelets in the shapelet layer, we use the softmin distance to measure the distance between a time series and a shapelet. (4) A linear layer is adopted to connect the softmin distances and model outputs. Therefore, good shapelets can be found by minimizing the gap between model outputs and real observations.

The major contributions of this paper are summarized below.

(1) The determination of COVID-19's incubation period in geographically connected areas.
(2) The obtainment of trends on COVID-19 transmission in geographically connected areas. These trends are visualized by the learned shapelets, which show the high interpretability of MSL.
(3) The MSL model is proposed to simultaneously predict the upcoming new COVID-19 cases with better performance.

The rest of this paper is organized as follows. Section 2 addresses this research. Section 3 gives formulations and notations. Section 4 illustrates our proposed method. Section 5 gives descriptions of the COVID-19 data, performance criteria and experimental configurations. Section 6 analyses evaluated results, visualizes and the learned shapelets. Finally, a conclusion is drawn in Section 7.

2 Related work

This section addresses this research via reviewing recent studies.

2.1 COVID-19 prediction

According to the nature of methods, we categorize COVID-19 prediction methods into parsimonious methods and mathematical methods.

Parsimonious methods fit a linear or non-linear function from training data and show positive effects on the early prediction of the pandemic [6]. [7] uses ARIMA models and polynomial functions to predict daily cumulative COVID-19 cases in 145 countries, where each country has a tuned model. [8] uses ARIMA models to predict the daily cumulative confirmed cases in 3 European countries. [9] uses ARIMA models to predict the daily new confirmed cases for the 7-day period.

Mathematical methods model epidemic situations to enhance predictions. [10] applies mathematical models to describe the outbreak among passengers and crew members on Princess Cruises Ship. [11] realizes forward prediction and backward inference of the epidemic situation. [12] introduces a Composite Monte Carlo method to predict daily new confirmed COVID-19 cases, which is enhanced by deep learning and fuzzy rule induction. [13] exploits epidemic propagation model to predict daily cumulative confirmed cases of five worst affected states in India.

The above methods build a model for each time series. Hence, the interrelationship of these time series are ignored, such as the transmission among geographical areas. Moreover, not only parsimonious methods but also mathematical methods can not interpret their models, such as the determination of incubation periods, and the key trends of disease transmission. In this paper, an interpretable model using shapelet learning is proposed to predict COVID-19 cases in several interconnected areas.

2.2 Shapelet learning

The shapelets concept is first introduced by [4] for data mining. Its original definition is "subsequences that are in some sense maximally representative of a class". The shapelet has high interpretability and good explanations. But it is still a challenge to efficiently find good shapelets.

According to the way of shapelet obtainment, these methods are divided into two categories: (1) shapelets mining, which optimizes the procedures on searching the optimal time series segment, such as brute force searching [14] and tree-based pruning [15]; (2) shapelet learning, which learns several shapelets by optimizing a classification loss function, such as LTS [5] and FastLTS [16]. As a key component in learning shapelets, the distance measurement between a shapelet and a segment is studied as well [17]. These methods are developed to deal with time series classification tasks. Their target is to learn discriminative shapelets from each inputted time series in a class, i.e., a classification task [18].

However, we learn representative shapelets from multiple time series, and generate predictions for each time series, i.e., a regression task. These shapelets have a probability of being similar or common segments. We adopt the shapelets concept to describe the key data points of the observed time series. Meanwhile, we use the one-step-ahead split method to segment multiple time series. Based on this split method, the shapelets are learned under linear complexity.

3 Formulations and notations

This section gives formulations and notations. The main symbols used are listed in Table 1.

Table 1 Symbols and semantics

Symbol	Semantic
I	area number
K	time step number
C	shapelet number
N	time step number in test set
T	look-back window size
$\boldsymbol{Z}$	outpatient cases matrix, $\boldsymbol{Z} \in \mathbb{R}^{I\times K}$
$\boldsymbol{S}$	shapelet matrix, $\boldsymbol{S} \in \mathbb{R}^{C\times T}$
$\boldsymbol{W}$	weight matrix of shapelets, $\boldsymbol{W} \in \mathbb{R}^{C}$
$\boldsymbol{D}$	distance matrix, $\boldsymbol{D} \in \mathbb{R}^{I\times C}$
$D_{i,c}$	element of distance matrix $\boldsymbol{D}$
$\boldsymbol{M}$	softmin distance matrix, $\boldsymbol{M} \in \mathbb{R}^{I\times C}$
$M_{i,c}$	element of softmin distance matrix $\boldsymbol{M}$
$\boldsymbol{X}$	input matrix $\boldsymbol{X} \in \mathbb{R}^{I\times T}$
$\boldsymbol{Y}$	output matrix $\boldsymbol{Y} \in \mathbb{R}^{1\times T}$
$\boldsymbol{\mathcal{X}}$	inputs $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{(K-T+1)\times I\times T}$
$\boldsymbol{\mathcal{Y}}$	outputs $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{(K-T+1)\times 1\times T}$

Multivariate time series (MTS) We adopt a MTS to describe the observed daily new confirmed cases in multiple states/provinces in the America. The symbol $\boldsymbol{Z} \in \mathbb{R}^{I\times K}$ denotes the cases of I areas in K consecutive days.

Look-back window A look-back window of size T is an ordered sub-sequence of a MTS. and is exploited to observe the cases in a certain period. We use symbol $\boldsymbol{Z}_{:,t+1:t+T} \in \mathbb{R}^{I\times T}$ to denote a look-back window.

Shapelet. A shapelet of size T is an ordered sequence of values, and is employed to represent the key data points of a series. To represent key data points of a MTS, we need several candidate shapelets. These shapelets is denoted by $\boldsymbol{S} \in \mathbb{R}^{C\times T}$. C is the number of shapelets.

Distances between shapelets and MTS The distance measurement is a critical step to learn shapelets from a MTS. The distance between a time series and a shapelet is defined as:

$$D_{i,c} \coloneqq \min_{t} \frac{1}{T}\sum_{k=1}^{T}(Z_{i,t+k} - S_{c,k})^2, \tag{1}$$

where $D_{i,c}$ is the distance between i-th time series and c-th shapelet, and $\boldsymbol{D} \in \mathbb{R}^{I\times C}$ is the total distances between candidate shapelet $\boldsymbol{S}$ and observed $\boldsymbol{Z}$.

