



Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.



# A data-driven hybrid ensemble AI model for COVID-19 infection forecast using multiple neural networks and reinforced learning

Weiqliu Jin<sup>a,b,1</sup>, Shuqing Dong<sup>c,1</sup>, Chengqing Yu<sup>c</sup>, Qingquan Luo<sup>a,b,\*</sup>

<sup>a</sup> Shanghai Lung Cancer Center, Shanghai Chest Hospital, Shanghai Jiao Tong University, Shanghai, 200030, PR China

<sup>b</sup> School of Medicine, Shanghai Jiao Tong University, Shanghai, 200025, PR China

<sup>c</sup> School of Traffic and Transportation Engineering, Central South University, Hunan, 410075, PR China

## ARTICLE INFO

### Keywords:

COVID-19  
Infection prediction  
Hybrid model  
Artificial intelligence

## ABSTRACT

The COVID-19 outbreak poses a huge challenge to international public health. Reliable forecast of the number of cases is of great significance to the planning of health resources and the investigation and evaluation of the epidemic situation. The data-driven machine learning models can adapt to complex changes in the epidemic situation without relying on correct physical dynamics modeling, which are sensitive and accurate in predicting the development of the epidemic. In this paper, an ensemble hybrid model based on Temporal Convolutional Networks (TCN), Gated Recurrent Unit (GRU), Deep Belief Networks (DBN), Q-learning, and Support Vector Machine (SVM) models, namely TCN-GRU-DBN-Q-SVM model, is proposed to achieve the forecasting of COVID-19 infections. Three widely-used predictors, TCN, GRU, and DBN are used as elements of the hybrid model ensemble by the weights provided by reinforcement learning method. Furthermore, an error predictor built by SVM, is trained with validation set, and the final prediction result could be obtained by combining the TCN-GRU-DBN-Q model with the SVM error predictor. In order to investigate the forecasting performance of the proposed hybrid model, several comparison models (TCN-GRU-DBN-Q, LSTM, N-BEATS, ANFIS, VMD-BP, WT-RVFL, and ARIMA models) are selected. The experimental results show that: (1) the prediction effect of the TCN-GRU-DBN-Q-SVM model on COVID-19 infection is satisfactory, which has been verified in three national infection data from the UK, India, and the US, and the proposed model has good generalization ability; (2) in the proposed hybrid model, SVM can efficiently predict the possible error of the predicted series given by TCN-GRU-DBN-Q components; (3) the integrated weights based on Q-learning can be adaptively adjusted according to the characteristics of the data in the forecasting tasks in different countries and multiple situations, which ensures the accuracy, robustness and generalization of the proposed model.

## 1. Introduction

### 1.1. Background

With the COVID-19 pandemic, the world's health system is suffering a huge impact. The effective estimation (or prediction, forecasting) of the number of COVID-19 cases will be of great help for each country to plan its own health policies (including vaccination, quarantine, isolation, lockdown, social distancing, etc.) and estimate the economic and social losses of the epidemic [1]. Scholars have been committed to solving the problems of COVID-19 incidence prediction and epidemiological modeling, and proposed epidemiological models (SIR [2], SEIR

[3,4], SIRD [5], phenomenology [6], etc.), time series models (autoregressive models [7,8], exponential models [9], regression model [10, 11], Prophet model [12], etc.), machine learning model (based on regression tree [13], LSTM [14], polynomial neural network [15], ANFIS [16], SVM [17], etc.) and other types of models [18].

Classical epidemiological studies are mostly deterministic and works with large populations [18]. They are constructed based on correct physical dynamic modeling, which are based on SIR dynamics models and parameter estimation methods in statistics to complete the modeling of the epidemiological pathology and transmission process, and then to predict the process characteristics of the disease epidemic, but the accuracy of such dynamic models depends on a complete and

\* Corresponding author. Shanghai Lung Cancer Center, Shanghai Chest Hospital, Shanghai Jiao Tong University, Shanghai, 200030, PR China.

E-mail address: [luoqingquan@hotmail.com](mailto:luoqingquan@hotmail.com) (Q. Luo).

<sup>1</sup> These authors contribute to this work equally and should be considered as co-first authors.

accurate description of the dynamics process and highly dependent on the results of reliable parameter estimation [18]. Therefore, although the SIR dynamics model can give a long-term analysis of transmission characteristics, their reliability is limited by many changing factors including population immunity status of diseases (such as vaccines), public events (such as quarantine, migration, and other policy changes) and so on. For example, in Bhattacharjee et al.'s SAHQD (Susceptible, infected, hospitalized, quarantined, deceased) model, a complex multi-compartment dynamics model, although information on the social distancing measures and diagnostic testing rates are incorporated to characterize the dynamics of the various compartments of their model, the degree of social distance restrictions and the mobility within the population were neglected [19].

Data-driven models, including statistic models and machine-learning models, overcome these shortcomings to a great extent. Statistic models such as Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA) were adopted by previous COVID-19 infection prediction studies. K.E. Arunkumar et al. adopted Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA) forecasts to predict the future trend (rising or falling) of the COVID-19 epidemic in the top 16 countries [20]. Christopher J Lynch et al. adopted and compared Holt-Winters exponential smoothing (HW), growth rate model (Growth), moving average (MA), autoregressive (AR), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models [21]. However, their prediction accuracy remains to be further improved.

Recently, deep learning models have been proved to be a reliable and promising tool in data prediction. Therefore, there are also many studies focusing on the potential of applying deep learning to predict the COVID-19 infection numbers. Most of previous researches were based on single models, such as LSTM, MLP, ELM, etc. Ammar H. Elsheikh et al. applied the LSTM model to predict the total number of confirmed cases and deaths in 6 different countries; Brazil, India, Saudi Arabia, South Africa, Spain and the United States. Their model is only based on a single LSTM model, and the prediction accuracy needs to be improved [22]. Nasrin Talkhi et al. compared nine models including NNETAR, ARIMA, Holt-Winter, BSTS, TBATS, Prophet, MLP and ELM network models by evaluating indicators RMSE, MAE and MAPE%, and they selected the best model that had the lowest value of the performance index [23].

Generally, hybrid models proposed by several scholars has better overall performances in COVID-19 infection number forecasting than single models. For example, Mohammed A.A. Al-qaness et al. proposed a new short-term prediction model, using an enhanced version of the Adaptive Neuro-Fuzzy Inference System (ANFIS). An improved marine predator algorithm (MPA), chaotic MPA (CMPA), is proposed, which improves the ANFIS algorithm [16]. Nanning Zheng proposed a hybrid AI model based on an improved susceptible-infected (ISI) model, the natural language processing (NLP) module and the long short-term memory (LSTM) to predict the cumulative infection numbers in China during February 19, 2020 to February 24, 2020 [24]. Although their research used hybrid models, the accuracy of the single predictor could be further improved, and the weights of their component models were given in a manual or randomized way. Meanwhile, the error compensation was not considered and embedded in their hybrid model systems [25].

### 1.2. Motivation and our work

We conducted this study based on several considerations: First, most of the current studies carry out prediction of COVID-19 incidence based on dynamics modeling or multivariate data regression, etc., often relying on complex compartment models or extensive but not always available multidimensional data for forecasting, which makes the models less effective and powerful. Meanwhile, due to the unavailability or incompleteness of data and factors that interfere with the dynamic state,

such as epidemic policies and virus variants, it is difficult to consistently describe the progress of the epidemic or to generalize across regions. Second, current studies of deep learning and reinforcement learning for COVID-19 forecast are relatively few compared to dynamics modeling and classical time series analysis (e.g., ARIMA), whereas they are very likely to have a significant role in COVID-19 morbidity prediction and deserve to be included in researchers' perspectives. Third, the proposed ensemble model can improve the generalizability of existing deep learning sub-models across geographies and time, achieving satisfactory predictions using only a single time series of COVID-19 daily incidence numbers. It also provides an idea of weight training and error compensation based on reinforcement learning framework, which can be used as a reference for subsequent integration of multimodal data and hybrid integration of multiple modeling approaches.

In this work, we present a data-driven TCN-GRU-DBN-Q-SVM ensemble hybrid model. First, three widely used networks, TCN, GRU and DBN, are used as single predictors. Second, three predictors are ensemble by reinforcement learning method (Q-learning) with different weights. Third, an error predictor built by SVM, is trained with validation set, and the final prediction result could be obtained by combining the TCN-GRU-DBN-Q model with the SVM error predictor. The proposed model adopted multiple newly-proposed predictors (TCN, GRU and DBN) with satisfying standalone performance to work collectively, and the integrated weights based on Q learning can be adaptively adjusted according to the characteristics of the data in different countries and multiple situations, which ensures the accuracy and generalization of the proposed model. Meanwhile, an SVM-based error compensation mechanism was utilized to further improve the accuracy of the model. Our study can better improve the generalization ability and accuracy of the model on COVID-19 prediction driven by single time series data through an ensemble way, and the study provides a possible framework for embedding other data modalities or other modeling methods based on reinforcement learning, which can enrich and inspire the methodology of COVID-19 prediction to some extent.

### 1.3. Structure of this article

In this article, Section 2 (**Literature review**) further subsumes and summarizes the existing literature on COVID-19 forecast based on a brief summary of basic COVID-19 forecast tools given in Section 1 (**Introduction**).

Section 3 (**Methods**) shows the architecture of the model and its training steps (section 3.1), and introduces the three sub-models TCN, GRU, DBN and the basic ideas of Q-learning and SVM error compensation (section 3.2).

Section 4 (**Experiment**) introduces the datasets used in the experiments and the split of training, validation, and test set (section 4.1), and specifies the metrics to evaluate the models (section 4.2). Section 4.3 gives the parameters for model training and shows the process of determining the number of input neurons (section 4.3.1). It gives the optimized weights derived from Q-learning during the model integration, and compares the performance of the ensemble model and the component models to clarify the effectiveness of model (section 4.3.2).

Section 5 (**Results and Discussion**) discusses the forecast results of the model and verifies the good generalization ability and prediction accuracy of the integrated model by comparing it with the existing model on three different national datasets.

Section 6 (**Limitations and Further Work**) gives the application scenario of the model, and suggests some possible directions for improvement and factors to be considered when improving the model.

Section 7 (**Conclusion**) gives the conclusion of the study.

## 2. Literature review

Table 1 and Fig. 1 (A) show the prediction models for the number of COVID-19 incidences from the existing literature. It can be seen that the

**Table 1**  
A summary of recent COVID-19 forecast models.

| (A) Recent COVID-19 forecast models   |   |   |   |                      |   |  |                   |      |
|---|---|---|---|----------------------|---|--|-------------------|------|
| Types   | Method  | Performance                                   |   |                      |   | Religion                                       | Ref.              |      |
|   |   | MAE   | RMSE  | Pearson              | Spearman  |  |                   |      |
| Dynamics  | SAHQD model (Susceptible, infected, hospitalized, quarantined, deceased)  | N.P.  | N.P.  | N.P.                 | N.P.  | U.S.   | [19]              |      |
|   | SCUAQIHMRD model (Susceptible, close contact, uninfected under home quarantine, asymptomatic under home quarantine, mild symptoms under home quarantine, severe symptoms under home quarantine, infectious in Designed Hospitals, infectious in Fangcang Hospitals, Recovered, Death) | N.P.  | N.P.  | N.P.                 | N.P.  | Wuhan, China                                   | [31]              |      |
|   | SEPIAHR model (Susceptible, exposed, pre-symptomatic infectious, ascertained infectious, unascertained infectious, isolation in hospital and removed)   | N.P.  | N.P.  | N.P.                 | N.P.  | Wuhan, China                                   | [30]              |      |
|   | SEI <sub>h</sub> IR model (Susceptible, incubation, asymptotic infected, recovered)   | N.P.  | N.P.  | N.P.                 | N.P.  | Wuhan, China                                   | [28]              |      |
|   | SEIRMH model (Susceptible, exposed without symptoms, infected with symptoms, with medical care, and removed from the system)  | N.P.  | N.P.  | 0.84                 | N.P.  | Belgium  | [29]              |      |
|   | Adaptive interacting cluster-based SEIR (AICSEIR) model   | N.P.  | N.P.  | 0.84                 | N.P.  | Italy, the U.S., and India                     | [44]              |      |
|   | modified SEIR model (Including vaccination)   | N.P.  | N.P.  | N.P.                 | N.P.  | NYC, U.S.                                      | [45]              |      |
|   | SEIR model with Bayesian inference  | N.P.  | N.P.  | N.P.                 | N.P.  | Israel   | [46]              |      |
|   | SLIR model (Susceptible, latent, infected, recovered)   | N.P.  | N.P.  | N.P.                 | N.P.  | China  | [27]              |      |
|   | SEIR model  | N.P.  | N.P.  | N.P.                 | N.P.  | Texas, USA                                     | [26]              |      |
|   | Sequential compartmental models   | N.P.  | N.P.  | N.P.                 | N.P.  | Homeless Shelter, Chicago, Illinois, USA       | [47]              |      |
|   | Time series   | smooth transition autoregressive (STAR) model | 0.208   | 0.297                | N.P.  | N.P.   | Africa sub-region | [35] |
|   |   | Linear AR model                               | 0.251   | 0.385                | N.P.  | N.P.   | Africa sub-region | [35] |
| ARIMA   |   | 27.86   | 35.69   | N.P.                 | N.P.  | Malaysia                                       | [32]              |      |
| ARIMA   |   | N.P.  | N.P.  | N.P.                 | N.P.  | France   | [33]              |      |
| Modified VAR regression   |   | 47.43   | N.P.  | N.P.                 | N.P.  | NYC, U.S.                                      | [36]              |      |
| Linear regression   |   | N.P.  | 7.562   | N.P.                 | N.P.  | Iran   | [40]              |      |
| Poisson count time series model (Disease surveillance and Twitter-based population mobility data) |   | N.P.  | N.P.  | N.P.                 | N.P.  | South Carolina                                 | [48]              |      |
| Grey forecast   | ARIMA   | 50.109  | 95.322  | N.P.                 | N.P.  | India  | [34]              |      |
|   | Fractional Order Accumulation Grey Model (FGM)  | N.P.  | 109496/96411/<br>14560/64253/15/<br>1123/106223 | N.P.                 | N.P.  | U.S., France, UK, Germany, China, Japan, India | [37]              |      |
|   | Hybrid grey exponential smoothing approach  | N.P.  | 5.05  | N.P.                 | N.P.  | Sri Lanka                                      | [38]              |      |
|   | Internally Optimized Grey Prediction Models (IOGMs)   | N.P.  | N.P.  | N.P.                 | N.P.  | Rajasthan, Maharashtra, Delhi                  | [39]              |      |
| ML methods  | random forest regression algorithm  | 5.42  | 9.27  | 0.89                 | 0.84  | 215 countries and territories                  | [49]              |      |
|   | long short-term memory (LSTM) models  | N.P.  | 27.187  | N.P.                 | N.P.  | Iran   | [40]              |      |
|   | multilayer perceptron (MLP) neural network ( <i>n</i> hidden layer)   | 0.36 ( <i>n</i> = 1)                          | 0.64 ( <i>n</i> = 1)                            | 0.36 ( <i>n</i> = 1) | N.P.  | U.S.   | [50]              |      |
|   |   | 0.40 ( <i>n</i> = 2)                          | 0.84 ( <i>n</i> = 2)                            | 0.47 ( <i>n</i> = 2) |   |  |                   |      |
|   | Pearson correlation test and general linear model   | N.P.  | N.P.  | 0.978                | N.P.  | U.S.   | [51]              |      |
|   | a simple random forest statistical model  | N.P.  | N.P.  | 0.89                 | N.P.  | Ohio, U.S.                                     | [52]              |      |
|   | WEKA tool   | ≈1200   | ≈1000   | N.P.                 | N.P.  | Pakistan                                       | [53]              |      |
|   | deep interval type-2 fuzzy LSTM (DIT2FLSTM)   | N.P.  | N.S.  | N.P.                 | N.P.  | USA, Brazil, etc.                              | [41]              |      |
|   | generalized linear and tree-based machine learning models   | 0.21  | N.P.  | 0.99                 | N.P.  | Tennessee                                      | [54]              |      |
|   | an ensemble of 10 LSTM-based networks   | 90.38   | N.P.  | N.P.                 | N.P.  | The county-level in the US                     | [42]              |      |
| LSTM + <i>R<sub>t</sub></i> method  | N.P.  | N.P.  | 0.872   | N.P.                 | West Virginia   | [43]   |                   |      |
| Least-Square Boosting Classification algorithm  | 1200  | N.P.  | N.P.  | N.P.                 | Countries having maximum number >2000 of confirmed cases in a day | [55]   |                   |      |

