



COVID-19 forecasting and intervention planning using gated recurrent unit and evolutionary algorithm

Luning Bi¹ · Mohammad Fili¹ · Guiping Hu²

Received: 15 September 2021 / Accepted: 3 May 2022 / Published online: 31 May 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

The rapid spread of COVID-19, caused by the SARS-CoV-2 virus, has had and continues to pose a significant threat to global health. We propose a predictive model based on the gated recurrent unit (GRU) that investigates the influence of non-pharmaceutical interventions (NPIs) on the progression of COVID-19. The proposed model is validated by case studies for multiple states in the United States. It should be noted that the proposed model can be generalized to other regions of interest. The results show that the predictive model can achieve accurate forecasts across the US. The forecast is then utilized to identify the optimal mitigation policies. The goal is to identify the best stringency level for each policy that can minimize the total number of new COVID-19 cases while minimizing the mitigation costs. A meta-heuristics method, named multi-population evolutionary algorithm with differential evolution (MPEA-DE), has been developed to identify optimal mitigation strategies that minimize COVID-19 infection cases while reducing economic and other negative implications. We compared the optimal mitigation strategies identified by the MPEA-DE model with three baseline search strategies. The results show that MPEA-DE performs better than other baseline models based on prescription dominance.

Keywords COVID-19 prediction · Differential evolution (DE) · Gated recurrent unit (GRU) · Multi-population evolutionary algorithm · Policy prescription

1 Introduction

Since March 2020, COVID-19 has been rapidly spreading worldwide. As of August 17th, 2021, there are over 208 million documented COVID-19 cases and over 4.3 million deaths, constituting major health, economic, and social harms to many countries and regions [1]. To proactively plan the needs for the healthcare and provide a good understanding of policy decisions, the accurate prediction of COVID-19 cases is essential. Many of the models to

predict COVID-19 cases are based on traditional epidemiological methods, such as susceptible-infected-recovered (SIR) [2]. These models consider susceptible individuals, the number of people infected, and the number of people recovered and incorporate parameters, such as the basic reproduction number R_0 , to predict the outbreak of the COVID-19 [3]. However, the susceptibility to infection has been manipulated dramatically because authorities have implemented different non-pharmaceutical interventions (NPIs), such as face coverings and social distancing, to reduce the spread of the virus [4]. The non-linear complexity of these NPIs will reduce the generalization and robustness of the models. Another issue of epidemiological methods is the non-stationarity of the factors [5]. Since these NPIs and their influence change over time, static models cannot achieve high prediction accuracy for long-term prediction.

To overcome the shortcomings of epidemiological methods, machine learning based methods have been developed for outbreak prediction. Yin et al. proposed a stacking model for the prediction of antigenic variants of

✉ Guiping Hu
ghgis@rit.edu

Luning Bi
luningbi@iastate.edu

Mohammad Fili
mfili@iastate.edu

¹ Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011, USA

² Department of Sustainability, Rochester Institute of Technology, Rochester, NY 14623, USA

H1N1 influenza virus, which achieved 80–95% prediction accuracy [6]. Agarwal et al. discovered the correlation between weather parameters and dengue outbreak using the regression model and the k-means clustering algorithm [7]. Liang et al. presented a random forest based algorithm for the prediction of global African swine fever outbreaks and achieved a higher test accuracy than other methods [8]. For the prediction of the COVID-19 spread, time-series prediction using the infection data (e.g., daily new cases, total cases, daily deaths, daily recoveries) has been widely adopted since the infection cases are the result of all unknown epidemiological, cultural, and economic factors [9]. However, a significant amount of information on transmission rates has been included in the NPIs [2]. Thus, it is necessary to consider the time-series infection data and NPIs simultaneously. Due to non-linear interactions among NPIs, linear models are inappropriate and ineffective. Instead, deep learning models can strengthen the generalization ability and flexibility. LSTM model have been used in some COVID-19 studies [10, 11]. One disadvantage of LSTM is that the training speed is much slower due to the increased number of parameters. To improve training speed, the gated recurrent unit (GRU) simplified the model structure by using only two gates, namely, reset gate and update gate [12]. In this paper, GRU has been implemented for the prediction of the COVID-19 spread.

After obtaining the predicted daily cases, the policymakers need to make appropriate intervention plans that optimize COVID-19 mitigation strategies while reducing the economic and social impacts. The two goals are often conflicting. For example, when people are required to quarantine in their homes, the number of new cases can be significantly reduced. However, the economy will also be negatively influenced. Therefore, this problem can be formulated as a bi-objective optimization problem that searches for the Pareto frontier between the two competing objectives.