MTS prediction problem Typically, the MTS prediction problem is formulated as:

$$\hat{\boldsymbol{Z}}_{:,T+1} = F(\boldsymbol{Z}_{:,1:T}), \tag{2}$$

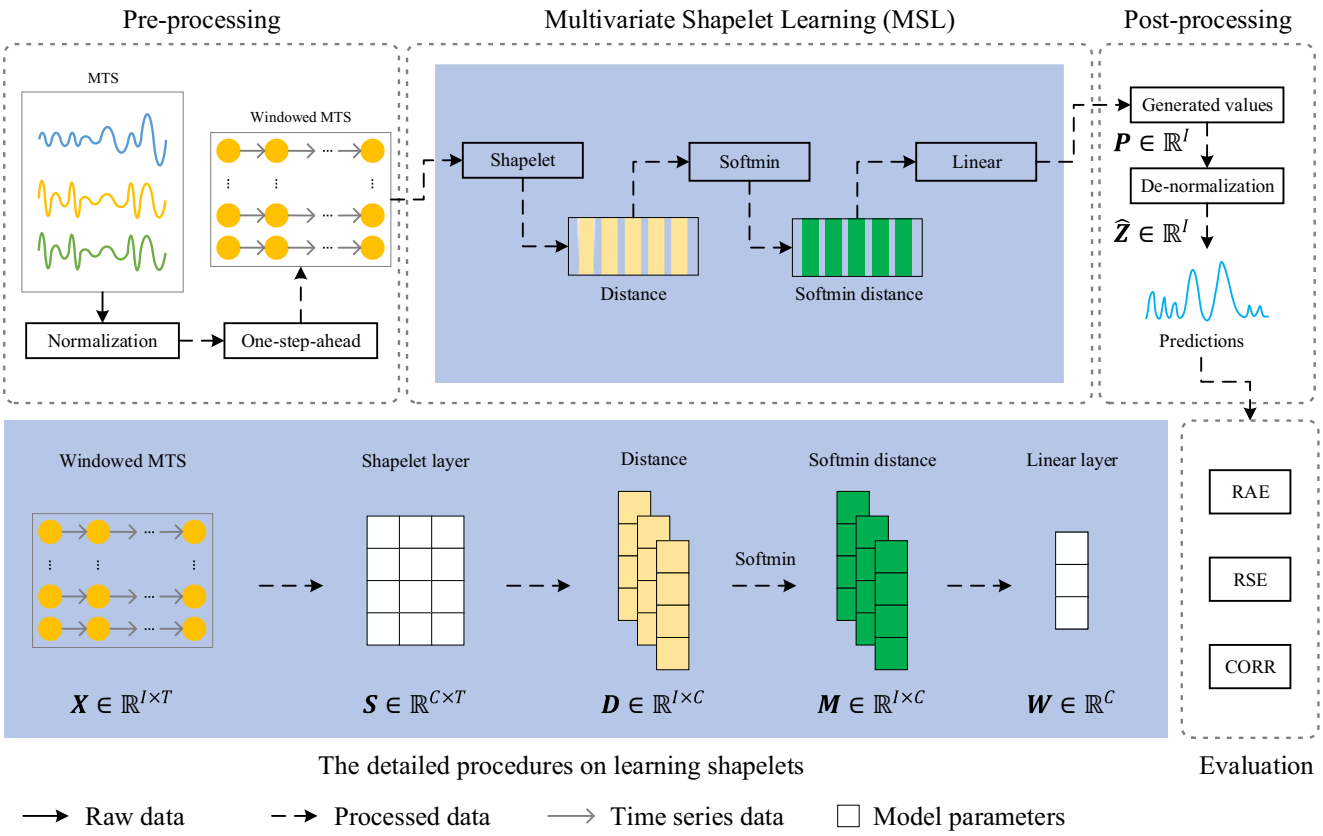

Fig. 1 The schematic illustration of Multivariate Shapelet Learning (MSL)

Algorithm 1: Pseudo code for training MSL.

Input: Observed cases $\boldsymbol{Z} \in \mathbb{R}^{I \times K}$, window size T and shapelet number C
Output: Shapelets $\boldsymbol{S} \in \mathbb{R}^{C \times T}$, weights $\boldsymbol{W} \in \mathbb{R}^{C}$

$Z' \leftarrow$ normalize $\boldsymbol{Z}$ using (4);
$\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \leftarrow$ split Z' using (6);
// Feed forward and backward gradient updating
foreach *sample* $(\boldsymbol{X}, \boldsymbol{Y})$ *in* $(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}})$ **do**
  // $\boldsymbol{D} \leftarrow$ Distances between $\boldsymbol{X}$ and $\boldsymbol{S}$
  for $i \leftarrow 1$ **to** I **do**
    for $c \leftarrow 1$ **to** C **do**
      $\boldsymbol{D}_{i,c} \leftarrow \boldsymbol{X}_{i,:}$ and $\boldsymbol{S}_{c,:}$ using (7);
    end
  end
  $\boldsymbol{M} \leftarrow$ softmin $\boldsymbol{D}$ using (8);
  $\hat{\boldsymbol{P}} \leftarrow \boldsymbol{M}$ and $\boldsymbol{W}$ using (9);
  Loss $L \leftarrow \boldsymbol{Y}$ and $\hat{\boldsymbol{P}}$ using MSE;
  Backward using Adam [19];
  $\hat{\boldsymbol{Z}} \leftarrow$ de-normalize $\hat{\boldsymbol{P}}$ using (5);
end
return $\boldsymbol{S}$, $\boldsymbol{W}$

where $\hat{\boldsymbol{Z}}_{:,T+1} \in \mathbb{R}^{I}$ is the predicted values of I areas in the upcoming day, $\boldsymbol{Z}_{:,1:T} = [\boldsymbol{Z}_{:,1}, \boldsymbol{Z}_{:,2}, \cdots, \boldsymbol{Z}_{:,T}]$ is observations over a look-back window of size T, and $F(\cdot)$ is the mapping.

The problem of MTS prediction via shapelet learning is formulated as:

$$\hat{\boldsymbol{Z}}_{:,T+1} = F(\boldsymbol{Z}_{:,1:T}, \boldsymbol{S}), \tag{3}$$

where $\boldsymbol{S} \in \mathbb{R}^{C \times T}$ is the learned shapelets.

4 The MSL model

This section illustrates the proposed MSL. The diagram of the proposed MSL is graphically displayed in Fig. 1.