(B) Comparison between different types of COVID-19 forecast models

| Types         | Strength   | Weakness   |
|---------------|--|--|
| Dynamics      | ➤ Able to forecast over a wide future time window                | ➤ Cannot be adapted to situations where the model subject has increased or where model parameters change with specific policies, disease pathogen variability, etc.<br>➤ High requirements for parameter estimation<br>➤ High demands on data sources, some of which are often missing or inaccessible, and their neglect often leads to unrealistic model assumptions |
|               | ➤ The physical meaning of the model is very clear                |  |
|               |  |  |
| Time series   | ➤ Simple and reproducible steps                                  | ➤ Application scenarios are limited, e.g. ARMA model can only be applied to stationary situations, etc.  |
|               | ➤ The required data are easily available                         |  |
|               | ➤ Particularly suitable for cases where time series are periodic |  |
| Grey forecast | ➤ Simple and reproducible steps                                  | ➤ Similar to fuzzy mathematical theory, the physical meaning of the model is not clear   |
|               | ➤ The required data are easily available                         |  |
|               |  |  |

(continued on next page)

Table 1 (continued)

| (B) Comparison between different types of COVID-19 forecast models |   |  |
|--|---|--|
| Types  | Strength  | Weakness   |
| Machine learning   | <ul style="list-style-type: none"> <li>➤ Higher potential for hybrid integration</li> <li>➤ Satisfactory performance for information mining</li> <li>➤ Satisfactory accuracy if the amount of data permits</li> </ul> | <ul style="list-style-type: none"> <li>➤ Features obtained by deep learning methods do not have a clear physical meaning</li> <li>➤ The generalization ability of the model may be insufficient</li> </ul> |

**Notes:** N.P. = Not provided. Meanwhile, it is worth noting that although we give specific model performance in above table, it is not generalizable and comparable across datasets due to the different number of infections within different geographic regions.

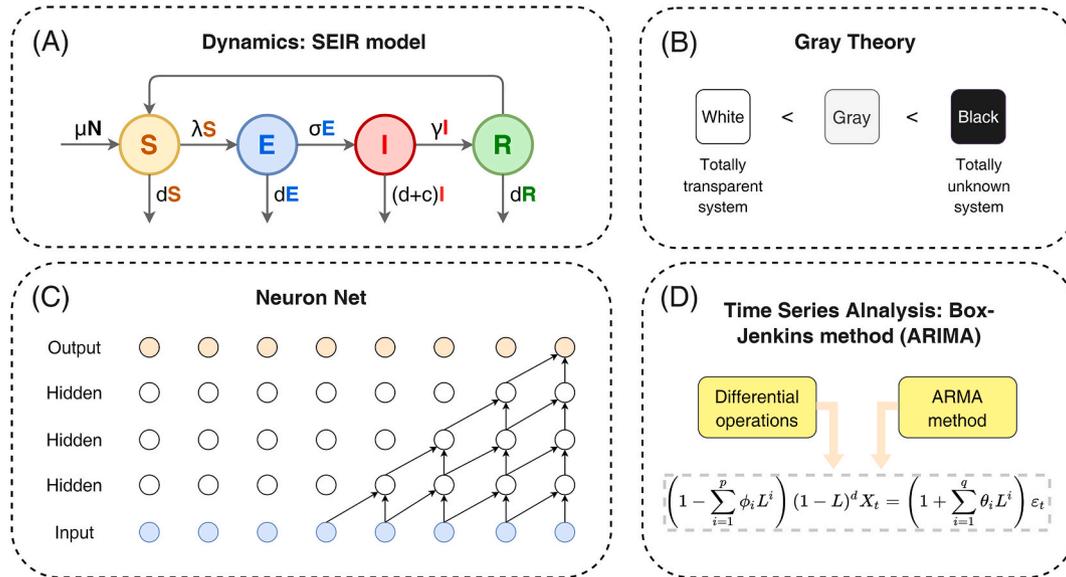


Fig. 1. (A) Dynamic modeling of COVID-19 pandemic, SEIR model as an example; (B) Basic idea of forecast with Grey Theory; (C) A neuron net structure in Deep-learning forecast of COVID-19; (D) ARIMA model, a combination of differential operation and ARMA method.

current COVID-19 prediction tools are mainly of the following types:

- 1) Models based on kinetic (or dynamics) modeling, which further clarify the pattern of changes in the number of a specific population by dividing the population into **Susceptible (S), Infected (I), Recovered (R)** compartments, etc., and defining the transition relationships among them through several differential equations. Since the onset of the COVID-19 epidemic, multi-generational kinetic models have emerged, whose evolution route could be summarized as: SIR→SEIR [26], SLIR [27] (considering close contacts or the latent) → SEI<sub>A</sub>IR [28] (considering asymptomatic infected persons), SAHQD [19] (considering quarantine policies) → SEIRMH [29], SEPIAHR [30] (considering medical-related factors) → SCUA-QIHMRD [31] (considering COVID-19 hierarchical treatment).
- 2) Time series analysis. The simpler exponential smoothing model arranges the data in chronological order from new to old. The weights are assigned from large to small, and the weight values are exponentially decreasing. In addition to exponential smoothing, which smoothes the data based on exponentially decreasing parameters, there is also the common method of fitting an ARIMA [32–34] model, which consists of three parts: the autoregressive process (AR), the differential part (Integrated) and the moving average process. In addition, there are multivariate time series analysis based on the standard Autoregressive model (AR) [35], including the VAR (Vector Autoregressive model) [36], STAR (smooth transition autoregressive) [35] and their modifications.
- 3) Forecast model based on Grey Theory. Grey prediction model is a prediction method that builds a mathematical model to make a forecast through a small amount of incomplete information. The modifications of the basic GM(1,1) model, such as Fractional Order

Accumulation Grey Model (FGM) [37], Hybrid grey exponential smoothing approach [38], and Internally Optimized Grey Prediction Models (IOGMs) [39], have been proposed to be effective tools for COVID-19 forecast.

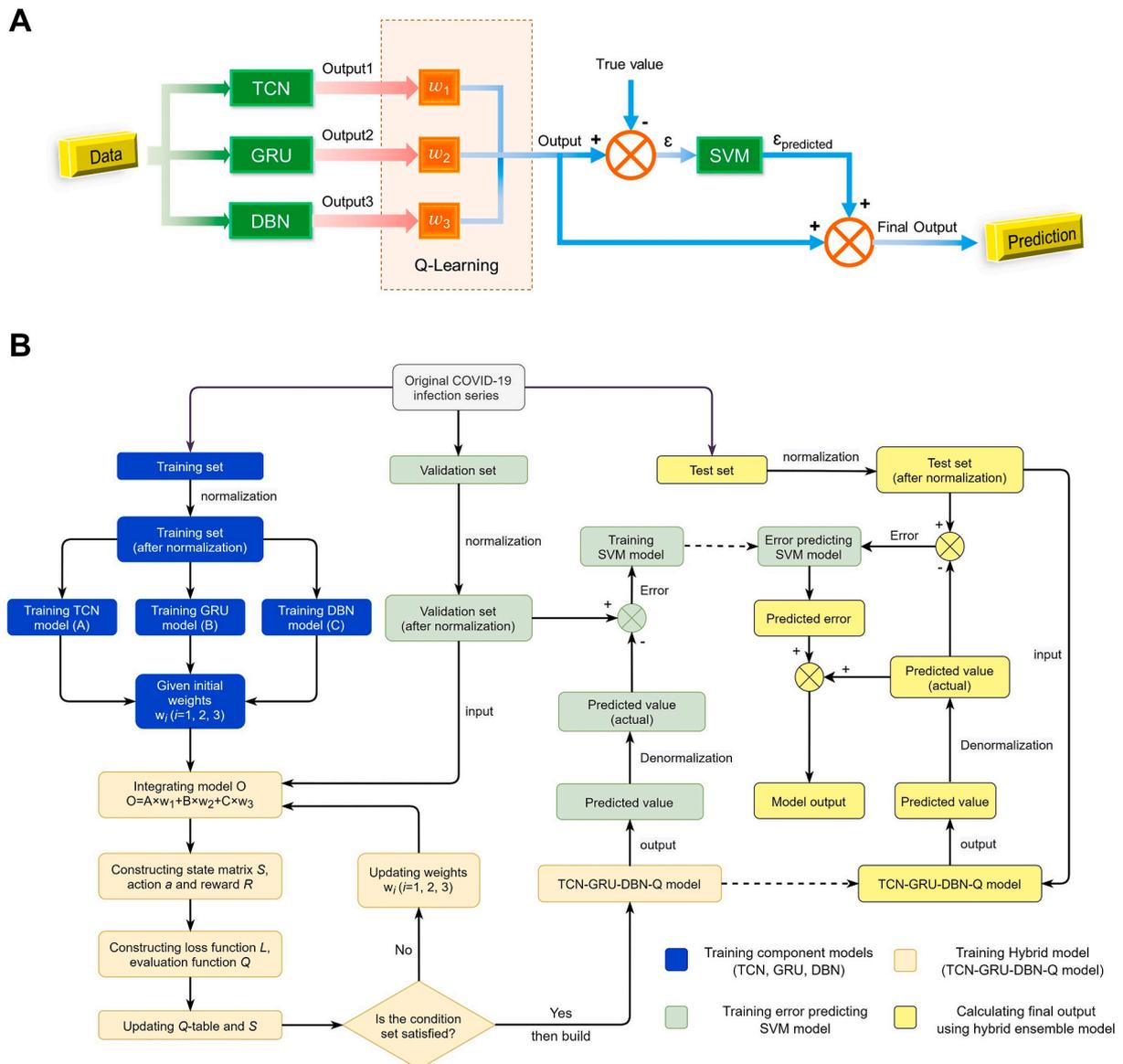
- 4) Forecast model based on Machine Learning. Models such as LSTM, RF regressions, shown in Table 1 (A), have been widely researched in the COVID-19 forecast. Long short-term memory (LSTM), the most representative one of deep-learning, is a special kind of RNN, which is mainly designed to solve the gradient disappearance and gradient explosion problems during the training of long sequences. LSTM has been proved to be a powerful deep-learning network for the forecast of COVID-19 infections [40–43].

A comparison of the advantages and disadvantages of the four types of prediction approaches is given in Table 1(B). From the literature review, it can be seen that the current research is still less on deep-learning based ensemble models and more focused on broadening the amount of information covered by existing models from multiple data modalities. It is meaningful to investigate how to further optimize the performance of deep learning models on single-input time series through weight optimization and error compensation strategies in the ensemble framework to further improve the accuracy and generalization of COVID-19 forecast models.

### 3. Methods

#### 3.1. Model architecture

As shown in Fig. 2, our model includes three sub-predictors TCN, GRU, and DNB. These sub-predictors give prediction results



**Fig. 2.** (A) The architecture of the proposed model; (B) The experimental procedures of this study, where the total experiment was divided into four parts: training the component model, training the hybrid model, training the SVM model for error predicting, and calculating the final prediction value by hybrid ensemble model composed of TCN-GRU-DBN-Q hybrid model and error-predicting SVM model.

respectively. Under the integrated effect of Q reinforcement learning, these three models are given different weights, and then an integrated model could be built based on TCN, GRU, DNB, and their weights  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$ . We further trained an error training model based on SVM by comparing the prediction results with the validation set. By integrating the SVM model with the synthetic predictor obtained by Q reinforcement learning, the final hybrid integration model is obtained. The detailed steps of model building are as follows:

- Step 1:** Preprocess and normalize time series data.
- Step 2:** Train the three neural networks, TCN, GRU and DBN, with the time-series data of the training set respectively, and then input the test set into the network to obtain the test result sequence, denoted as  $x_1$ ,  $x_2$ , and  $x_3$ .
- Step 3:** Give a randomized initial weight of each output sequence as  $\omega_1$ ,  $\omega_2$ ,  $\omega_3$ , and set the output:

$$output = \omega_1 X_1 + \omega_2 X_2 + \omega_3 X_3$$

In order to optimize the weights of three network, reinforce learning

method, Q-learning, is used. The optimization goal of Q-learning was to minimize the output RMSE (namely the loss function, or the evaluation function  $Q$  in Q-learning training) value:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - y'_i)^2}$$

where  $y_i$  is the actual value, and  $y'_i$  is the forecasting value. The Q-learning is trained with validation set.

- Step 4:** Calculate the error between  $O$  and the actual sequence (in validation set) and get the error sequence  $R$ . Taking  $R$  as the training set, SVM is used to model the error sequence, which gives a compensation for the predictive result given by the TCN-GRU-DBN-Q model.