The major challenge is that the time-series prediction model of COVID-19 is a non-linear model which cannot be solved by existing available solvers. Under this context, evolutionary algorithms which search for improved solutions iteratively become an option. An evolutionary algorithm (EA) is an optimization algorithm that mimics biological mechanisms such as mutation, recombination, and natural selection to find an optimal design within specific constraints [4]. Since evolutionary algorithms don't make any assumption about the objective function, they perform well in searching for approximate optimal solutions to all types of problems [13]. In this paper, the evolutionary algorithm generates groups of prescriptions (NPIs) for each region. Then these prescriptions are evaluated from two perspectives. One is to evaluate the new cases using the prediction model. The other one is to

evaluate the social and economic impacts. We assume that for each region, the impact of different levels of a specific NPI can be represented by a fixed value defined by a cost matrix. According to this cost matrix, the social and economic cost of prescriptions can be measured by addition. To ensure the generalization ability and stability of the algorithm, multi-population evolutionary algorithm with differential evolution (MPEA-DE) is proposed in this paper.

The main contributions of this research can be summarized as follows:

- An approach that connects the predictive model with the prescriptive model has been proposed to provide optimal NPI prescriptions for policymakers based on the historical NPIs and other context information.
- For prediction, a GRU-based hybrid model has been implemented to predict the spread of COVID-19 in the 50 regions of the United States. The time-series infection data and NPIs are considered simultaneously.
- For prescription, based on the prediction of the hybrid model, a multi-population evolutionary algorithm using blind greedy has been proposed to search for optimal intervention plans that can minimize the newly infected cases as well as the social and economic cost. Different initialization strategies are used to generate the population. To improve the exploration ability and exploitation ability of the algorithm, multiple evolutionary algorithms are utilized to evolve the sub-populations together in a cooperative way by using the DE strategy.
- Both the proposed predictive model and the prescriptive model have been compared with other baseline models using different test time windows in multiple geographical areas. The comparison results have proved the effectiveness of the proposed approach.

The rest of this paper is organized as follows: Sect. 2 describes the datasets used in this study as well as the data preprocessing steps. Section 3 introduces the framework of the whole process and the principle of the prediction model and the prescription model. In Sect. 4, the proposed prediction model and prescription model have been tested under various scenarios. The comparison results are also discussed in this section. In Sect. 5, we further discuss the findings and applications of the proposed method. Finally, we summarize the contributions and conclude the future research directions in Sect. 6.

2 Materials and methods

In this section, we describe the datasets used for the predictive and the prescriptive models. We also explain the additional variables as a part of the data preprocessing

procedure. Moreover, we propose a method to generate cost matrix to simulate the cost of policies in the real world.

2.1 Dataset

The OxCGRT data contains 20 indicators of the government responses. Eight measures are corresponding to the closure policies (e.g., school closures). Four of the measure are related to economic policies such as income support. There are also 8 healthcare-related policies (e.g., emergency investment in healthcare) recorded for different countries. OxCGRT updates the dataset continuously, but at the time of this study, we limited the time window from January 2020 to May 2021. In this study, we used closure and healthcare related policies for the purpose of predicting the new cases. In addition to the policies mentioned above, COVID-19 dataset contains information regarding the number of confirmed cases, and confirmed deaths for different states on a daily basis. In the following, the two categories are explained in more detail, and the set of policies in each of them are described.

- Containment and closure policies** These policies are denoted as C1–C8. The policies in this category include school closure, workplace closing, public events cancellation, gathering restrictions, public transport closure, stay-at-home requirement, internal movements restriction (between cities and regions), and international travel control for foreign travelers. All policies are ordinal variables for each the number of levels are summarized in Table 1. The values for any policy start from 0, which means no measures taken or restrictions applied, and goes up to the most stringent level, which includes maximum restriction or closure. Figure 1 shows the mean stringency level for closure policies in the United States. For most policies, we can see that the mean stringency increases and reaches a maximum around April 2020, and then goes down afterward.

Table 1 Closure policies description

Policy	Description	Levels
C1	School closure	4
C2	Workplace closure	4
C3	Public events cancellation	3
C4	Gathering restriction	5
C5	Public transport closure	3
C6	Stay at home requirements	4
C7	Internal movement restrictions	3
C8	International travel controls	5

- Health system policies** This category includes the policies pertinent to the healthcare system. Variables in this category are denoted by H1–H8. Presence of public information campaigns is one of the policies in this category. Testing availability is another recorded policy which shows at a certain time who has access to the testing. There are also other policies in this category such as contact tracing, emergency investment, vaccination investment, facial coverings policies, vaccination policies, and protection of elderly people. Table 2 shows the number of levels for discrete variables each starting from 0 (no action) to the most stringent level. H4 and H5 policies are continuous variables; therefore, they are not in Table 2.