Firstly, the normalization of MTS data, see the upper left part in Fig. 1. Because there are significant differences in the range of confirmed cases data in different regions, we normalize those data into [0, 1]. The normalized data can also speed up the training process of models.

Secondly, the transformation from MTS data to supervised data, see the upper left part in Fig. 1. Due to the MTS data can not be directly fed into a model, we use one-step-ahead to split MTS data into supervised data, and use the supervised data to train models.

Thirdly, the shapelet learning stage, see light blue shade parts in Fig. 1. There are two tasks in this stage: (1) obtainment of key data points, i.e., shapelets. (2) accurate predictions of future values; For the first task, we designed a shapelet layer, a distance layer, and a softmin layer to learn parameters that are close or similar to import inputs. For the second task, we add a linear layer to receive the minimum distances and generate predictions.

Finally, the de-normalization from model outputs to predicted values, see the upper right part in Fig. 1. To obtain the predictions, we de-normalize the model outputs, since the model are trained using normalized data.

The pseudo code for training MSL is shown in Algorithm 1.

4.1 Normalization and time series transformation

Min-Max normalization The Min-Max normalization is chosen to compress all the variables into the range [0, 1]. The normalization formula and its de-normalization formula are as follows:

$$\boldsymbol{d}' = \frac{\boldsymbol{d} - \min(\boldsymbol{d})}{\max(\boldsymbol{d}) - \min(\boldsymbol{d})}, \tag{4}$$

$$\boldsymbol{d} = \boldsymbol{d}' * (\max(\boldsymbol{d}) - \min(\boldsymbol{d})) + \min(\boldsymbol{d}), \tag{5}$$

where $\boldsymbol{d} \in \mathbb{R}^{K}$ denotes a vector of all the observed samples, M is the number of observed samples, $\boldsymbol{d}' \in \mathbb{R}^{K}$ 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 de-normalization formula is applied for outputs of models in the post-processing stage.

One-step-ahead split Given a MTS $\boldsymbol{Z}$ with K consecutive time intervals, the one-step-ahead split is formulated as:

$$\begin{bmatrix} \boldsymbol{Z}_{:,1} & \boldsymbol{Z}_{:,2} & \cdots & \boldsymbol{Z}_{:,T} \\ \boldsymbol{Z}_{:,2} & \boldsymbol{Z}_{:,3} & \cdots & \boldsymbol{Z}_{:,T+1} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{Z}_{:,K-T-1} & \boldsymbol{Z}_{:,K-T} & \cdots & \boldsymbol{Z}_{:,K-1} \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{Z}_{:,T+1} \\ \boldsymbol{Z}_{:,T+2} \\ \cdots \\ \boldsymbol{Z}_{:,K} \end{bmatrix}, \tag{6}$$

where the left part is inputs of a model, a.k.a, windowed MTS, the right part is the output of a model. For lucid presentation, let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{(K-T-1) \times I \times T}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{(K-T-1) \times 1 \times T}$ denote inputs and outputs, respectively. Each sample in $(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}})$ is denoted by $(\boldsymbol{X} \in \mathbb{R}^{I \times T}, \boldsymbol{Y} \in \mathbb{R}^{1 \times T})$.

4.2 Shapelet learning and prediction generation

Shapelet layer The shapelet layer receives the input $\boldsymbol{X}$, and passes the values of shapelets to the subsequent distance layer.

Let symbol C denote the number of shapelets. The shapelets can be denoted by $S \in \mathbb{R}^{C \times T}$. From the data structure perspective, a shapelet of length T is an ordered sequence of values [5].

The shapelets are parameters, and designed to approximate the key data points of input data, which can help

predicting future values. The approximation of shapelets from historical observations consists of two steps. Firstly, we calculate the distances between all inputted windowed time series and shapelets. Secondly, we use a softmin layer to figure out the minimum distance.

Distance layer This layer receives inputs and shapelets from prior layers. To figure out the minimum distance between inputs and shapelets, the distances of inputs and shapelets should be calculated.

The distances between inputs and shapelets are formulated as:

$$D_{i,c} = \frac{1}{T}\sum_{t=1}^{T}(X_{i,t} - S_{c,t})^2, \tag{7}$$

where $D_{i,c}$ denotes the distance between the i-th windowed time series and the c-th shapelet, $X_{i,t}$ is the COVID-19 case value in the i-th area at the t-th inputted time step, and $S_{c,t}$ is the shapelet value of the c-th shapelet at the t-th time step.

Other distance measurements can be applied to calculate the distances between inputs and shapelets as well, such as Euclidean distance. The reason we choose mean squared error (MSE) is that, the loss function of the MSL is also MSE.

Softmin layer This layer receives distances, applies softmin function to those distances, and the softmin distances are delivered to the subsequent linear layer.

The minimum distance is employed to search a target shapelet, which is the closest or most similar to the inputs. We adopt softmin function to select the closest shapelet, and generate predictions based on the shapelet, via inputting it into a linear layer. The softmin distance is defined as:

$$M_{i,c} = \frac{D_{i,c}\exp(\alpha D_{i,c})}{\sum_{c'=1}^{C}\exp(\alpha D_{i,c'})}, \tag{8}$$

where α is a constant parameter, and $M_{i,c}$ is the softmin distance. The α should be a small negative value, since a big value would cause numeric overflow. According to our experience, we set α to -10.

Linear layer This layer linearly combines the softmin distances, and generates values for post-processing. The combination of the softmin distances is formulated as follows:

$$\hat{\boldsymbol{P}} = \boldsymbol{M}_{:,c} \cdot \boldsymbol{W}, \tag{9}$$

where $\hat{\boldsymbol{P}} \in \mathbb{R}^{1\times T}$ is the model outputs, and $\boldsymbol{W} \in \mathbb{R}^{C}$ is the shared weights assigned to received softmin distances.

5 Experimental configuration

This section gives evaluation metrics and comparable methods.

5.1 Data collection

The COVID-19 data collections are publicly available and daily updated on the GitHub website[1]. The basic statistics of COVID-19 cases on provinces/states in the US are listed in Table 2.

The duration of the collected data ranges from January 22, 2020 to September 17, 2020. Tens of thousands of COVID-19 infected people were newly confirmed in a day. Those colonized regions of America are removed. The confirmed cases in 50 provinces or states are counted in days. We organize those statistics into two groups: cumulative confirmed cases and new confirmed cases. The statistics consist of minimum value, maximum value, mean value and standard deviation.