- Step 5:** Input the test set into the hybrid ensemble model to get the *output*. For each *output*, input it into the error prediction SVM model to get the prediction error  $error_{pred}$ , then the final output  $O$  is:

$$O = \text{output} + \text{error}_{\text{pred}}$$

### 3.2. Components of the proposed model

#### 3.2.1. Temporal Convolutional Networks (TCN)

TCN is adopted as a sub-predictor of the proposed ensemble model. In essence, TCN is an integration of fully convolutional networks, causal convolution, dilated convolution, and residual connections [56]. First, generally, TCN combines the 1D FCN and casual convolutions [57]. In FCN architecture, each layer is the same length as the input layer, and a zero padding layer is added to keep subsequent layers the same length as previous ones, which assures that the output produced by the network is of the same length as the input [56]. Second, the casual convolutions, where an output at time  $t$  is convolved only with elements from the time  $t$  and earlier in the previous layer, are also adopted in TCN to ensure that there is no information leakage. Third, dilated convolutions enable TCN to adapt to forecasting tasks with a longer history, i.e., to expand the receptive field. For each 1-D input sequence  $\mathbf{x} \in \mathbb{R}^n$  and a filter  $f: \{0, \dots, k-1\} \rightarrow \mathbb{R}$ , the operation  $F$  on element  $s$  of the sequence is defined as [57]:

$$F(s) = (\mathbf{x}^*_{d} f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i}$$

where the dilation factor is  $d$ ,  $k$  denotes the size of the filter  $f: \{0, \dots, k-1\} \rightarrow \mathbb{R}$ ,  $s-d \cdot i$  is the direction of the past [57]. Thus, a fixed step is introduced between every two adjacent filter taps, and the larger the dilation is, the wider the input ranges (i.e. an output at the top level) could be, which increase the receptive field to a great extent. Fourth, a residual block (shown in Fig. 3 (B) and (C)) is added to the model to allow layers to adapt to the modifications to the identity mapping. It contains a branch leading out to serial transformations  $\mathcal{F}(\mathbf{x})$ :

$$o = \text{Activation}(x + \mathcal{F}(\mathbf{x}))$$

where  $o$  accounts for the outputs added to the input  $\mathbf{x}$  of the residual block.

#### 3.2.2. Gated Recurrent Unit (GRU)

GRU is one of the predictors of the ensemble model in this study. The classical LSTM solves the problem of long-term dependencies of Recurrent Neural Networks (RNN). However, its complex structure reduces the efficiency. Therefore, GRU was proposed in 2014 as a simpler design of RNN with the accuracy of original RNN maintained and its efficiency improved [58,59]. There are only two gate structures, reset gate and update gate (a combination of the forgetting gate and the input gate), in a GRU network, which reduces the parameter numbers significantly and improves the model efficiency to a great extent. The ratio between the transmission and retention of information in the past moment is determined by the update gate and the reset gate jointly, and the mathematical expressions of the reset gate and update gate are shown in the following formulas:

The update gate  $z_t$  is:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

The reset gate  $r_t$  is:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

where  $x_t$  is the  $t$ -th component of the input sequence,  $h_{t-1}$  is the information of the previous time step,  $W^{(z)}$ ,  $W^{(r)}$ ,  $U^{(z)}$  and  $U^{(r)}$  are the weight matrices, and the sigmoid function  $\sigma(x)$  is:

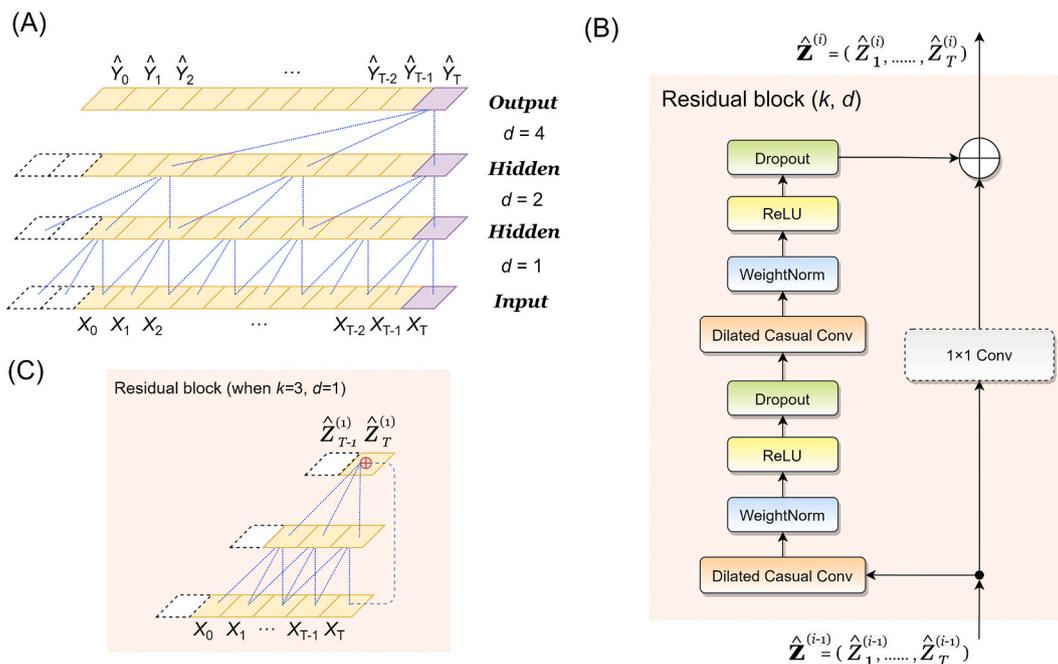
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Due to the introduction of the sign function, the values of  $z_t$  and  $r_t$  could only be 0 or 1, which indicates the switch state of the corresponding gate. To put it simply, if  $z_t = 0$ , the update gate is closed and the information in the initial time step can be kept.

Meanwhile, referring to the stored historical data, the reset gate could be calculated as follows:

$$h'_t = \tanh(Wx_t + r_t \odot U h_{t-1})$$

where  $W$  and  $U$  are the weight matrices. The information content could



**Fig. 3.** Architecture and elements in a TCN [57]. (A) A dilated casual convolution where the dilation factors  $d = 1, 2, 4$  and filter size is 3; (B) The residual block of TCN, where a unit convolution ( $1 \times 1$  conv) is added to adapt the model structure to situations that the residual input is of different dimension with the output; (C) A possible residual connection in a TCN, where the solid purple lines are filters and the dashed purple lines are identity mappings. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

be retained or forgotten, which is determined by calculating the Hadamard product of  $r_t$  and  $Uh_{t-1}$ .

### 3.2.3. Deep Belief Networks (DBN)

In this paper, the DBN is one of the predictors of the ensemble model. The essence of DBN is a stacking of a series of well-trained Restricted Boltzmann Machine (RBM). Boltzmann machines are a large class of neural network models, but the most used in practical applications is RBM, a specific type of Markov Random Fields (MRFs). The model of RBM is a two-layer simple neural network, so it cannot be regarded as the category of deep learning in the strict sense. However, as a stack of RBMs, DBM can be regarded as the promotion of RBM. In an RBM, visible variables  $\mathbf{v}$  are connected to stochastic hidden units  $\mathbf{h}$  using undirected weighted connections [60]. The aim of RBM training is to get a favored probability distribution  $P(\mathbf{v}, \mathbf{h})$  described by an energy function  $E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})$  where  $\boldsymbol{\theta}=(\boldsymbol{\omega}, \mathbf{b}, \mathbf{a})$  denotes a parameter set (where  $\omega_{ij}$  denotes the symmetric weight between visible unit  $i$  and hidden unit  $j$ ,  $b_i$  is the bias of  $i$ -th visible unit  $v_i$ , and  $a_j$  is the bias of  $j$ -th hidden unit  $h_j$ ) which could be described as (for a binary RBM [61]):

$$-\log P(\mathbf{v}, \mathbf{h}) \propto E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) = - \sum_{i=1}^{|\mathbf{V}|} \sum_{j=1}^{|\mathbf{H}|} \omega_{ij} v_i h_j - \sum_{i=1}^{|\mathbf{V}|} b_i v_i - \sum_{j=1}^{|\mathbf{H}|} a_j h_j$$

where  $|\mathbf{V}|$  and  $|\mathbf{H}|$  are the numbers of visible and hidden units, respectively. Thus, supposing the  $\mathbf{v}$  or  $\mathbf{h}$  is fixed, the conditional probability distributions of  $p(h_j|\mathbf{v}; \boldsymbol{\theta})$  and  $p(v_i|\mathbf{h}; \boldsymbol{\theta})$  could be calculated as [61]:

$$p(h_j|\mathbf{v}; \boldsymbol{\theta}) = \sigma \left( \sum_{i=1}^{|\mathbf{V}|} \omega_{ij} v_i + a_j \right)$$

$$p(v_i|\mathbf{h}; \boldsymbol{\theta}) = \sigma \left( \sum_{j=1}^{|\mathbf{H}|} \omega_{ij} h_j + b_i \right)$$

where the sigmoid function  $\sigma(x)$  is defined in previous section, and the parameters  $\boldsymbol{\theta} = (\boldsymbol{\omega}, \mathbf{b}, \mathbf{a})$  of each RBM could be learned from training. The training process is based on the contrastive divergence (CD) algorithm raised by G. Hinton [62]. Stacking the obtained RBMs together, we get a DBN. For each RBN, we have a parameter set  $\boldsymbol{\theta} = (\boldsymbol{\omega}, \mathbf{b}, \mathbf{a})$  that defines the distribution of possibility  $p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})$  and prior possibility  $p(\mathbf{h}; \boldsymbol{\theta})$  [63]. In this way, the possibility of generating visible variables is:

$$p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{h}|\boldsymbol{\theta}) p(\mathbf{v}|\mathbf{h}, \boldsymbol{\theta})$$

where the  $p(\mathbf{v}|\mathbf{h}, \boldsymbol{\theta})$  is fixed when parameter set  $\boldsymbol{\theta}$  is obtained from an RBM, and  $p(\mathbf{h}|\boldsymbol{\theta})$  could be replaced by a consecutive RBM, where the hidden layer of the previous RBM could be regarded as visible data.

### 3.2.4. Q-learning

As an online learning approach, reinforcement learning (RL) is different from supervised/unsupervised learning. During the process of interaction with the environment, the model obtains the optimal decision through trial-and-error, and then obtains the optimal result [65] (see Fig. 4). As a widely-used RL algorithm in feature selection, driver-less, route planning, and other fields, the Q-learning algorithm was proposed by Watkins et al., in 1989 [64]. Considering its good convergence and strong decision-making ability, the Q-learning method is applied as an ensemble learning method in this study, i.e., the Q-learning method is used to integrate three deep networks.

The steps of the ensemble method based on reinforcement learning are shown as follows:

**Step 1:** Build the state matrix  $S$  and the action matrix  $a$ , where the state matrix  $S$  denotes the weights of the three deep networks in the ensemble model, and the action matrix  $a$  is the weight adjustment action.

$$S = [w_1, w_2, w_3]$$

$$a = [\Delta w_1, \Delta w_2, \Delta w_3]$$

where  $w_1$  is the weight of the TCN network,  $w_2$  is the weight of the GRU network, and  $w_3$  is the weight of the DBN network.  $\Delta w_i$  ( $i = 1, 2, 3$ ) in action matrix  $a$  represent the weight change of the deep networks.

**Step 2:** Construct the Loss function  $L$ , the reward  $R$ , and the function  $Q$  for evaluation. In this study, the optimization goal was to minimize the output RMSE value. Therefore, the evaluation function  $Q$  is defined as:

$$Q = \text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - y'_i)^2}$$

where  $y_i$  is the actual value, and  $y'_i$  is the forecasting value.

**Step 3:** Train the agent (namely the ensemble model) based on the training sets of three kinds of deep network. According to the current state  $S$ , the agent performs an action  $a$ . During this process, the action is selected based on the  $\epsilon$ -greedy policy as [66]:

$$a_m = \begin{cases} \text{Based on } Q_{\max} & (p = 1 - \epsilon) \\ \text{Random} & (p = \epsilon) \end{cases}$$

where the parameter  $\epsilon \in (0, 1)$  is the exploration probability.

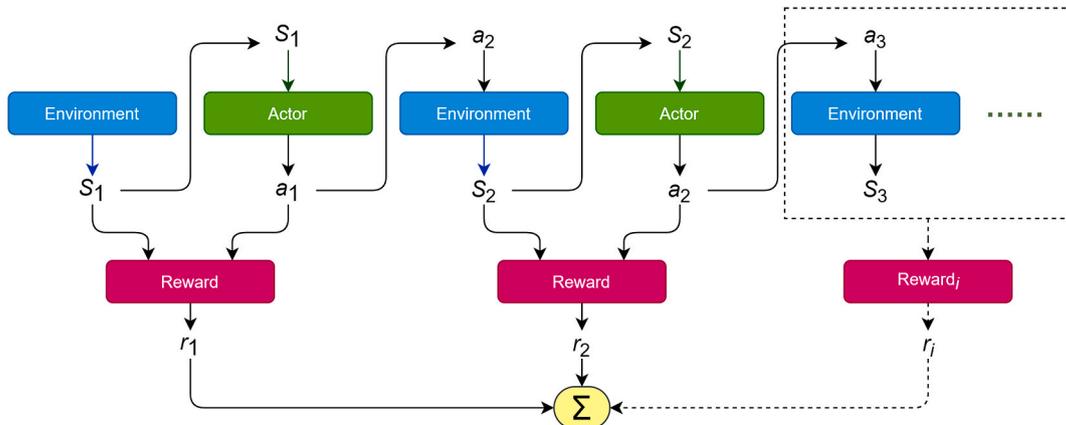


Fig. 4. The principle of Q-learning [64].

**Step 4:** Calculate the loss function  $L$ , get the reward  $R$ , and develop the next step strategy.

$$L = \frac{1}{n} \sum_{m=1}^n (A(m) - \hat{A}(m))^2$$

$$R = \begin{cases} +1 + L_m - L_{m+1} & (L_{m+1} < L_m) \\ -1 + L_m - L_{m+1} & (L_{m+1} > L_m) \end{cases}$$

where  $A(m)$  is the measured wind speed data in the training set,  $\hat{A}(m)$  is the forecasted COVID-19 infection data in the training set.

**Step 5:** Calculate the evaluation function  $Q$ , and update the Q-table [67].

$$Q_{m+1}(S_m, a_m) = Q_m(S_m, a_m) + \gamma_m (R(S_m, a_m) + \lambda \max_{a_{m+1}} Q_m(S_{m+1}, a_{m+1}) - Q_m(S_m, a_m))$$

where  $\lambda$  is the discount parameter, and  $\gamma$  is the learning rate.