In this study, we only used H1, H2, H3, and H6 for the prediction. Figure 2 shows the mean stringency level for the policies in healthcare category for United States.

In this study, we have also used the US states population dataset (<http://www2.census.gov/programs-surveys/popest/datasets/2010-2019/national/totals/nst-est2019-alldata.csv>). Later, the ratio of confirmed cases to the overall population is calculated as the proportion of people infected, and is added to the dataset as a new feature.

2.2 Data preprocessing

Data preprocessing is one of the critical steps in almost any machine learning project. In this section, we explain the preprocessing steps taken to prepare the data for model training. One of the new variables introduced is “new cases” which is obtained by sequential differencing of the confirmed cases. Eq. (1) shows the new cases:

$$y_a^t = c_a^t - c_a^{t-1} \tag{1}$$

where c_a^t is the number of cumulative number of COVID-19 cases at time t for state a and y_a^t is the number of new cases for state a at time t . For those with missing values, we assumed there was no new cases reported that day; i.e., confirmed cases for that day was the same as previous day. We also replaced any negative values of y_a^t with 0 since it should be a non-negative variable. To smooth the number of new cases, we used rolling mean with weekly windows. Equation (2) shows the formula:

$$\tilde{y}_a^t = \sum_{T=t-6}^t /7, t \geq 7 \tag{2}$$

where \tilde{y}_a^t is the smoothed number of new cases at time point t for state a . We also defined the percent change in the smoothed number of new cases. This variable is denoted as RC_a^t and is defined as in Eq. (3):

Fig. 1 Mean stringency level for healthcare policies in United States

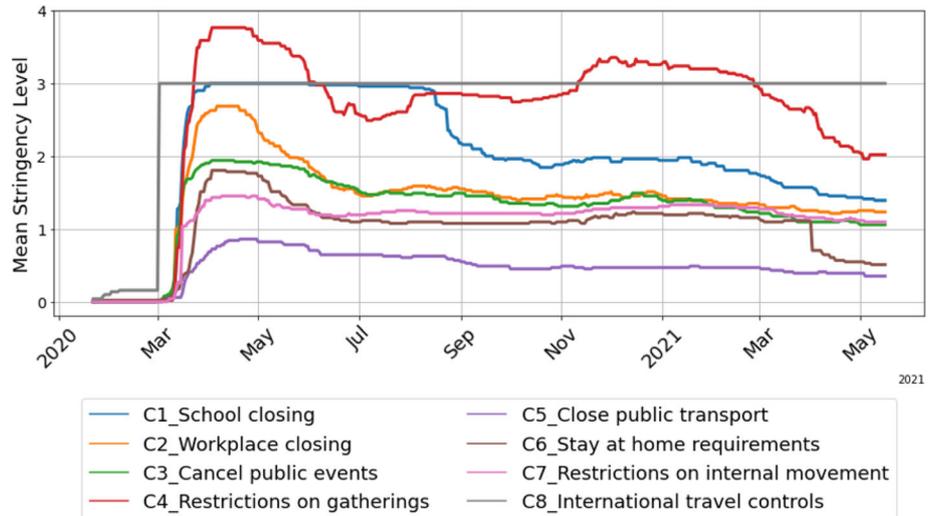


Table 2 Healthcare system policies description

Policy	Description	Levels
H1	Public information campaigns	3
H2	Testing policy	4
H3	Contact tracing	3
H6	Facial coverings	5
H7	Vaccination policy	6
H8	Protection of elderly people	4

$$\hat{z}_a^t = \sum_{T=t-6}^t \hat{z}_a^T / 7, t \geq 7 \tag{5}$$

$$RD_a^t = (\hat{z}_a^t - \hat{z}_a^{t-1}) / \hat{z}_a^{t-1} \tag{6}$$

where z_a^t is the number of new deaths at time t in state a , d_a^t is the number of confirmed deaths at time point t in state a , \hat{z}_a^t is the smoothed number of new deaths at time t and state a , and RD_a^t is the percentage of change in the smoothed number of new deaths at time point t and state a .