The infected statuses of California, Florida and Texas are the most serious in America, and the new confirmed cases in these states commonly increase more than 10,000 in recent days. Whereas, the statuses of Vermont, Wyoming, Maine and Alaska are moderate, and the total number of infected persons are all less than 7,000.

STD denotes standard deviation, which reflects the degree of dispersion of a set of data points. When observing the STD values of new confirmed cases of each province/state, (1) California, Florida and Texas all exceed 200,000, which means serious outbreaks; (2) but for Vermont, Wyoming, Maine and Alaska, their infectious statuses are relatively stable.

When observing the maximum values of new confirmed cases, (1) the cases of California, Florida and Texas are all about 1500, which means the situation has always been threats; (2) but the cases of Vermont and Maine are all less than 100, which shows their situations are under control. The main reason for these phenomenons is these states have a large population. Another potential reason is government control and policies.

This paper aims to predict upcoming new COVID-19 cases for these 50 provinces/states. Moreover, the COVID-19's incubation periods of the US and major transmission trends should be learned from the above observations. To create a model with high interpretability, we learn core shapelets from past observations.

[1] https://github.com/datasets/COVID-19

Table 2 The basic statistics of COVID-19 cases on provinces/states in US

Province/State	Cumulative confirmed cases						New confirmed cases					
	Jan. 22	Sep. 17	Min	Max	Mean	STD	Jan. 22	Sep. 17	Min	Max	Mean	STD
Alabama	0	141757	0	141757	36947.39	45718.81	0	670	0	2399	593.13	629.13
Alaska	0	6537	0	6537	1370.08	1880.15	0	105	0	186	27.35	35.81
Arizona	0	211660	0	211660	64232.76	79516.87	0	1753	0	4877	885.61	1179.11
Arkansas	0	73211	0	73211	18151.74	22659.75	0	992	0	1799	308.03	327.04
California	0	775037	0	775037	214083.88	256530.43	0	3716	0	15117	3242.83	3390.47
Colorado	0	63125	0	63125	23762.43	20826.01	0	459	0	978	264.12	196.09
Connecticut	0	55386	0	55386	28919.85	21610.45	0	220	0	2109	231.85	338.79
Delaware	0	19318	0	19318	7655.05	6635.96	0	84	0	458	81.01	82.31
Florida	0	674456	0	674456	179394.29	233791.23	0	3255	0	15300	2821.99	3582.07
Georgia	0	300903	0	300903	82195.76	96708.81	0	1847	0	4813	1259.01	1264.64
Hawaii	0	11105	0	11105	1791.38	2878.14	0	159	0	354	46.49	79.93
Idaho	0	36489	0	36489	8638.55	11587.30	0	396	0	729	152.68	194.31
Illinois	0	270294	0	270294	97540.25	86778.57	0	2056	0	5594	1130.94	903.57
Indiana	0	108646	0	108646	34472.90	33179.01	0	837	0	1660	454.59	347.82
Iowa	0	77204	0	77204	21994.40	23189.08	0	552	0	2681	323.03	323.78
Kansas	0	51164	0	51164	13198.72	14992.99	0	444	0	1019	214.08	233.71
Kentucky	0	59370	0	59370	14637.71	17108.50	0	606	0	1152	248.42	271.10
Louisiana	0	159304	0	159304	52856.44	53069.35	0	478	0	3840	667.04	778.09
Maine	0	4962	0	4962	2014.03	1720.29	0	21	0	78	20.77	17.34
Maryland	0	118519	0	118519	44839.28	40592.53	0	631	0	1784	495.91	376.92
Massachusetts	0	126128	0	129182	67834.98	51331.95	0	429	0	4973	559.68	699.48
Michigan	0	126722	0	126722	51045.24	40587.95	0	980	0	1991	530.22	451.32
Minnesota	0	86722	0	86722	26087.94	27645.40	0	909	0	1154	362.85	307.00

Table 2 (continued)

Province/State	Cumulative confirmed cases						New confirmed cases					
	Jan. 22	Sep. 17	Min	Max	Mean	STD	Jan. 22	Sep. 17	Min	Max	Mean	STD
Mississippi	0	91935	0	91935	25341.64	29785.48	0	701	0	1775	384.67	409.11
Missouri	0	109557	0	109557	24191.67	30221.85	0	1704	0	2197	458.40	531.13
Montana	0	9647	0	9647	1832.43	2657.63	0	216	0	221	40.37	56.03
Nebraska	0	39921	0	39921	12990.62	12755.38	0	502	0	727	167.22	151.74
Nevada	0	74595	0	74595	19988.43	24952.28	0	347	0	1447	312.55	358.78
New Hampshire	0	7781	0	7781	3507.97	2887.02	0	0	0	217	32.82	35.41
New Jersey	0	198361	0	198361	109793.85	78471.83	0	569	0	4305	830.09	1123.26
New Mexico	0	27199	0	27199	9169.31	9371.14	0	158	0	460	113.80	97.31
New York	0	447262	0	447262	258941.11	176566.31	0	896	0	11434	1871.39	2717.55
North Carolina	0	189576	0	189576	52557.93	61548.51	0	1552	0	2603	793.21	725.05
North Dakota	0	16723	0	16723	3468.78	4219.70	0	390	0	467	70.12	91.57
Ohio	0	141585	0	141585	42668.13	44362.01	0	1067	0	1733	592.41	475.22
Oklahoma	0	73318	0	73318	16100.90	21232.70	0	1034	0	1400	306.77	358.16
Oregon	0	30060	0	30060	8186.89	9562.41	0	210	0	430	125.77	123.57
Pennsylvania	0	152775	0	152775	62616.12	51557.84	0	925	0	2297	639.23	487.05
Rhode Island	0	23488	0	23488	10640.30	8551.32	0	130	0	648	98.28	122.57
South Carolina	0	135446	0	135446	35195.04	44694.39	0	1324	0	2454	566.76	635.48
South Dakota	0	17686	0	17686	4863.82	4784.92	0	395	0	623	74.00	86.92
Tennessee	0	178140	0	178140	45448.44	55820.45	0	1053	0	3314	745.36	804.80
Texas	0	701350	0	701350	178059.59	231505.99	0	4543	0	14962	2934.52	3349.69
Utah	0	60658	0	60658	17292.81	19452.15	0	911	0	954	253.80	229.25
Vermont	0	1705	0	1705	840.71	590.89	0	3	0	72	7.14	10.12
Virginia	0	137367	0	137367	44273.90	44122.85	0	1098	0	2015	574.76	432.00
Washington	1	81198	1	81198	27132.35	26221.48	1	386	0	1738	339.90	293.07
West Virginia	0	13434	0	13434	3128.39	3716.86	0	232	0	351	56.23	65.37
Wisconsin	0	93819	0	93819	24643.87	27612.16	0	1660	0	1660	392.55	376.29
Wyoming	0	4652	0	4652	1269.30	1369.22	0	86	0	126	19.53	21.39