**Step 6:** Repeat steps 3 to 5 until the iteration stop condition is satisfied. The state matrix  $S$  currently is the optimal weight of three deep networks.

**Step 7:** Input the test set into three well-trained deep networks to obtain the final prediction results. Then, the prediction results of the three deep networks are multiplied by the weight, and then they are ensemble together to obtain the final prediction result.

### 3.2.5. Support Vector Machine (SVM)

In this study, SVM is applied as a tool for error prediction, which further improves generalization ability and the accuracy of the final ensemble hybrid model. As a classical soft computing learning algorithm, SVM is widely-adopted in regression analysis, classification, pattern recognition and forecasting [68]. Based on the theory proposed by Vapnik [68,69], suppose a data series  $\{x_i, d_i\}_i^n$  ( $n$  is the data size,  $x_i$  is the input space vector,  $d_i$  represents the target value), SVM estimates the function represented in following equations:

$$f(\mathbf{x}) = k\varphi(\mathbf{x}) + \mathbf{b}$$

$$R_{SVM}(C) = \frac{1}{2}k^2 + \frac{C}{n} \sum_{i=1}^n L(x_i, d_i)$$

where  $\varphi(\mathbf{x})$  denotes the high-dimensional space feature that plots the input space vector  $\mathbf{x}$ ,  $\mathbf{b}$  represents the scalar,  $k$  is a normal vector, and  $\frac{C}{n} \sum_{i=1}^n L(x_i, d_i)$  denotes the empirical error. The positive slack variables  $\xi_i$  and  $\xi_i^*$  denotes the upper and lower excess deviation. By minimalizing the regularized risk  $R_{SVM}(C)$ , the scalar  $\mathbf{b}$  and the space feature  $\varphi(\mathbf{x})$  could be obtained.

In this way, the object of building an SVM is to:

$$\text{Minimize } R_{SVM}(k, \xi^{(*)}) = \frac{1}{2}\|k\|^2 + C \sum_{(i=1)}^n (\xi_i + \xi_i^*)$$

$$\text{s.t. } \begin{cases} d_i - k\varphi(x_i) + b_i \leq \epsilon + \xi_i \\ k\varphi(x_i) + b_i - d_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, m \end{cases}$$

Here,  $k^2/2$  is the regularization term, the quantity of features in the training dataset is  $l$ . In order to control the difference between the empirical error and regularization term, the error penalty factor  $C$  is introduced in the object function.  $\epsilon$  represents the loss function determined by approximation precision of the training set.

Given the optimality constraints, the problem  $f(\mathbf{x}) = k\varphi(\mathbf{x}) + \mathbf{b}$  could be solved by Lagrange multiplier. A generic function could be obtained

by following formula:

$$f(x, \omega_i, \omega_i^*) = \sum_{i=1}^n (\omega_i - \omega_i^*) \cdot K(x, x_i) + b$$

where the  $K(x_i, x_j)$  is kernel function and  $K(x, x_i) = \varphi(x_i)\varphi(x_j)$  is product of  $x_i$  and  $x_j$  inner vectors in the feature spaces  $\varphi(x_i)$  and  $\varphi(x_j)$ , respectively.

## 4. Experiment

### 4.1. Dataset

We used a dataset containing a daily situation update on COVID-19 and the global geographical distribution collected and provided by Epidemic Intelligence team of European Centre for Disease Prevention and Control (ECDC). Since the beginning of the COVID-19 pandemic, ECDC's Epidemic Intelligence team has been collecting the number of COVID-19 cases and deaths daily, based on reports from health authorities worldwide. To ensure the accuracy and reliability of the data, this process is being constantly refined. Every day between 6:00 and 10:00 CET, a team of epidemiologists screens up to 500 relevant sources to collect the latest figures. The data screening is followed by ECDC's standard epidemic intelligence process for which every single data entry is validated and documented in an ECDC database. An extract of this database, complete with up-to-date figures and data visualizations, is then shared on the ECDC website, ensuring a maximum level of transparency [70]. This study used 300 daily data of national cumulative infection numbers from India, the United Kingdom (UK), and the United States (US) from February 19, 2020 to December 14, 2020 (the data were accessed at 19:00 CST, January 19, 2021), and divided the data into three at a ratio of 3:1:1, which are training set (2020/2/19–2020/8/16, 180 days), validation set (2020/8/17–2020/10/15, 60 days), and test set (2020/10/16–2020/12/14, 60 days). The specific conditions of each data set are shown in Table 2(A). The training set is used to train the three neural networks of TCN/GRU/DBN, and the validation set is used to train the integrated model based on Q reinforcement learning and the error correction model based on SVM.

For the validation of the ensemble model, we adopted the Day Forward-chaining, a nested cross-validation method that is suitable for time-series data [71,72]. Day forward-chaining method is essential to keep the sequence of time-series data and prevent the possible information leakage that will caused by the k-fold cross validation. The concrete split of data is shown in Table 2(B).

**Table 2**  
Data information [70].

| (A) Content and split of data                            |                     |                      |                       |                       |
|--|---------------------|----------------------|-----------------------|-----------------------|
| Nation   | Training set (60%)  | Validation set (20%) | Test set (20%)        |                       |
| India  | 2020/2/19–2020/8/16 | 2020/8/17–2020/10/15 | 2020/10/16–2020/12/14 |                       |
| UK   | 16 (180 days)       | 15 (60 days)         | 12/14 (60 days)       |                       |
| US   |                     |                      |                       |                       |
| (B) Split of data in the day forward-chaining validation |                     |                      |                       |                       |
| #1   | #2                  | #3                   | #4                    | #5                    |
| 2020/2/19–2020/4/18                                      | 2020/4/19–2020/6/17 | 2020/6/18–2020/8/16  | 2020/8/17–2020/10/15  | 2020/10/16–2020/12/14 |
| (60 days)  | (60 days)           | (60 days)            | (60 days)             | (60 days)             |
| Training set   | Validation set      | Test set             | /                     | /                     |
| Training set   | Training set        | Validation set       | Test set              | /                     |
| Training set   | Training set        | Training set         | Validation set        | Test set              |

## 4.2. Performance evaluation indices

We adopted four widely-used and well-acknowledged indices to comprehensively evaluate the prediction performance of the proposed model. They were the mean absolute error (MAE), the mean absolute percentage error (MAPE%), the root mean squared error (RMSE), and the Pearson correlation coefficient (PCC).

For the first three indices (MAE, MAPE%, RMSE), the lower the values were, the better the prediction effect of the model was. As for the PCC, it is a commonly-used statistic to reflect the degree of linear correlation between two variables. The value range of PCC is  $[-1,1]$ , and the closer the absolute value of PCC is to 1, the stronger the linear correlation between the two variables is. In this study, PCCs were calculated to evaluate the correlation between the predicted number and the actual number. The calculation of RMSE is defined before, and the other three indices could be calculated as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - y'_i|$$

$$MAPE\% = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - y'_i}{y_i} \right| \times 100\%$$

$$PCC = \frac{\sum_{i=1}^N (y_i - \bar{y}_i)(y'_i - \bar{y}'_i)}{\sqrt{\sum_{i=1}^N (y_i - \bar{y}_i)^2} \cdot \sqrt{\sum_{i=1}^N (y'_i - \bar{y}'_i)^2}}$$

where  $N$  is the number of the samples,  $y_i$  is the actual value,  $\bar{y}_i$  is the mean value of the actual value,  $y'_i$  is the forecasting value, and  $\bar{y}'_i$  is the mean value of the forecasting value.

## 4.3. Model training

### 4.3.1. Determine the best input neuron numbers

The number of neurons in the input layer needs to be determined experimentally to ensure its matching with the prediction task. We used the data of the number of infected people in India for experiments and took different numbers of input neurons (3, 5, 7, 9) for experiments to determine an optimal number of input neurons. As for the experimental set-up, the essential parameters used in model training are given in Table 3.

Fig. 5 shows the prediction results and the corresponding model structures of models with different numbers of input neurons (see Fig. 4). To put it intuitively, we calculated the MAE, MAPE%, RMSE and PCC values of four investigated models with different input neuron numbers (shown in Table 4). As can be seen from Table 4, as the number of input neurons increases, the MAPE% value of the model shows a trend of first decreasing and then increasing. According to the four indices, when the number of input neurons is 5, the prediction effect of the model is the best. Therefore, in subsequent experiments, we set the number of input layers of each model as 5, that is, the data of the first 5 days are used to predict the data of the next day.

### 4.3.2. Model integration

The model proposed in this paper is mainly composed of three parts: predictor, optimizer, and error correction. The RMSE is used as the error when the model is trained. Fig. 6 (A) shows the error iteration diagram of the model during training, and Fig. 6(B)–(D) gives the prediction results of different components of the model on three data sets. Table 5 shows the weight optimization results of the three predictors of TCU, GRU and DBN optimized by reinforcement learning.

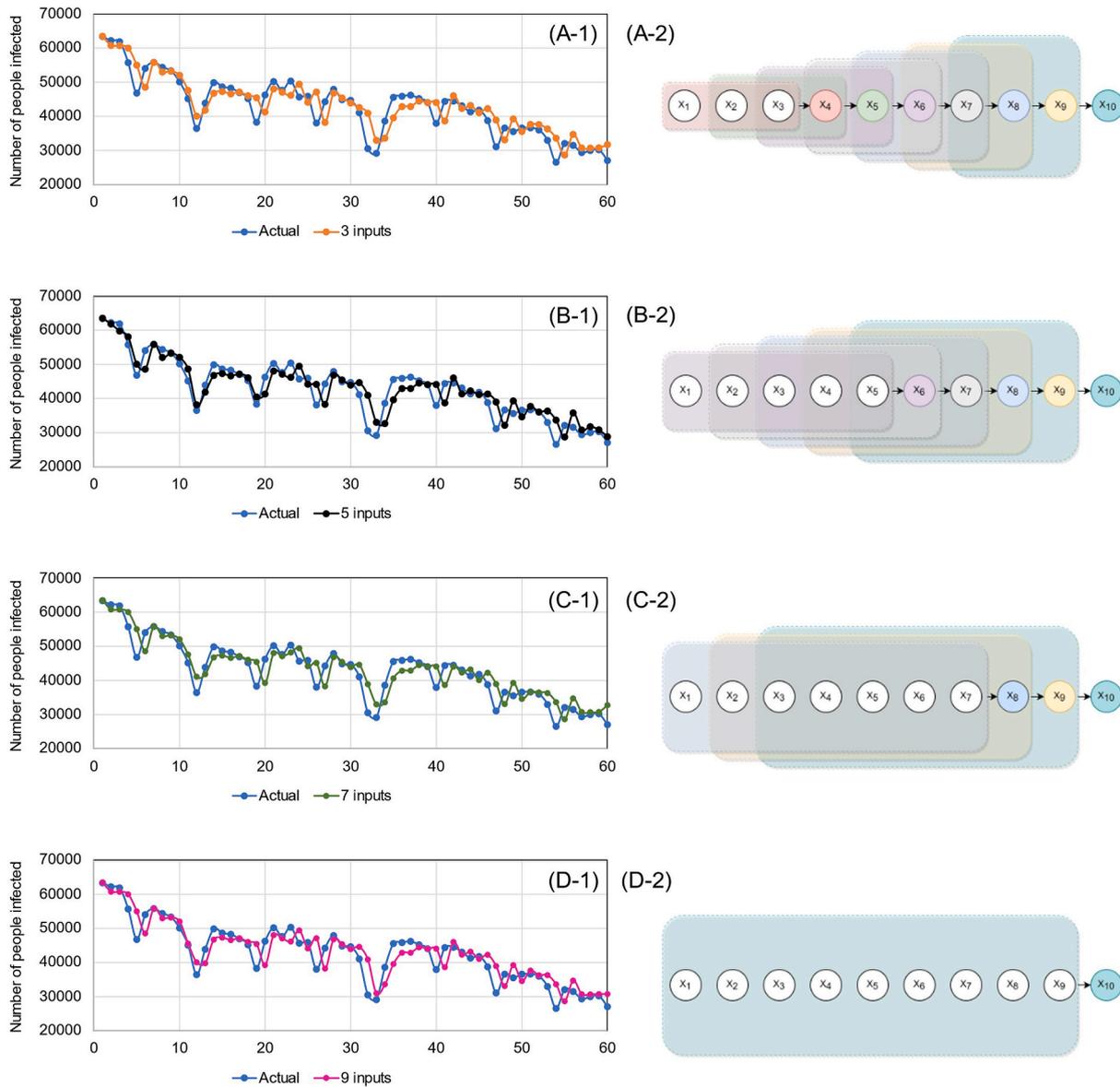
By analyzing Fig. 6 and Table 6, the following conclusions can be drawn:

**Table 3**

Essential parameters used in model training.

| Name of parameter           | Essential parameters |
|-----------------------------|----------------------|
| <b>Q-learning</b>           |                      |
| Maximum iteration           | 50                   |
| Learning rate               | 0.95                 |
| Discount parameter          | 0.5                  |
| <b>GRU</b>                  |                      |
| Size of input units         | 3/5/7/9              |
| Size of hidden units        | 100                  |
| Size of output units        | 1                    |
| Number of the Hidden layers | 16                   |
| Optimizer                   | Adam                 |
| Learning rate               | 0.01                 |
| Training epochs             | 200                  |
| <b>DBN</b>                  |                      |
| Size of input units         | 3/5/7/9              |
| Size of hidden units        | 20                   |
| Size of output units        | 1                    |
| number of the Hidden layer  | 1                    |
| Momentum factor             | 0                    |
| Optimizer                   | Adam                 |
| Learning rate               | 0.01                 |
| Training epochs             | 200                  |
| <b>TCN</b>                  |                      |
| Size of input units         | 3/5/7/9              |
| Size of hidden units        | 60                   |
| Size of output units        | 1                    |
| number of the Hidden layer  | 6                    |
| Learning rate               | 0.01                 |
| Optimizer                   | Adam                 |
| Filter size                 | 2                    |
| Training epochs             | 100                  |
| Dropout                     | 0.05                 |
| <b>SVM</b>                  |                      |
| Size of input units         | 3/5/7/9              |
| Size of output units        | 1                    |
| Kernel function             | RBF                  |
| Gamma                       | 10                   |
| $\sigma^2$                  | 20                   |

- (1) It can be seen from the results in the table that the prediction accuracy of TCN, GRU, and DBN is not much different on the same data set, but there is still a certain difference in the prediction accuracy between each other, and the performance of TCN, GRU, and DBN varies on different data sets, which indicates the deficiency of a single model in the generalization ability. For example, on the UK data set, TCN (MAE: 2266.439) and GRU (MAPE%: 11.988, RMSE: 3022.743, PCC: 0.707) has better performance than DBN, but DBN has the best performance on the Indian dataset (MAE: 3485.223, MAPE%: 8.861, RMSE: 4563.702), which shows that the performance of a single predictor is not stable enough and is related to specific data sets.
- (2) Basically, the proposed hybrid ensemble model obtained by using Q-learning to integrate the TCN, GRU and DBN achieved higher prediction accuracy than a single component on the three data sets, which shows that the integration method proposed in this paper is effective, as can be seen from Table 6. By adjusting the weight of each predictor in the component, the model can adjust the weight adaptively according to the characteristics of the data set, integrate the advantages of each sub-predictor, and achieve the improvement of the overall accuracy. This feature of the model also improves the robustness of the model, which enables the proposed model to achieve high-precision prediction results on various data sets. Additionally, comparing different data sets, we can see that the accuracy change of the integrated model on the three data sets is lower than its single components (TCN, GRU, and DBN), which indicates that the integrated model has a better stability than single model.
- (3) In order to further improve the accuracy of the integrated model, it is feasible to adopt the method of error compensation. As can be



**Fig. 5.** The prediction results (X-1) and the model structure (X-2) (X = A, B, C, D) for different numbers of input neurons: (A) three input neurons; (B) five input neurons; (C) seven input neurons; (D) nine input neurons.

**Table 4**  
Comparative results of performance indices from models with different input neuron numbers.

| Number of inputs | MAE             | MAPE%           | RMSE            | PCC             |
|------------------|-----------------|-----------------|-----------------|-----------------|
| 3                | 2970.198        | 7.649074        | 3859.494        | 0.897752        |
| <b>5</b>         | <b>2744.219</b> | <b>7.064694</b> | <b>3527.817</b> | <b>0.913801</b> |
| 7                | 2934.2          | 7.564229        | 3773.753        | 0.902113        |
| 9                | 2982.553        | 7.618867        | 3906.756        | 0.894298        |

seen from the results in the table, the accuracy of the model has been further improved by adding the error correction mechanism based on SVM, and the optimal performance has been achieved on each data set.

(4) In the ablation study, the weights of three sub-model are 1/3. Namely the effect of Q-learning in weight optimization was

cancelled. Compared to TCN-GRU-DBN-Q model that the weights of each output sequence ( $X_1, X_2, X_3$  given by TCN, GRU, DBN, respectively) are determined by Q-learning, the model output  $O$  in ablation study could be denoted as:

$$O = \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3$$

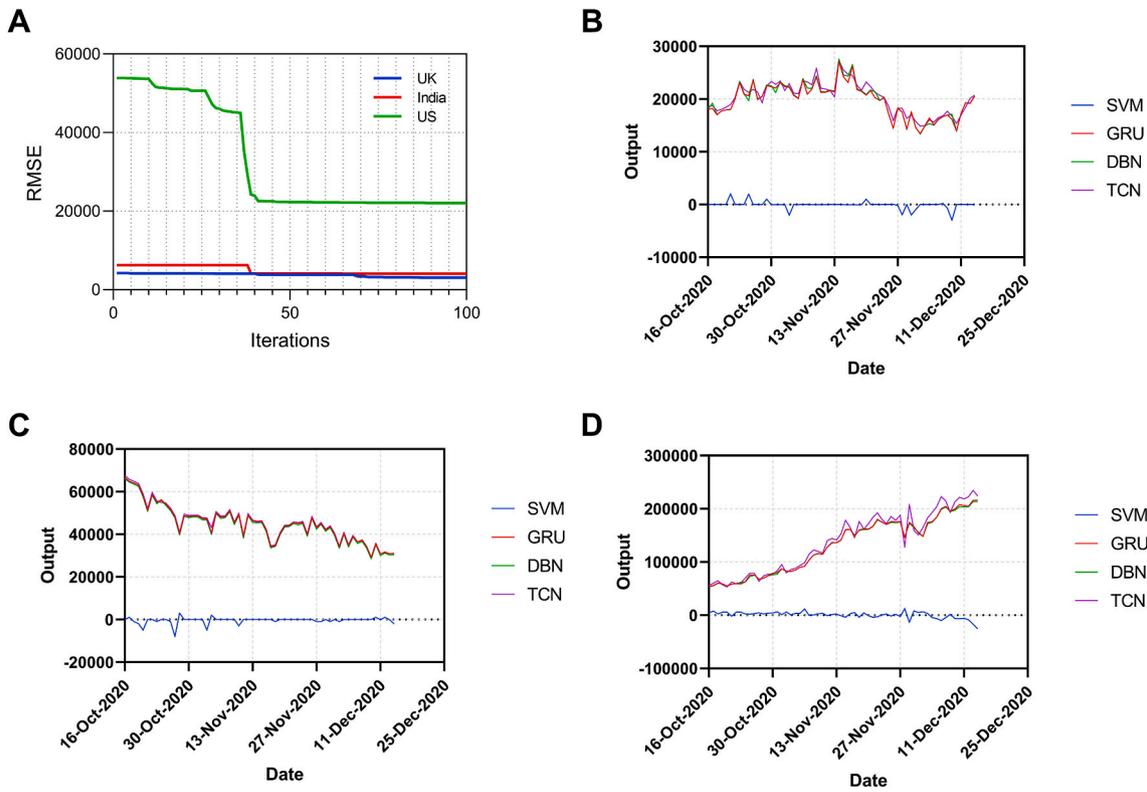


Fig. 6. (A) Model training processes (object function: RMSE, data: the number of infected people in UK/India/US); (B) to (D) Predictive output of elements (GRU, DBN, TCN) and error output predicted by SVM: (B) UK; (C) India; (D) US.

Table 5  
The weights of ensemble model (TCN, GRU, DBN) determined by Q-learning.

| Weights | $w_1$ (TCN) | $w_2$ (GRU) | $w_3$ (DBN) |
|---------|-------------|-------------|-------------|
| UK      | 0.46557     | 0.16110     | 0.29218     |
| India   | 0.32941     | 0.65251     | 0.02186     |
| USA     | 0.08401     | 0.21992     | 0.74092     |

It can be seen that compared the TCN-GRU-DBN-Q model, the ensemble model constituted by equal weights of TCN, GRU, DBN has a bigger forecast error, where the MAE, MAPE%, RMSE in each task are generally higher, and the PCC is generally lower than the model with Q-learning weight optimization. Meanwhile, it can be seen that in some tasks the model in ablation study performed less well than some sub-models (e.g., UK:  $MAE_{TCN-GRU-DBN-Q}$  (2237.399) <  $MAE_{TCN}$  (2266.439) <  $MAE_{Ablation}$  (2274.252) <  $MAE_{GRU}$  (2292.62) <  $MAE_{DBN}$

Table 6  
The performance indices of models in case studies<sup>†</sup>.

| Nations | Indices | Models    |           |           |               |                 |                   |  |
|---------|---------|-----------|-----------|-----------|---------------|-----------------|-------------------|--|
|         |         | TCN       | GRU       | DBN       | TCN-GRU-DBN-Q | Ablation study* | TCN-GRU-DBN-Q-SVM | Forward-chaining validation <sup>#</sup> |
| UK      | MAE     | 2266.439  | 2292.62   | 2325.047  | 2237.399      | 2274.252        | 1952.113          | 1963.548                                 |
|         | MAPE%   | 12.176    | 11.988    | 12.228    | 11.727        | 12.029          | 9.955             | 10.027                                   |
|         | RMSE    | 3097.349  | 3022.743  | 3078.515  | 3000.013      | 3029.000        | 2779.902          | 2801.472                                 |
|         | PCC     | 0.692     | 0.707     | 0.694     | 0.712         | 0.704           | 0.76              | 0.756                                    |
| India   | MAE     | 3794.701  | 3643.962  | 3485.223  | 3194.219      | 3617.004        | 2744.219          | 2720.219                                 |
|         | MAPE%   | 9.653     | 9.297     | 8.861     | 8.18          | 9.215           | 7.065             | 6.923                                    |
|         | RMSE    | 4955.327  | 4751.091  | 4563.702  | 4154.917      | 4246.751        | 3527.817          | 3461.18                                  |
|         | PCC     | 0.882     | 0.88      | 0.88      | 0.881         | 0.881           | 0.914             | 0.921                                    |
| US      | MAE     | 15813.502 | 16842.525 | 17068.942 | 15519.84      | 16238.379       | 13760             | 13664                                    |
|         | MAPE%   | 12.005    | 12.399    | 12.528    | 11.862        | 12.098          | 10.722            | 10.784                                   |
|         | RMSE    | 22489.499 | 22363.878 | 22447.302 | 22655.396     | 21821.405       | 20602.581         | 20772.94                                 |
|         | PCC     | 0.925     | 0.924     | 0.919     | 0.919         | 0.925           | 0.929             | 0.922                                    |

(2325.047)), which further proved the effectiveness of Q-learning in the optimization of sub-model weights.

(5) Day Forward-chaining validation results are recorded in Table 6 as well. The MAE, MAPE%, RMSE, and PCC values were calculated as the average value of three models trained with the split of data shown in Table 2(B). It can be seen that the metrics of the model remain largely stable, which proves the generalization ability of the proposed model.

In summary, it can be known that the model integration method proposed in this paper is effective. Each component of the model improves the performance of the model. With their cooperation, the integrated model can provide high-precision COVID-19 prediction results.

## 5. Results and Discussions

### 5.1. Case studies

In order to verify the accuracy of the model, as mentioned in the previous section, this paper uses actual case data from three countries, including the United States, India, and the United Kingdom, for experiments. The prediction results of the model are shown in Fig. 7 and Table 7.

As can be seen from Fig. 7 and Table 7, on all data sets, the model proposed in this paper can accurately predict the increase in the number of new crown cases in the data set. It has achieved the highest prediction accuracy in the USA, and its MPAE value has reached 7.06. This shows that the model proposed in this article has a strong practicability and can play a good role in assisting decision-making in the prevention and control of the COVID-19 epidemic. Meanwhile, in order to verify the effectiveness of each component of the integrated model, Table 6 compares the prediction results of each component of the model (TCN, GRU, DBN, TCN-GRU-DBN-Q) with the final hybrid model (TCN-GRU-DBN-Q-SVM), and an ablation study was conducted with model of  $0.333 \cdot \text{TCN} - 0.333 \cdot \text{GRU} - 0.333 \cdot \text{DBN}$  to verify the effectiveness of Q-learning in weights optimization. The results, as described in Section 4.3.2, demonstrated the effectiveness of ensemble and error compensation strategy.

### 5.2. Contrast study

In order to prove the advancement and superiority of the proposed TCN-GRU-DBN-Q-SVM algorithm, this paper compares it with two classic models (LSTM [73] and ANFIS [74]), three state-of-the-art models (VMD-BP [75], N-Beats [76], and WT-RVFL [77]), and time series analysis methods (ARIMA). In addition, in order to prove that residual prediction can effectively improve the model's comprehensive prediction and data analysis accuracy, the proposed TCN-GRU-DBN-Q-SVM was compared with TCN-GRU-DBN-Q. Fig. 8 and Table 8 show the comparison results of the models.

Through the contrast study, we can see that model 1 (the proposed

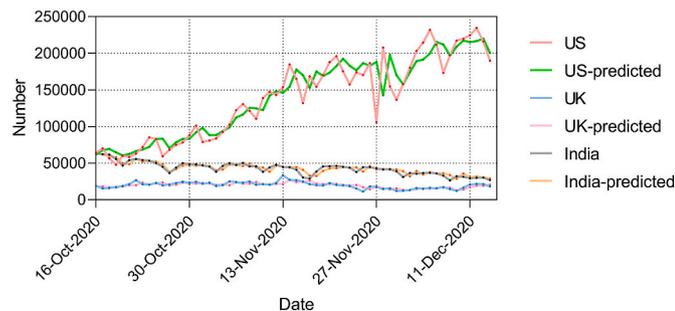


Fig. 7. Case studies: infection prediction of the UK, India, and the US.

Table 7

The performance indices in case studies.

| Nation | MAE      | MAPE% | RMSE     | PCC  |
|--------|----------|-------|----------|------|
| UK     | 1952.11  | 9.95  | 2779.90  | 0.76 |
| India  | 13760.00 | 10.72 | 20602.58 | 0.93 |
| US     | 2744.22  | 7.06  | 3527.82  | 0.91 |

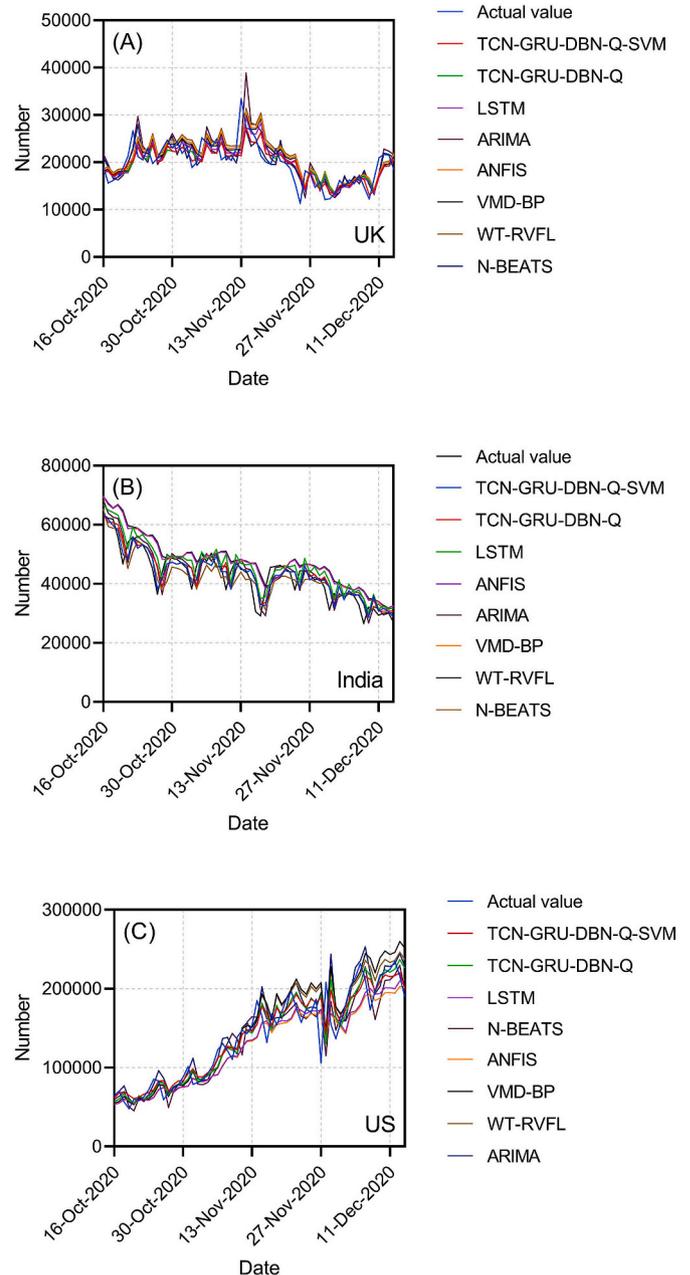


Fig. 8. Contrast studies of different models (TCN-GRU-DBN-Q-SVM, TCN-GRU-DBN-Q, LSTM, N-BEATS, ANFIS, VMD-BP, WT-RVFL, ARIMA) used for infection prediction in the UK, India, and the US.

hybrid model) has the best accuracy, which shows the superiority of the proposed model. From the results, it can be found that for the number of infections in the UK, India, and the US, model 1 (TCN-GRU-DBN-Q-SVM) has a higher prediction accuracy than model 2 (TCN-GRU-DBN-Q) ( $MRSE_1 < MRSE_2$ ,  $MAE_1 < MAE_2$ ,  $MAPE\%_1 < MAPE\%_2$ ,  $||PCC1|-1| < ||PCC2|-1|$ ), which proves that the establishment of an error prediction model is meaningful for improving the prediction accuracy.