"STD" denotes standard deviation

5.2 Metrics

Many evaluation metrics can be applied to measure the performance of MTS prediction. For a fair competition, we follow the metrics in [20–22]. The three metrics are formulated as follows:

– Relative Absolute Error (RAE):

$$RAE = \frac{\sqrt{\sum_{(i,t)\in\Omega_{Test}} |\mathbf{Z}_{i,t} - \hat{\mathbf{Z}}_{i,t}|}}{\sqrt{\sum_{(i,t)\in\Omega_{Test}} |\mathbf{Z}_{i,t} - mean(\hat{\mathbf{Z}}_{i,:})|}}, \tag{10}$$

– Relative Squared Error (RSE):

$$RSE = \frac{\sqrt{\sum_{(i,t)\in\Omega_{Test}} (\mathbf{Z}_{i,t} - \hat{\mathbf{Z}}_{i,t})^2}}{\sqrt{\sum_{(i,t)\in\Omega_{Test}} (\mathbf{Z}_{i,t} - mean(\hat{\mathbf{Z}}_{i,:}))^2}}, \tag{11}$$

– Empirical Correlation Coefficient ($CORR$):

$$CORR = \frac{1}{I}\sum_{i=1}^{I} \frac{\sum_t (\mathbf{Z}_{i,t} - mean(\mathbf{Z}_{i,:})) \sum_t (\hat{\mathbf{Z}}_{i,t} - mean(\hat{\mathbf{Z}}_{i,:}))}{\sqrt{\sum_t (\mathbf{Z}_{i,t} - mean(\mathbf{Z}_{i,:}))^2 \sum_t (\hat{\mathbf{Z}}_{i,t} - mean(\hat{\mathbf{Z}}_{i,:}))^2}}, \tag{12}$$

where $\mathbf{Z}, \hat{\mathbf{Z}} \in \mathbb{R}^{I\times N}$ are ground true values and model predictions in the MTS task, respectively. I is the number of areas, N is the number of time steps in the testing set, and Ω_{Test} is the set of time stamps used for testing.

RAE is a normalized version of mean absolute error (MAE), and RSE is also a normalized version of mean absolute error ($RMSE$). Hence, both RAE and RSE are not sensitive to the data scale. For RAE and RSE, the lower value is the better performance. Whereas, for $CORR$, the higher value is the better performance. In reality, RAE and RSE describe the prediction accuracy, and $CORR$ describes the similarity.

5.3 Methods for comparison

The proposed model is compared with following methods:

– GAR combines an autoregressive component with a log-linear component, and allows the use of global features to compensate for the lack of data.
– AR is a statistical method to process time series, which is a kind of linear predictive model. The advantage of this method is that it needs little data and can be predicted by its own variable sequence.
– VAR [23] is a generalization of AR, which maps the future values to all past observed values. MA and ARMA can also be transformed into VAR under certain conditions.
– LSTM [24] is a kind of recurrent neural network, which is composed of a cell, an input gate, an output gate and a forget gate. The number of hidden neurons is tuned to optimize the model.
– GRU [25] is a variant of LSTM, which uses an update gate to replace the hidden gates and cell gates of LSTM. The GRU method adjusts hidden neurons to control the scale of a neural network.
– Encoder-decoder (ED) [26] is an extraordinarily ordinary framework in deep learning, which uses RNN in the encoding process and the decoding process, respectively. It's an end-to-end learning framework.
– LSTNet [21] contains a convolutional layer [27] to extract the local dependency patterns, a recurrent layer to capture long-term dependency patterns, and a recurrent-skip layer to capture periodic properties in the input data for prediction.

GAR, AR and GAR are traditional baseline methods. LSTM, GRU and ED are RNN series, which are designed for time series data or sequential data. LSTNet is a state-of-the-art method based on deep neural networks, which is designed for MTS data.

5.4 Configurations

All models are trained using the Adam optimizer [19]. The mean squared error (MSE) is chosen as the loss function of all the models. The batch size is set to 32. For RNN, LSTM, ED and LSTNet, the number of hidden neurons is in {32, 64}. Their learning rates are set to 0.001.

The COVID-19 cases data are divided into two subsets: the first part, from the January 22, 2020 to the July 31, 2020, is used to build and train models; the remaining part, from August 1, 2020 to the September 17, 2020, is utilized to assess the learned models. The ratio of the training set to the test set is 8 : 2.

6 Experimental results and analyses

This section gives several experiments on parameters of the proposed MSL, and compares the MSL with other methods. These experiments are intended to address the following questions:

(1) How T and C affect predictions? Technically, how the inputted data and the learned shapelets affect predictions?
(2) Can MSL outperform other comparable methods?
(3) Finally, what is harvested from the learned shapelets?

6.1 Effects on *C* and *T*

To study how the shapelets and windowed time series affect shapelet learning and predictions, we measure the