**Table 8**  
Comparative results.

| Models                     | MAE            | MAPE%         | RMSE           | PCC           |
|----------------------------|----------------|---------------|----------------|---------------|
| <b>1 TCN-GRU-DBN-Q-SVM</b> | <b>1952.11</b> | <b>9.9545</b> | <b>2779.90</b> | <b>0.7601</b> |
| 2 TCN-GRU-DBN-Q            | 2237.32        | 11.7273       | 3000.01        | 0.7118        |
| 3 LSTM                     | 2337.70        | 12.2365       | 3102.24        | 0.7094        |
| 4 N-BEATS                  | 2317.97        | 12.1520       | 2921.08        | 0.7357        |
| 5 ANFIS                    | 2578.31        | 13.7316       | 3345.64        | 0.7092        |
| 6 VMD-BP                   | 2465.31        | 13.0623       | 3233.25        | 0.7084        |
| 7 WT-RVFL                  | 2721.69        | 14.5113       | 3511.51        | 0.7087        |
| 8 ARIMA                    | 2733.65        | 14.0605       | 3492.70        | 0.7338        |

(2) India

| Models                     | MAE            | MAPE%         | RMSE           | PCC           |
|----------------------------|----------------|---------------|----------------|---------------|
| <b>1 TCN-GRU-DBN-Q-SVM</b> | <b>2744.22</b> | <b>7.0647</b> | <b>3527.82</b> | <b>0.9138</b> |
| 2 TCN-GRU-DBN-Q            | 3194.22        | 8.1796        | 4154.92        | 0.8811        |
| 3 LSTM                     | 3907.83        | 10.0347       | 5034.43        | 0.8804        |
| 4 N-BEATS                  | 3662.47        | 8.9759        | 4503.18        | 0.8594        |
| 5 ANFIS                    | 5062.78        | 12.9261       | 6609.05        | 0.8409        |
| 6 VMD-BP                   | 5042.04        | 12.8732       | 6602.21        | 0.8385        |
| 7 WT-RVFL                  | 5294.62        | 13.5146       | 6872.85        | 0.8316        |
| 8 ARIMA                    | 3531.50        | 8.7465        | 4571.09        | 0.8729        |

(3) US

| Models                     | MAE             | MAPE%          | RMSE            | PCC           |
|----------------------------|-----------------|----------------|-----------------|---------------|
| <b>1 TCN-GRU-DBN-Q-SVM</b> | <b>13760.00</b> | <b>10.7218</b> | <b>20602.58</b> | <b>0.9292</b> |
| 2 TCN-GRU-DBN-Q            | 15519.84        | 11.8625        | 22655.40        | 0.9194        |
| 3 LSTM                     | 17736.30        | 12.8808        | 23153.17        | 0.9246        |
| 4 N-BEATS                  | 16403.06        | 12.5258        | 22156.28        | 0.9205        |
| 5 ANFIS                    | 18629.92        | 13.2489        | 24059.56        | 0.9260        |
| 6 VMD-BP                   | 19578.87        | 14.3822        | 28475.36        | 0.9203        |
| 7 WT-RVFL                  | 17445.92        | 13.2414        | 25746.29        | 0.9227        |
| 8 ARIMA                    | 17100.31        | 13.2273        | 25639.68        | 0.9120        |

Furthermore, in order to verify the high forecast accuracy of the proposed hybrid model quantitatively, we propose the prediction performance indices improvement percentages  $P_{MAPE\%}$  (%),  $P_{MAE}$  (%),  $P_{RMSE}$  (%) and  $P_{PCC}$  (%) to compare and analyze the improvement of the prediction accuracy of the proposed model (TCN-GRU-DBN-Q-SVM) compared with the TCN-GRU-DBN-Q (proposed model without SVM error predictor), LSTM [73], ANFIS [74], VMD-BP [75], N-Beats [76], WT-RVFL [77] and ARIMA models (see Table 9). The specific calculation method is as follows:

$$\begin{cases} P_{MAPE\%} = \frac{MAPE\%_2 - MAPE\%_1}{MAPE\%_2} \times 100\% \\ P_{MAE} = \frac{MAE_2 - MAE_1}{MAE_2} \times 100\% \\ P_{RMSE} = \frac{RMSE_2 - RMSE_1}{RMSE_2} \times 100\% \\ P_{PCC} = \frac{PCC_1 - PCC_2}{PCC_2} \times 100\% \end{cases}$$

where  $MAPE\%_1$ ,  $MAE_1$ ,  $RMSE_1$  and  $PCC_1$  are forecasting performance

indices of the proposed model, while  $MAPE\%_2$ ,  $MAE_2$ ,  $RMSE_2$  and  $PCC_2$  are the indices of the comparison model.

- (1) The prediction result of TCN-GRU-DBN-Q-SVM is better than that of TCN-GRU-DBN-Q algorithm, which proves that the residual prediction modeling based on SVM can effectively improve the overall prediction ability of the model. The possible reason is that residual prediction analyzes the deviation information between the predictor and the real data to further correct the prediction results of the model and improve the accuracy comprehensively.
- (2) Although VMD-BP and WT-RVFL algorithms can achieve good prediction results, their prediction performance is difficult to surpass the classic model (ANFIS, LSTM, ARIMA), the recently proposed model (N-Beats) and the model proposed in this paper. The possible reason is that the decomposition algorithm has a certain boundary effect in the modeling process, which affects the model's ability to analyze and identify the original time series to a certain extent.
- (3) The TCN-GRU-DBN-Q-SVM proposed in this paper can achieve satisfactory prediction results in all case studies. Compared with the suboptimal model, the performance improvement of the

**Table 9**

The promoting percentages of the proposed model comparing to other experimental models.

| INDICES              | COMPARISON MODELS    | UK                   | INDIA  | US     |
|----------------------|----------------------|----------------------|--------|--------|
| $P_{MAPE}$ (%)       | Model 1 v.s. Model 2 | 12.75%               | 14.09% | 11.34% |
|                      | Model 1 v.s. Model 3 | 16.49%               | 29.78% | 22.42% |
|                      | Model 1 v.s. Model 4 | 18.08%               | 21.29% | 14.40% |
|                      | Model 1 v.s. Model 5 | 24.29%               | 45.80% | 26.14% |
|                      | Model 1 v.s. Model 6 | 20.82%               | 45.57% | 29.72% |
|                      | Model 1 v.s. Model 7 | 28.28%               | 48.17% | 21.13% |
|                      | Model 1 v.s. Model 8 | 29.20%               | 19.23% | 18.94% |
|                      | $P_{MAE}$ (%)        | Model 1 v.s. Model 2 | 15.12% | 13.63% |
| Model 1 v.s. Model 3 |                      | 18.65%               | 29.60% | 16.76% |
| Model 1 v.s. Model 4 |                      | 15.78%               | 25.07% | 16.11% |
| Model 1 v.s. Model 5 |                      | 27.51%               | 45.35% | 19.07% |
| Model 1 v.s. Model 6 |                      | 23.79%               | 45.12% | 25.45% |
| Model 1 v.s. Model 7 |                      | 31.40%               | 47.73% | 19.03% |
| Model 1 v.s. Model 8 |                      | 28.59%               | 22.29% | 19.53% |
| $P_{RMSE}$ (%)       |                      | Model 1 v.s. Model 2 | 7.34%  | 15.09% |
|                      | Model 1 v.s. Model 3 | 10.39%               | 29.93% | 11.02% |
|                      | Model 1 v.s. Model 4 | 4.83%                | 21.66% | 7.01%  |
|                      | Model 1 v.s. Model 5 | 16.91%               | 46.62% | 14.37% |
|                      | Model 1 v.s. Model 6 | 14.02%               | 46.57% | 27.65% |
|                      | Model 1 v.s. Model 7 | 20.83%               | 48.67% | 19.98% |
|                      | Model 1 v.s. Model 8 | 20.41%               | 22.82% | 19.65% |
|                      | $P_{pcc}$ (%)        | Model 1 v.s. Model 2 | 6.79%  | 3.71%  |
| Model 1 v.s. Model 3 |                      | 7.16%                | 3.80%  | 0.50%  |
| Model 1 v.s. Model 4 |                      | 3.32%                | 6.33%  | 0.95%  |
| Model 1 v.s. Model 5 |                      | 7.18%                | 8.67%  | 0.35%  |
| Model 1 v.s. Model 6 |                      | 7.30%                | 8.98%  | 0.98%  |
| Model 1 v.s. Model 7 |                      | 7.27%                | 9.88%  | 0.71%  |
| Model 1 v.s. Model 8 |                      | 3.58%                | 4.69%  | 1.89%  |

**Note:** Where Model 1 is TCN-GRU-DBN-Q-SVM (the proposed model), Model 2 is TCN-GRU-DBN-Q, Model 3 is LSTM, Model 4 is N-BEATS, Model 5 is ANFIS, Model 6 is VMD-BP, Model 7 is WT-RVFL, Model 8 is ARIMA. Based on the experimental results, the following conclusions can be drawn.

model on the three data sets is huge. The possible reasons of the performance improvement are: first, the three neural networks (TCN, GRU, DBN) used in this article have their own unique network frameworks, which improves their ability to extract deep features of time series. Second, the integrated learning algorithm based on Q-learning can comprehensively analyze the modeling capabilities of these types of neural networks for different time series to construct a satisfying hybrid ensemble model with improved comprehensive robustness and adaptability (or, generalization). Finally, SVM constructs a residual correction model to further optimize the result of the hybrid model by analyzing the deviation between the real data and the predicted value of the TCN-GRU-DBN-Q network. Therefore, the model proposed in this article has extremely high application value in the field of COVID-19 infection prediction.

## 6. Limitations and further work

The TCN-GRU-DBN-Q-SVM model in this paper has the following points that are still worth improving. First, as an ensemble model, the higher the variability of the sub-model, the higher the accuracy of the integrated model obtained in general. Therefore, a larger integrated model can be explored in the future. Although a larger model will take more time to train and forecast, it is feasible for COVID-19 morbidity prediction, a task that does not require high real-time performance (generally forecast in days). Second, integration with other modeling approaches (e.g., time series models such as ARIMA, STAR models, dynamic models such as SAHQD model [19], SEIAIR model [28], even the SCUAQIHMRD model [31]) can also be performed, where reinforcement learning can still provide an integration framework. Third, the data of other modalities, such as Mobility Report released by Google [78], search interest [51], local weather data, human contact data [33], etc., could also be considered as meaningful input into the model [79],

although the inaccessibility and incompleteness of some data modalities may limit the power and generalizability of the model with integrated multi-modalities. Additionally, as a deep learning model, there are some inherent drawbacks. First, the “features” decoded by the forecast model are abstract information, which are not sufficiently interpretable. Second, as with all data-driven models, the model relies on the accuracy of the data provided for model training, and the prediction step is not as long as dynamics modeling. However, the corresponding advantages are obvious: deep learning is more adequate for mining the information contained in the time series data, and does not rely on complicated (often multi-compartment, multi-stage, and multi-parameter) dynamic modeling that requires delicate analysis of epidemic situation ranging from the policy changes to the pathogenicity and crowd immunity, and accordingly requires less parameter estimation and manual feature extraction [80].

It must be noted that in the above elaboration, it is easy to see that for the task of forecasting the number of COVID-19 infections, there are various forecasting models, but their respective advantages and disadvantages are also apparent. Researchers need to make trade-offs and choose the appropriate prediction scheme according to the needs of prediction. The TCN-GRU-DBN-Q-SVM model involved in this study can give better case number forecasts than many existing models for scenarios where information such as temperature data, transportation data, and network human behavior is missing but only daily incidence case number information is available, but the forecast step size is limited.

In a word, the forecast models must keep up with a rapidly changing situation [80]. The modeling of COVID-19 pandemic is, in essential, a difficult trade-off (see Fig. 9). First, the data-driven models were easier to acquire compared to those dynamic model-based forecasting models. However, how to determine the ensemble strategy can vary when the training time requirements, hardware space, and the amount of training data change, although the ensemble model can achieve higher forecast performance than single ones. Second, the deep-learning models may sacrifice the interpretability of forecast model since the “features” they learn could be abstract and obscure, but the accuracy of deep-learning, if the data quantity permits, is generally higher than interpretable methods ranging from time-series models to regression models where the artificial feature engineering may be performed to acquire the features in data’s statistical, spectral, and temporal domain. Last but not least, the modality (or the source) of data should be flexibly and smartly determined since the ease of quantification and the accessibility to data may reduce when the modality and source of data become multiple and extensive.

## 7. Conclusion

The COVID-19 pandemic is posing a huge challenge to international public health. Accurate and effective prediction of the number of cases is significant for the health resource planning and the epidemic situation evaluation. Unlike SIR models, the data-driven machine learning model does not rely on accurate physical dynamics modeling and can adapt to complex changes in the epidemic situation (for example, vaccination, quarantine, isolation, lockdown, social distancing, etc.), and can be sensitive and accurate in predicting the development of the epidemic. In this work, a novel TCN-GRU-DBN-Q-SVM ensemble hybrid model is proposed for COVID-19 infection prediction. First, three widely used networks, TCN, GRU, and DBN, are used as single predictors. Second, three predictors are ensembled by reinforcement learning method (Q-learning) with different weights. Third, an error predictor built by SVM, is trained with validation set, and the final prediction result could be obtained by combining the TCN-GRU-DBN-Q model with the SVM error predictor.