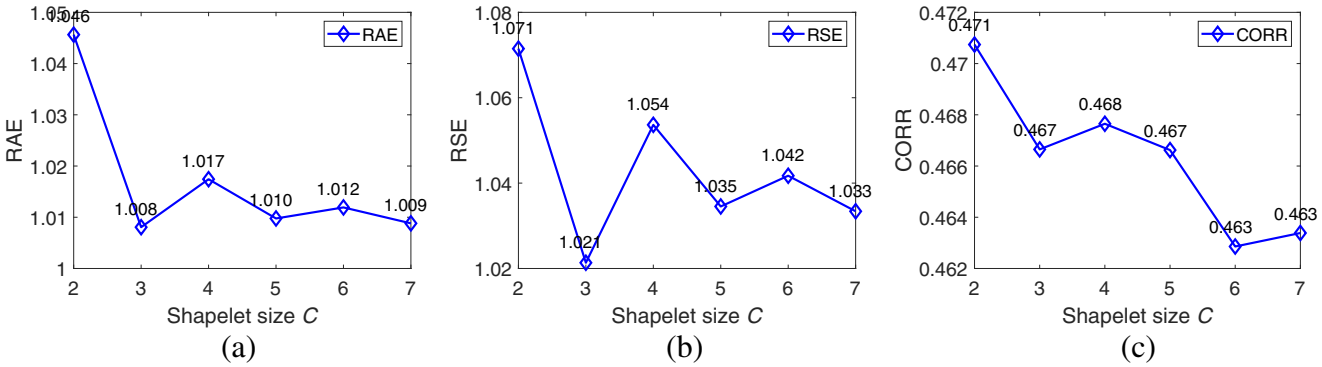

Fig. 2 The sensitiveness of shapelet size C in terms of RAE, RSE and $CORR$. The windows size T is fixed at 28. Both the optimal values of RAE and RSE are found when $C = 3$, and the optimal value of $CORR$ is found at $C = 2$ **a** RAE versus C. **b** RSE versus C. **c** $CORR$ versus C

performance in terms of RAE, RSE and $CORR$. We change C or T while holding other parameters. There are so many compositions of parameter C or parameter T. To efficiently search the parameters, C is first randomly set to a small value, and then T is tuned to obtain the optimal prediction performance. The window size T can be tuned by one of the comparable methods as well. In these experiments, the best performance of comparable methods is found when $T = 28$.

The effects on C are plotted in Fig. 2. As Fig. 2a and b state, we hold window size $T = 28$ and change shapelet size C from 2 to 7 with step size 1, both the optimal values of RAE and RSE are found when $C = 3$. As Fig. 2c reveals, the optimal value of $CORR$ is found at $C = 2$. There are few learned shapelets if the shapelet size C is small. Meanwhile, few learned shapelets have better prediction performances, which means the features of disease outbreaks are few. In reality, the trends of COVID-19 outbreaks in the provinces/states of America are similar.

The effects on T are plotted in Fig. 3. As presented in Figs. 2a, 2b and 2c, the optimal values of these metrics are found at $T = 28$, which is shown in the red dash lines. Compared the window size T with some quick onset disease, such as HFMD [28, 29] and infectious diarrhea [2], the optimal value T of COVID-19 is larger than the value of other diseases. A possible reason is that the COVID-19 has a longer incubation period than other quick onset diseases, or it can spread to other persons in the incubation period.

6.2 Method comparison

According to the conducted experiments on shapelet size C and window size T, we learn the effects on the parameters with respect to the three metrics, the optimal values of these metrics can be found at around $C = 3$ and $T = 28$. Note that, the optimal value of $CORR$ is found at 2 when $T = 28$. The $CORR$ metric measures the directions differences between two vectors, which can also be adopted to measure the trend differences between real confirmed cases and predicted confirmed cases. For a fair competition, the inputted window size T of all methods is set to 28. The parameter C of MSL is set to 3.

As presented in Fig. 4, all the comparable methods are well-tuned, and their performances are measured in terms of RAE, RSE and $CORR$.

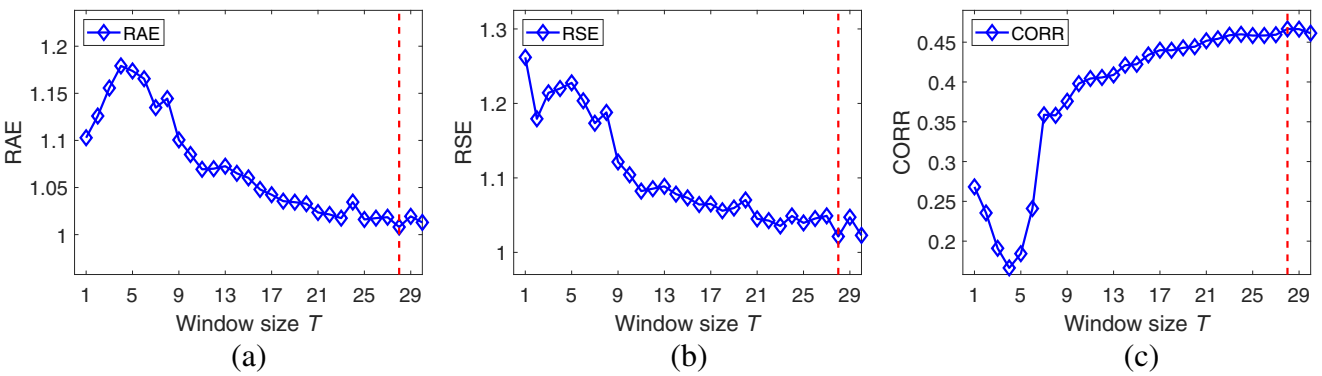

Fig. 3 The sensitiveness of window size T in terms of RAE, RSE and $CORR$. The shapelet size C is fixed at 3. The optimal values of these metrics are found at $T = 28$, see the red dash lines **a** RAE versus T. **b** RSE versus T. **c** $CORR$ versus T

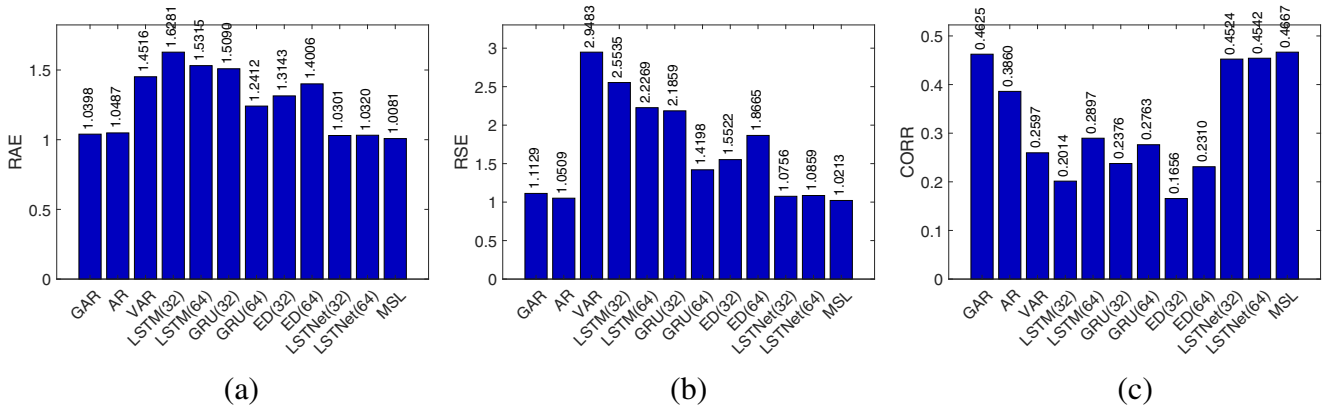

Fig. 4 The comparison of twelve methods in terms of RAE, RSE and $CORR$. The windows size T is fixed at 28. The shapelet size C is set to 3 **a** RAE comparison. **b** RSE comparison. **c** $CORR$ comparison

The following summarizes the key conclusions we observe from the results:

(1) The proposed MSL outperforms other methods in terms of three metrics.
(2) The GAR has the second best performance in terms of RAE and $CORR$.
(3) The AR has the second best performance in terms of RSE.
(4) From the perspective of RAE, the LSTM poorly performs than other methods.
(5) From the perspective of RSE, the VAR and LSTM has the worst performance and second worst performance, respectively.
(6) From the perspective of $CORR$, the ED has the worst performance.

The GAR linearly transforms inputs to targets, and shares common weights for all the inputted variables (i.e., areas). The AR linearly transforms the inputs of an area to the target of this area, and the weights of areas are not shared. The GAR has the second best performance in terms of RAE and $CORR$. Meanwhile, the AR has the second best performance in terms of RSE. This reveals that new confirmed cases in the US are linearly increased somehow.

The VAR linearly connects all inputs to all targets, where each target is mapped from all the inputs. However, the performance is worst in terms of RSE. Meanwhile, the performances of RNN models are poor, such as LSTM, GRU and ED. This discovers that the connections of disease statuses between any two states are relatively independent.

The MSL globally considers important subsequences of time series data. It learns core shapelets from inputs, and generates prediction based on the learned shapelets. Hence, the learned shapelets should be analyzed to investigate the improvements.

6.3 Analyses on shapelets

The visualization of three shapelets is given in Fig. 5. Since the best prediction performance is found at shapelet size $C = 3$ and window size $T = 28$, we obtain three shapelets and the length of each shapelet is 28. For a better comparison and understanding of these shapelets, these shapelets are mapped to range [0, 1] using Min-Max normalization, which is presented in (4).

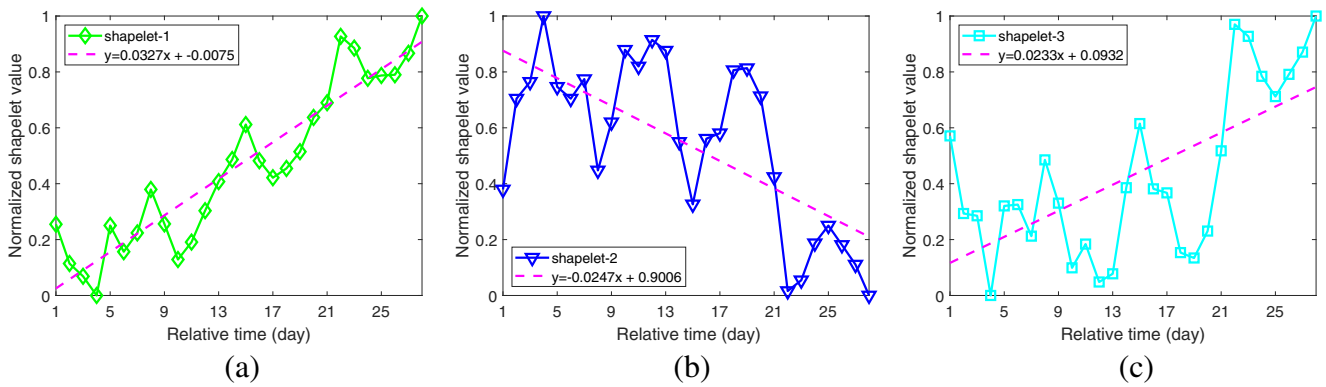

Fig. 5 The visualization of normalized learned shapelets within MSL, given shapelet size $C = 3$ and window size $T = 28$ **a** Shapelet 1. **b** Shapelet 2. **c** Shapelet 3

The major observations from these learned shapelets are as follows:

(1) The length of periods in the three shapelets are around seven days. This reveals that the activities in American have a strong periodic connection to the disease.
(2) When observing periods in Fig. 5a and 5c, it takes two days to degrade the new confirmed cases, but will strongly increase in the coming five days.
(3) According to the slopes of the trend lines, the growth rate is larger than the descent rate. This suggests that the infection among people is going stronger.

From the above observations, the trends of new cases in some areas are still increasing quickly (e.g., California), while some areas are slowly descending. The proposed MSL learns these trends, and then generates predictions based on them. It also suggests that future prediction models should consider the periodic events, such as weekdays and weekends.

7 Conclusions

This paper investigated the simultaneous prediction of the upcoming new confirmed COVID-19 cases in 50 provinces/states in America. The MSL is proposed to generate predictions via shapelet learning. Experimental results on real data collections show the effectiveness of the proposed method. Meanwhile, experimental analyses show the COVID-19's incubation period is around 28 days. Moreover, three learned shapelets depict the growing trend and descending trend of the disease.

In the future, the multi-horizon COVID-19 prediction will be further investigated, which would provide further visions for disease prevention and control.

Acknowledgments This work was supported in part by Jimei University (no. ZP2021013), the Education Department of Fujian Province (CN) (no. JAT200277), the Natural Science Foundation of Fujian Province (CN) (no. 2019J01713) and the Natural Science Foundation of China (nos. 41971424 and 61701191). The authors would like to thank the editor and anonymous reviewers for their helpful comments in improving the quality of this paper.

Declarations

Conflict of Interests None.