The strengths of our model could be concluded as follows. First, we use multiple predictors to work collectively. The integrated weights based on Q reinforcement learning can be adaptively adjusted according to the characteristics of the data, which ensures model’s capability of

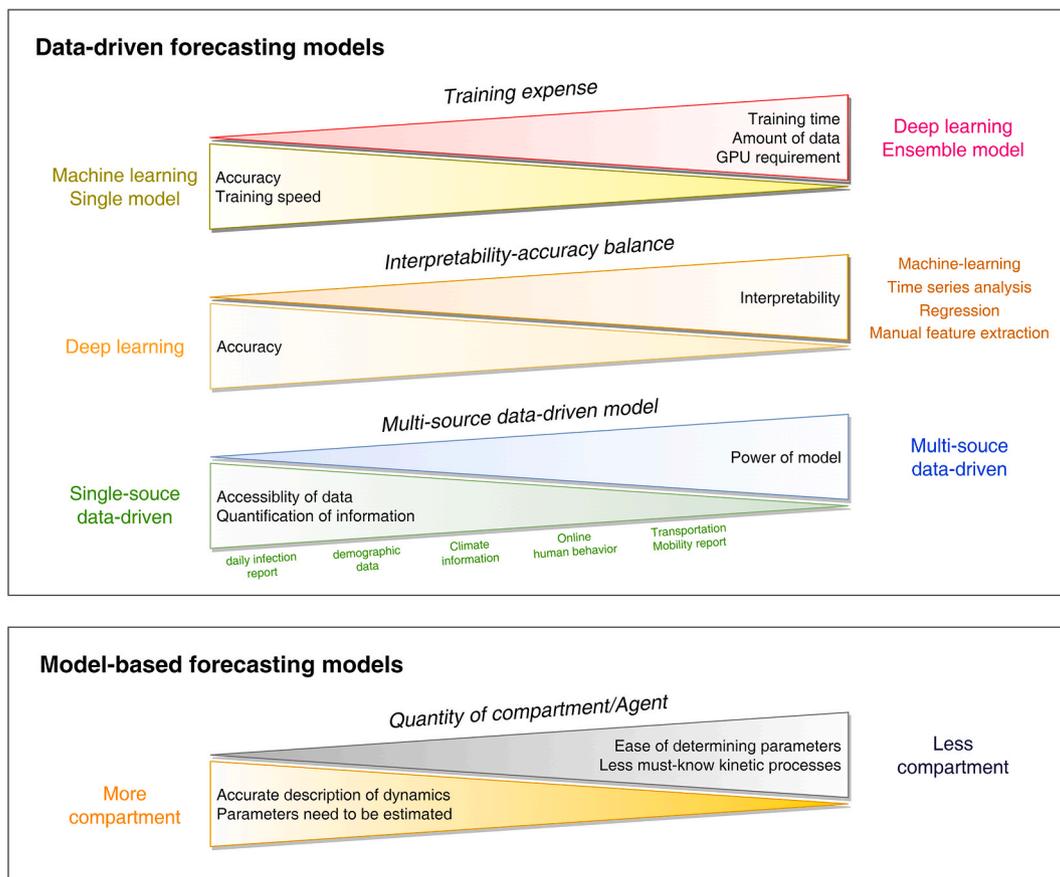


Fig. 9. Trade-offs to consider in data-driven and dynamics model-driven COVID-19 prevalence modeling.

various forecasting work in different countries and multiple situations (such as social-distancing, vaccination, migration, etc.). Therefore, the accuracy, robustness and generalization of the proposed model are ensured. Second, our model uses an SVM-based error compensation mechanism to further improve the accuracy of the model. Third, we use relatively newly-proposed deep learning network to ensure the accuracy of the model. Fourth, the proposed model is data-driven, and the amount of data required is easily met.

In the future, we will further consider whether the proposed integrated model can integrate more information modalities. For example, Wensen Huang et al. evaluated predictive value of regional outbreaks of online medical behavior data, including online medical consultation (OMC), online medical appointment (OMA) and online medical search (OMS) for the 2019 coronavirus disease in Shenzhen, China from January 1, 2020 to March 5, 2020 [81]. If this type of information model can be integrated with data-driven predictive models, and natural language processing (NLP) and other algorithms can be used to extract more information and merge into the model, it will have a certain significance for further improving the predictive effect of COVID-19. The integration with other modeling approaches is also very interesting. Sumit Mohan et al. have proposed a hybrid ARIMA and Prophet model to predict daily confirmed and cumulative confirmed cases in India, which is an inspiring step [82].

However, as we discuss in this paper, the exact modeling approach, data modality, and integration strategy needs to consider a number of different factors, including data sources, model generalization capabilities, model accuracy, and hardware conditions during model training, which is a difficult trade-off [80]. Researchers must put more effort to make machine learning, time series analysis, dynamical modeling, and other data science models more useful for COVID-19 forecasting to provide more accurate and reliable information for the outbreak

prevention and control.

**Declaration of competing interest**

Declare none.

**Acknowledgement**

The authors would like to thank European Centre for Disease Prevention and Control for the public database they provided on <https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data-daily-up-to-14-december-2020>.

**Abbreviations**

|          |  |
|----------|--|
| ANFIS    | Adaptive Neuro-Fuzzy Inference System    |
| AR       | autoregressive                           |
| ARIMA    | Autoregressive Integrated Moving Average |
| ARMA     | autoregressive moving average            |
| BP       | Back Propagation                         |
| CET      | Central European Time                    |
| CST      | Central Standard Time                    |
| CMPA     | chaotic marine predator algorithm        |
| CD       | contrastive divergence                   |
| COVID-19 | Corona Virus Disease 2019                |
| DBN      | Deep Belief Networks                     |
| GRU      | Gated Recurrent Unit                     |
| GRM      | growth rate model                        |
| HW       | Holt-Winters exponential smoothing       |
| MPA      | marine predator algorithm                |
| MRFs     | Markov Random Fields                     |

|         |   |
|---------|---|
| MA      | moving average  |
| OMA     | online medical appointment  |
| OMC     | online medical consultation   |
| OMS     | online medical search   |
| RL      | reinforcement learning  |
| RBM     | Restricted Boltzmann Machine  |
| RVFL    | random vector functional link   |
| SARIMA  | Seasonal Autoregressive Integrated Moving Average                         |
| SVM     | Support Vector Machine  |
| SEIR    | Susceptible-Exposed-Infected-Recovered Model                              |
| SIR     | Susceptible-Infected-Recovered Model                                      |
| TNC     | Temporal Convolutional Networks   |
| TCN     | Temporal Convolutional Networks   |
| MAE     | the mean absolute error   |
| MAPE%   | the mean absolute percentage error  |
| N-BEATS | Neural basis expansion analysis for interpretable time series forecasting |
| PCC     | the Pearson correlation coefficient                                       |
| RMSE    | the root mean squared error   |
| UK      | the United Kingdom  |
| US      | the United States   |
| VMD     | Variational Mode Decomposition  |
| WT      | wavelet transform   |

## References

- [1] S. Xu, Y. Li, Beware of the second wave of COVID-19, *Lancet* 395 (10233) (2020) 1321–1322, [https://doi.org/10.1016/S0140-6736\(20\)30845-X](https://doi.org/10.1016/S0140-6736(20)30845-X).
- [2] S. Dil, N. Dil, Z.H. Maken, COVID-19 trends and forecast in the Eastern mediterranean region with a particular focus on Pakistan, *Cureus* 12 (6) (2020) 6–13, <https://doi.org/10.7759/cureus.8582>.
- [3] C. Zhao, B. Tepekule, N.G. Criscuolo, et al., icumonitoring.ch: a platform for short-term forecasting of intensive care unit occupancy during the COVID-19 epidemic in Switzerland, *Swiss Med. Wkly.* 150 (March) (2020) w20277, <https://doi.org/10.4414/smww.2020.20277>.
- [4] C. Reno, J. Lenzi, A. Navarra, et al., Forecasting COVID-19-associated hospitalizations under different levels of social distancing in Lombardy and Emilia-Romagna, Northern Italy: results from an extended SEIR compartmental model, *J. Clin. Med.* 9 (5) (2020) 1492, <https://doi.org/10.3390/jcm9051492>.
- [5] D. Fanelli, F. Piazza, Analysis and forecast of COVID-19 spreading in China, Italy and France, *Chaos, Solit. Fractals* 134 (January) (2020), 109761, <https://doi.org/10.1016/j.chaos.2020.109761>.
- [6] K. Roosa, Y. Lee, R. Luo, et al., Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020, *Infect. Dis. Model.* 5 (2020) 256–263, <https://doi.org/10.1016/j.idm.2020.02.002>.
- [7] R.M. Chaudhry, A. Hanif, M. Chaudhary, et al., Coronavirus disease 2019 (COVID-19): forecast of an emerging urgency in Pakistan, *Cureus* 2019 (5) (2020), <https://doi.org/10.7759/cureus.8346>.
- [8] A.S. Ahmar, E.B. del Val, SutteARIMA: short-term forecasting method, a case: covid-19 and stock market in Spain, *Sci. Total Environ.* (2020) 729, <https://doi.org/10.1016/j.scitotenv.2020.138883>.
- [9] F. Qeadan, T. Honda, L.H. Gren, et al., Naive forecast for COVID-19 in Utah based on the South Korea and Italy models-the fluctuation between two extremes, *Int. J. Environ. Res. Publ. Health* 17 (8) (2020) 1–14, <https://doi.org/10.3390/ijerph17082750>.
- [10] M.H.D.M. Ribeiro, R.G. da Silva, V.C. Mariani, L. dos S. Coelho, Short-term forecasting COVID-19 cumulative confirmed cases: perspectives for Brazil, *Chaos, Solit. Fractals* 135 (2020), <https://doi.org/10.1016/j.chaos.2020.109853>.
- [11] R. Sujath, J.M. Chatterjee, A.E. Hassanien, A machine learning forecasting model for COVID-19 pandemic in India, *Stoch. Environ. Res. Risk Assess.* 34 (7) (2020) 959–972, <https://doi.org/10.1007/s00477-020-01827-8>.
- [12] K. Abdulmajeed, M. Adeleke, L. Popoola, Online forecasting of covid-19 cases in Nigeria using limited data, *Data Brief* 30 (2020), 105683, <https://doi.org/10.1016/j.dib.2020.105683>.
- [13] T. Chakraborty, I. Ghosh, Real-time forecasts and risk assessment of novel coronavirus (COVID-19) cases: a data-driven analysis, *Chaos, Solit. Fractals* 135 (2020), <https://doi.org/10.1016/j.chaos.2020.109850>.
- [14] V.K.R. Chimmula, L. Zhang, Time series forecasting of COVID-19 transmission in Canada using LSTM networks, *Chaos, Solit. Fractals* 135 (2020), <https://doi.org/10.1016/j.chaos.2020.109864>.
- [15] S.J. Fong, G. Li, N. Dey, R.G. Crespo, E. Herrera-Viedma, Finding an accurate early forecasting model from small dataset: a Case of 2019-nCoV novel coronavirus outbreak, *Int. J. Interact. Multimed. Artif. Intell.* 6 (2020) 132–140, <https://doi.org/10.9781/ijimai.2020.02.002>.
- [16] M.A.A. Al-Qaness, A.A. Ewees, H. Fan, MAEI Aziz, Optimization method for forecasting confirmed cases of COVID-19 in China, *Appl. Sci.* 9 (3) (2020), <https://doi.org/10.3390/JCM9030674>.
- [17] D. Parbat, M. Chakraborty, A python based support vector regression model for prediction of COVID19 cases in India, *Chaos, Solit. Fractals* 138 (January) (2020) 337–339, <https://doi.org/10.1016/j.chaos.2020.109942>.
- [18] I. Rahimi, F. Chen, A.H. Gandomi, A review on COVID-19 forecasting models, *Neural Comput. Appl.* 8 (2021), <https://doi.org/10.1007/s00521-020-05626-8>.
- [19] S. Bhattacharjee, S. Liao, D. Paul, S. Chaudhuri, Inference on the dynamics of COVID-19 in the United States, *Sci. Rep.* 12 (1) (2022) 1–15, <https://doi.org/10.1038/s41598-021-04494-z>.
- [20] K.E. Arunkumar, D.V. Kalaga, C.M. Sai Kumar, G. Chilkoor, M. Kawaji, T. M. Brenza, Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average, *Appl. Soft Comput.* 103 (December 2019) (2021), 107161, <https://doi.org/10.1016/j.asoc.2021.107161>.
- [21] C.J. Lynch, R. Gore, Short-range forecasting of coronavirus disease 2019 (COVID-19) during early onset at county, health district, and state geographic levels: comparative forecasting approach using seven forecasting methods (Preprint), *J. Med. Internet Res.* 10 (2020), <https://doi.org/10.2196/24925>. Published online October.
- [22] A.H. Elsheikh, A.I. Saba, M.A. Elaziz, et al., Deep learning-based forecasting model for COVID-19 outbreak in Saudi Arabia, *Process Saf. Environ. Protect.* 149 (2021) 223–233, <https://doi.org/10.1016/j.psep.2020.10.048>.
- [23] N. Talkhi, N.A. Fatemi, Z. Ataei, M.J. Nooghabi, Modeling and forecasting number of confirmed and death caused COVID-19 in Iran: a comparison of time series forecasting methods, *Biomed. Signal Process Control* 66 (October 2020) (2021), 102494, <https://doi.org/10.1016/j.bspc.2021.102494>.
- [24] N. Zheng, S. Du, J. Wang, et al., Predicting COVID-19 in China using hybrid AI model, *IEEE Trans. Cybern.* 50 (7) (2020) 2891–2904, <https://doi.org/10.1109/TCYB.2020.2990162>.
- [25] A. Abbas, M.M. Abdelsamea, M.M. Gaber, et al., Deep-COVID: predicting COVID-19 from chest X-ray images using deep transfer learning, *Chaos, Solit. Fractals* 140 (2) (2020), 110153, <https://doi.org/10.1016/j.media.2020.101794>.
- [26] H. Zhao, N.N. Merchant, A. McNulty, et al., COVID-19: short term prediction model using daily incidence data, *PLoS One* 16 (4 April) (2021) 1–14, <https://doi.org/10.1371/journal.pone.0250110>.
- [27] M.A. Kuddus, A. Rahman, Analysis of COVID-19 using a modified SLIR model with nonlinear incidence, *Results Phys.* 27 (June) (2021), 104478, <https://doi.org/10.1016/j.rinp.2021.104478>.
- [28] Z. Ma, S. Wang, X. Lin, et al., Modeling for COVID-19 with the contacting distance, *Nonlinear Dynam.* 107 (3) (2022) 3065–3084, <https://doi.org/10.1007/s11071-021-07107-6>.
- [29] T. Qiu, H. Xiao, V. Brusci, Estimating the effects of public health measures by SEIR (MH) model of COVID-19 epidemic in local geographic areas, *Front. Public Health* 9 (January) (2022) 1–17, <https://doi.org/10.3389/fpubh.2021.728525>.
- [30] X. Hao, S. Cheng, D. Wu, T. Wu, X. Lin, C. Wang, Reconstruction of the full transmission dynamics of COVID-19 in Wuhan, *Nature* 584 (7821) (2020) 420–424, <https://doi.org/10.1038/s41586-020-2554-8>.
- [31] L. Zhou, X. Rong, M. Fan, et al., Modeling and evaluation of the joint prevention and control mechanism for curbing COVID-19 in Wuhan, *Bull. Math. Biol.* 84 (2) (2022), <https://doi.org/10.1007/s11538-021-00983-4>.
- [32] E. Ma, M.A. Za, J. Ar, Forecasting Malaysia COVID-19 incidence based on movement control order using ARIMA and expert modeler, *IJUM Med. J. Malaysia* 19 (2) (2020) 1–8, <https://doi.org/10.31436/ijmjm.v19i2.1606>.
- [33] C. Selinger, M. Choisy, S. Alizon, Predicting COVID-19 incidence in French hospitals using human contact network analytics, *Int. J. Infect. Dis.* 111 (2021) 100–107, <https://doi.org/10.1016/j.ijid.2021.08.029>.
- [34] S. Roy, G.S. Bhunia, P.K. Shit, Spatial prediction of COVID-19 epidemic using ARIMA techniques in India, *Model. Earth Syst. Environ.* 2019 (2020), <https://doi.org/10.1007/s40808-020-00890-y>, 123456789.
- [35] E.N. Aidoo, R.T. Ampofo, G.E. Awashie, S.K. Appiah, A.O. Adebajji, Modelling COVID-19 incidence in the African sub-region using smooth transition autoregressive model, *Model. Earth Syst. Environ.* 8 (1) (2021) 961–966, <https://doi.org/10.1007/s40808-021-01136-1>.
- [36] A.C. Shang, K.E. Galow, G.G. Galow, Regional forecasting of COVID-19 caseload by non-parametric regression: a VAR epidemiological model, *AIMS Public Health* 8 (1) (2021) 124–136, <https://doi.org/10.3934/publichealth.2021010>.
- [37] K.L. Lo, M. Zhang, Y. Chen, J.J. Mi, Forecasting the Trend of COVID-19 Considering the Impacts of Public Health Interventions: an Application of FGM and Buffer Level, vol. 5, Springer International Publishing, 2021, <https://doi.org/10.1007/s41666-021-00103-w>.
- [38] D.M.K.N. Seneviratna, R.M.K.T. Rathnayaka, Hybrid Grey Exponential Smoothing Approach for Predicting Transmission Dynamics of the COVID-19 Outbreak in Sri Lanka, *Grey Syst*, 2022, <https://doi.org/10.1108/GS-06-2021-0085>. Published online.
- [39] A. Saxena, Grey forecasting models based on internal optimization for Novel Corona virus (COVID-19), *Appl. Soft Comput.* 111 (2021), 107735, <https://doi.org/10.1016/j.asoc.2021.107735>.
- [40] S.M. Ayyoubzadeh, S.M. Ayyoubzadeh, H. Zahedi, M. Ahmadi, S.R. Niakan Kalhori, Predicting COVID-19 incidence through analysis of Google trends data in Iran: data mining and deep learning pilot study, *JMIR Public Health Surveill.* 6 (2) (2020) 1–6, <https://doi.org/10.2196/18828>.
- [41] A. Safari, R. Hosseini, M. Mazinani, A novel deep interval type-2 fuzzy LSTM (DIT2FLSTM) model applied to COVID-19 pandemic time-series prediction, *J. Biomed. Inf.* 123 (December 2020) (2021), 103920, <https://doi.org/10.1016/j.jbi.2021.103920>.