References

1. WHO (2020) Coronavirus disease (covid-19) outbreak situation. https://www.who.int/emergencies/diseases/novel-coronavirus-2019. (accessed 2020)
2. Wang Z, Huang Y, He B, Luo T, Wang Y, Fu Y (2020) Short-term infectious diarrhea prediction using weather and search data in xiamen, china. Sci Program 2020:1–12. https://doi.org/10.1155/2020/8814222
3. Assaf R, Schumann A (2019) Explainable deep neural networks for multivariate time series predictions. In: Proceedings of the 28th international joint conference on artificial intelligence, IJCAI Organization, Macao, China, pp 6488–6490
4. Ye L, Keogh EJ (2009) Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th international conference on knowledge discovery and data mining, ACM, Paris, France, pp 947–956
5. Grabocka J, Schilling N, Wistuba M, Schmidt-Thieme L (2014) Learning time-series shapelets. In: Proceedings of the 20th international conference on knowledge discovery and data mining, ACM, New York, US, pp 392–401
6. Bertozzi AL, Franco E, Mohler G, Short MB, Sledge D (2020) The challenges of modeling and forecasting the spread of COVID-19. Proceedings of the National Academy of Sciences 117(29):16732–16738. https://doi.org/10.1073/pnas.2006520117
7. Hernandez-Matamoros A, Fujita H, Hayashi T, Perez-Meana H (2020) Forecasting of COVID19 per regions using ARIMA models and polynomial functions. Appl Soft Comput 96:106610. https://doi.org/10.1016/j.asoc.2020.106610
8. Ceylan Z (2020) Estimation of COVID-19 prevalence in italy, spain, and france. Science of The Total Environment 729:138817. https://doi.org/10.1016/j.scitotenv.2020.138817
9. Singh RK, Rani M, Bhagavathula AS, Sah R, Rodriguez-Morales AJ, Kalita H, Nanda C, Sharma S, Sharma YD, Rabaan AA, Rahmani J, Kumar P (2020) Prediction of the covid-19 pandemic for the top 15 affected countries: Advanced autoregressive integrated moving average (arima) model. JMIR Public Health Surveill 6(2):e19115. https://doi.org/10.2196/19115
10. Mizumoto K, Chowell G (2020) Transmission potential of the novel coronavirus (COVID-19) onboard the diamond princess cruises ship, 2020. Infectious Disease Modelling 5:264–270. https://doi.org/10.1016/j.idm.2020.02.003
11. Li L, Yang Z, Dang Z, Meng C, Huang J, Meng H, Wang D, Chen G, Zhang J, Peng H, Shao Y (2020) Propagation analysis and prediction of the COVID-19. Infectious Disease Modelling 5:282–292. https://doi.org/10.1016/j.idm.2020.03.002
12. Fong SJ, Li G, Dey N, Crespo RG, Herrera-Viedma E (2020) Composite monte carlo decision making under high uncertainty of novel coronavirus epidemic using hybridized deep learning and fuzzy rule induction. Appl Soft Comput 93:106282. https://doi.org/10.1016/j.asoc.2020.106282
13. Farooq J, Bazaz MA (2020) A deep learning algorithm for modeling and forecasting of COVID-19 in five worst affected states of india. Alexandria Engineering Journal. https://doi.org/10.1016/j.aej.2020.09.037
14. Keogh EJ, Rakthanmanon T (2013) Fast shapelets: A scalable algorithm for discovering time series shapelets. In: Proceedings of the 13th SIAM international conference on data mining, SIAM, Austin, Texas, USA, pp 668–676
15. Li G, Yan W, Wu Z (2019) Discovering shapelets with key points in time series classification. Expert Syst Appl 132:76–86. https://doi.org/10.1016/j.eswa.2019.04.062
16. Hou L, Kwok JT, Zurada JM (2016) Efficient learning of timeseries shapelets. In: Proceedings of the 30th international conference on artificial intelligence, AAAI Press, Phoenix, Arizona, USA, pp 1209–1215
17. Deng H, Chen W, Shen Q, Ma AJ, Yuen PC, Feng G (2020) Invariant subspace learning for time series data based on dynamic time warping distance. Pattern Recogn 102:107210. https://doi.org/10.1016/j.patcog.2020.107210

18. Raychaudhuri DS, Grabocka J, Schmidt-Thieme L (2017) Channel masking for multivariate time series shapelets. arXiv:1711.00812 [cs]
19. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: Proceedings of the 3rd international conference on learning representations, OpenReview.net, San Diego, CA, USA
20. Shih S-Y, Sun F-K, Lee H-Y (2019) Temporal pattern attention for multivariate time series forecasting. Mach Learn 108(8-9):1421–1441. https://doi.org/10.1007/s10994-019-05815-0
21. Lai G, Chang W-C, Yang Y, Liu H (2018) Modeling long- and short-term temporal patterns with deep neural networks. In: Proceedings of the 41st international conference on research development in information retrieval, ACM, Ann Arbor, MI, USA, pp 95–104
22. Chang Y-Y, Sun F-Y, Wu Y-H, Lin S-D (2018) A memory-network based solution for multivariate time-series forecasting. arXiv:1809.02105 [cs]
23. Zhang GP (2003) Time series forecasting using a hybrid arima and neural network model. Neurocomputing 50:159–175. https://doi.org/10.1016/S0925-2312(01)00702-0
24. Cao J, Li Z, Li J (2019) Financial time series forecasting model based on ceemdan and lstm. Physica A: Statistical mechanics and its applications 519:127–139. https://doi.org/10.1016/j.physa.2018.11.061
25. Cho K, van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder-decoder approaches. In: The 8th EMNLP workshop on syntax, semantics and structure in statistical translation, Doha, Qatar, pp 103–111
26. Cho K, van Merrienboer B, Gülçehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, Doha, Qatar, pp 1724–1734
27. Djenouri Y, Srivastava G, Li JC-W (2020) Fast and accurate convolution neural network for detecting manufacturing data. IEEE Transactions on Industrial Informatics, p 1–1. https://doi.org/10.1109/TII.2020.3001493
28. Wang Z, Huang Y, He B (2021) Dual-grained representation for hand, foot, and mouth disease prediction within public health cyber-physical systems. Software: Practice and Experience, Early View. https://doi.org/10.1002/spe.2940
29. Wang Z, Huang Y, He B, Luo T, Wang Y, Lin Y (2019) TDDF: HFMD outpatients prediction based on time series decomposition and heterogenous data fusion in xiamen, china. In: Proceedings of the 15th international conference advanced data mining and applications, Dalian, China, pp 658–667

Zhijin Wang received the Ph.D. degree from the Department of Computer Science and Technology, East China Normal University, Shanghai, China, in 2016. He is currently with the Computer Engineering College, Jimei University, Xiamen, China. His current research interests include recommender system, data mining, and artificial intelligence in healthcare.

Bing Cai is currently pursuing a B.S. degree in computer science at Jimei University, Xiamen, China. His research interests include artificial intelligence and time series forecasting.