- [42] B. Lucas, B. Vahedi, M. Karimzadeh, A spatiotemporal machine learning approach to forecasting COVID-19 incidence at the county level in the USA, *Int. J. Data Sci. Anal.* (2022), <https://doi.org/10.1007/s41060-021-00295-9>. Published online.
- [43] B.S. Price, M. Khodaverdi, A. Halasz, et al., Predicting increases in COVID-19 incidence to identify locations for targeted testing in West Virginia: a machine learning enhanced approach, *PLoS One* 16 (11 November) (2021) 1–16, <https://doi.org/10.1371/journal.pone.0259538>.
- [44] R. Ravinder, S. Singh, S. Bishnoi, et al., An adaptive, interacting, cluster-based model for predicting the transmission dynamics of COVID-19, *Heliyon* 6 (12) (2020), e05722, <https://doi.org/10.1016/j.heliyon.2020.e05722>.
- [45] S. Zhang, J. Ponce, Z. Zhang, G. Lin, G. Karniadakis, An integrated framework for building trustworthy data-driven epidemiological models: application to the COVID-19 outbreak in New York City, *PLoS Comput. Biol.* 17 (9) (2021) 1–29, <https://doi.org/10.1371/journal.pcbi.1009334>.
- [46] J. Xu, Y. Tang, Bayesian framework for multi-wave covid-19 epidemic analysis using empirical vaccination data, *Mathematics* 10 (1) (2022) 1–22, <https://doi.org/10.3390/math10010021>.
- [47] Y.S. Chang, S. Mayer, E.S. Davis, et al., Transmission dynamics of large coronavirus disease outbreak in homeless shelter, Chicago, Illinois, USA, 2020, *Emerg. Infect. Dis.* 28 (1) (2022) 76–84, <https://doi.org/10.3201/eid2801.210780>.
- [48] C. Zeng, J. Zhang, Z. Li, et al., Spatial-temporal relationship between population mobility and COVID-19 outbreaks in South Carolina: time series forecasting analysis, *J. Med. Internet Res.* 23 (4) (2021), <https://doi.org/10.2196/27045>.
- [49] Y. Peng, C. Li, Y. Rong, C.P. Pang, X. Chen, H. Chen, Real-time prediction of the daily incidence of COVID-19 in 215 countries and territories using machine learning: model development and validation, *J. Med. Internet Res.* 23 (6) (2021) 1–12, <https://doi.org/10.2196/24285>.
- [50] A. Mollalo, K.M. Rivera, B. Vahedi, Artificial neural network modeling of novel coronavirus (COVID-19) incidence rates across the continental United States, *Int. J. Environ. Res. Publ. Health* 17 (12) (2020) 1–13, <https://doi.org/10.3390/ijerph17124204>.
- [51] X. Yuan, J. Xu, S. Hussain, H. Wang, N. Gao, L. Zhang, Trends and prediction in daily new cases and deaths of COVID-19 in the United States: an internet search-interest based model, *Explor. Res. Hypothesis Med.* (2020) 1–6, <https://doi.org/10.14218/erhm.2020.00023>.
- [52] S.Y. Sahai, S. Gurukar, W.R. KhudaBukhsh, S. Parthasarathy, G.A. Rempala, A machine learning model for nowcasting epidemic incidence, *Math. Biosci.* 343 (April 2021) (2022), 108677, <https://doi.org/10.1016/j.mbs.2021.108677>.
- [53] S. Noor, W. Akram, T. Ahmed, Q.-A. Qurat-ul-Ain, Predicting COVID-19 incidence using data mining techniques: a case study of Pakistan, *BRAIN Broad Res. Artif. Intell. Neurosci.* 11 (4) (2020) 168–184, <https://doi.org/10.18662/brain/11.4/147>.
- [54] L.S. Wylezinski, C.R. Harris, C.N. Heiser, J.D. Gray, C.F. Spurlock, Influence of social determinants of health and county vaccination rates on machine learning models to predict COVID-19 case growth in Tennessee, *BMJ Health Care Inf.* 28 (1) (2021) 1–3, <https://doi.org/10.1136/bmjhci-2021-100439>.
- [55] F. Ahouz, A. Golabpour, Predicting the incidence of COVID-19 using data mining, *BMC Publ. Health* 21 (1) (2021) 1–12, <https://doi.org/10.1186/s12889-021-11058-3>.
- [56] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, 2014. Published online November 14, <http://arxiv.org/abs/1411.4038>.
- [57] S. Bai, J.Z. Kolter, V. Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, 2018. Published online March 3, <http://arxiv.org/abs/1803.01271>.
- [58] J. Chung, Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling, 2014. April 2015.
- [59] C. Gulcehre, F. Bougares, H. Schwenk, Learning Phrase Representations Using RNN Encoder – Decoder, 2014, <https://doi.org/10.3115/v1/D14-1179>. June).
- [60] Y. Teh, G. Hinton, Rate-coded restricted Boltzmann machines for face recognition, in: T. Leen, T. Dietterich, V. Tresp (Eds.), *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 13*, MIT PRESS, 2001, pp. 908–914.
- [61] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning, *IEEE Trans. Intell. Transport. Syst.* 15 (5) (2014) 2191–2201, <https://doi.org/10.1109/ITTS.2014.2311123>.
- [62] G.E. Hinton, Training products of experts by minimizing contrastive divergence, *Neural Comput.* 14 (8) (2002) 1771–1800, <https://doi.org/10.1162/089976602760128018>.
- [63] A. Mohamed, T.N. Sainath, G. Dahl, B. Ramabhadran, G.E. Hinton, M.A. Picheny, in: *Deep Belief Networks Using Discriminative Features for Phone Recognition*, 2011, <https://doi.org/10.1109/ICASSP.2011.5947494>.
- [64] C. Watkins, Learning from delayed rewards, Published online, [https://www.researchgate.net/publication/33784417\\_Learning\\_From\\_Delayed\\_Rewards](https://www.researchgate.net/publication/33784417_Learning_From_Delayed_Rewards), 1989.
- [65] A. Maoudj, A. Hentout, Optimal path planning approach based on Q-learning algorithm for mobile robots, *Appl. Soft Comput.* 97 (2020), 106796, <https://doi.org/10.1016/j.asoc.2020.106796>.
- [66] H. Liu, C. Yu, C. Yu, C. Chen, H. Wu, A novel axle temperature forecasting method based on decomposition, reinforcement learning optimization and neural network, *Adv. Eng. Inf.* 44 (March) (2020), 101089, <https://doi.org/10.1016/j.aei.2020.101089>.
- [67] Z. Miljković, M. Mitić, M. Lazarević, B. Babić, Neural network Reinforcement Learning for visual control of robot manipulators, *Expert Syst. Appl.* 40 (5) (2013) 1721–1736, <https://doi.org/10.1016/j.eswa.2012.09.010>.
- [68] J.A.M. Sidey-Gibbons, C.J. Sidey-Gibbons, Machine learning in medicine: a practical introduction, *BMC Med. Res. Methodol.* 19 (1) (2019) 1–18, <https://doi.org/10.1186/s12874-019-0681-4>.
- [69] C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.* 20 (1995) 273–297.
- [70] European Centre for Disease Prevention and Control, COVID-19 Coronavirus data - daily (up to 14 December 2020). <https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data-daily-up-to-14-december-2020>. (Accessed 28 February 2020).
- [71] C. Bergmeir, J.M. Benítez, On the use of cross-validation for time series predictor evaluation, *Inf. Sci.* 191 (2012) 192–213, <https://doi.org/10.1016/j.ins.2011.12.028>.
- [72] L.J. Tashman, Out-of-sample tests of forecasting accuracy: an analysis and review, *Int. J. Forecast.* 16 (4) (2000) 437–450, [https://doi.org/10.1016/S0169-2070\(00\)00065-0](https://doi.org/10.1016/S0169-2070(00)00065-0).
- [73] Y. Gautam, Transfer Learning for COVID-19 cases and deaths forecast using LSTM network, *ISA Trans.* (2021), <https://doi.org/10.1016/j.isatra.2020.12.057> xxx.
- [74] K.T. Ly, A COVID-19 forecasting system using adaptive neuro-fuzzy inference, *Finance Res. Lett.* (September) (2020), 101844, <https://doi.org/10.1016/j.frl.2020.101844>.
- [75] N. Science, C. Phenomena, R. Gomes, et al., Forecasting Brazilian and American COVID-19 cases based on artificial intelligence coupled with climatic exogenous variables, *Chaos, Solit. Fractals* 139 (2020), 110027, <https://doi.org/10.1016/j.chaos.2020.110027>.
- [76] B.N. Oreshkin, D. Carпов, N. Chapados, Y. Bengio, N.- Beats, Neural basis expansion analysis for interpretable time series forecasting, in: *ICLR 2020*, 2019, pp. 1–31. <http://arxiv.org/abs/1905.10437>.
- [77] B.B. Hazarika, D. Gupta, Modelling and forecasting of COVID-19 spread using wavelet-coupled random vector functional link networks, *Appl. Soft Comput.* J 96 (2020), 106626, <https://doi.org/10.1016/j.asoc.2020.106626>.
- [78] M. Sulyok, M. Walker, Community movement and covid-19: a global study using google's community mobility reports, *Epidemiol. Infect.* (2020), <https://doi.org/10.1017/S0950268820002757>. Published online.
- [79] Q. Zhang, J. Gao, J.T. Wu, Z. Cao, D. Dajun Zeng, Data science approaches to confronting the COVID-19 pandemic: a narrative review, *Phil. Trans. Roy. Soc. Lond. A Math. Phys. Eng. Sci.* (2214) (2022) 380, <https://doi.org/10.1098/rsta.2021.0127>.
- [80] Y. Wei, F. Sha, Y. Zhao, Q. Jiang, Y. Hao, F. Chen, Better modelling of infectious diseases: lessons from covid-19 in China, *BMJ* (2021) 375, <https://doi.org/10.1136/bmj.n2365>.
- [81] W. Huang, B. Cao, G. Yang, N. Luo, N. Chao, Turn to the internet first? Using online medical behavioral data to forecast COVID-19 epidemic trend, *Inf. Process. Manag.* 58 (3) (2021), 102486, <https://doi.org/10.1016/j.ipm.2020.102486>.
- [82] S. Mohan, A.K. Solanki, H.K. Taluja, Anuradha, A. Singh, Predicting the impact of the third wave of COVID-19 in India using hybrid statistical machine learning models: a time series forecasting and sentiment analysis approach, *Comput. Biol. Med.* 144 (February) (2022), 105354, <https://doi.org/10.1016/j.combiomed.2022.105354>.