$$RC_a^t = \hat{y}_a^t / \hat{y}_a^{t-1} \tag{3}$$

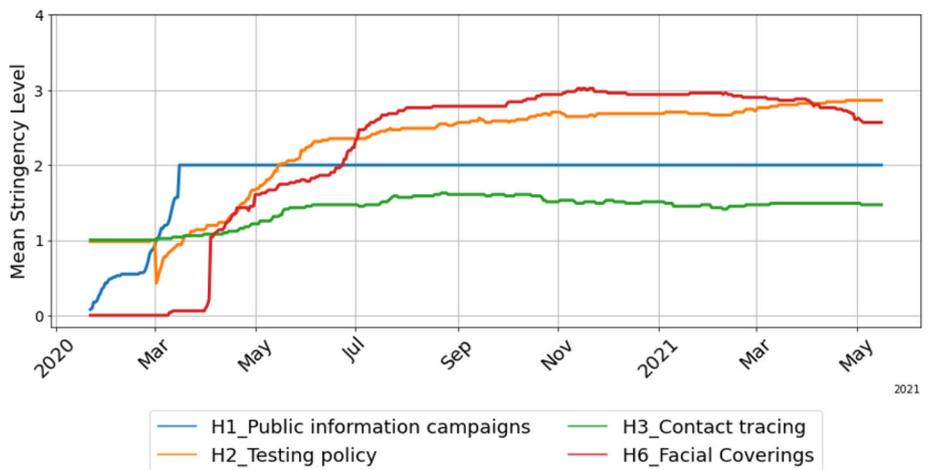
The same procedure is applied for the number of deaths. The equations for these introduced variables are summarized in Eqs. (4)–(6).

$$z_a^t = d_a^t - d_a^{t-1} \tag{4}$$

Figure 3A, B show the total number of smoothed new cases and deaths, respectively. For the smoothed new cases, we can see that the peak occurred in the beginning of 2021, while for the smoothed new deaths we can observe two peaks, one in April 2020, and the other one in the beginning of 2021.

As mentioned earlier, we used population data in the analysis. We defined a new variable as the proportion of

Fig. 2 Mean stringency level for closure policies in United States



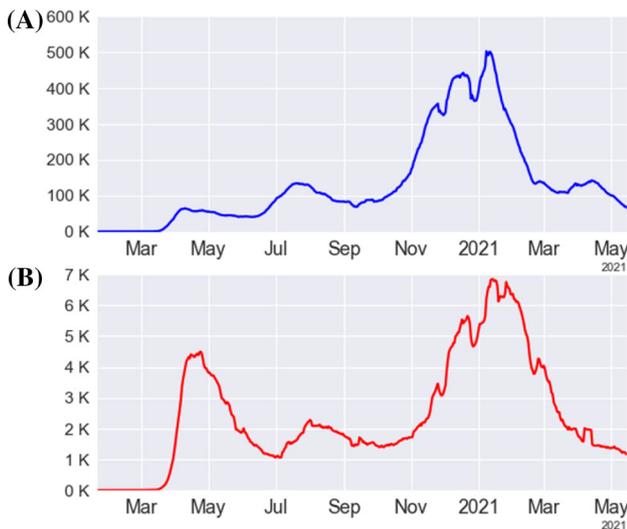


Fig. 3 Total number of smoothed new cases (A), and total number of smoothed new deaths recorded in United States (B)

people infected by Covid-19 in state a at time t (denoted as p_a^t) using population of state a , which is shown in Eq. (7):

$$p_a^t = c_a^t / N_a \tag{7}$$

where N_a is the population of state a . In this study, the response variable is defined as the ratio of the percent of change in the number of smoothed new cases to the proportion of people not being infected at time t and state a (φ_a^t). This new dependent variable, named prediction ratio, is represented in Eq. (8).

$$\varphi_a^t = RC_a^t / (1 - p_a^t) \tag{8}$$

The target of this paper is to model the relationship between NPIs and the prediction ratio and then search for the optimal solution which can minimize infected cases as well as the impacts based on the predictive model.

2.3 Cost matrix

In order to find the best policies for a region, one needs the cost corresponding to each policy. If we do not consider the cost, then the best mitigation strategy is to apply the most stringent level of each policy as this reduces the new cases the most. However, in reality, governments may have serious limitations in applying such strategies due to the infrastructure, budget constraints, or other restrictions.

Estimating the cost of each policy can be really challenging due to the complexity of estimation and the presence of many factors at the same time. Identifying these factors and the magnitude of the effect of each may require a separate in-depth study to quantify the relationships and estimate cost for a policy and a given region. However, in this study, we tried to take a simpler approach for the cost

matrix. It worth mentioning that our study proposes a framework which is able to work with any given cost matrix regardless of the structure or underlying assumptions for policies. Nevertheless, to validate the performance of our model, we created different scenarios for the cost matrix and explored each of them to see how well our model performs in different situations.

We can assume no significant difference between the policy costs, i.e., the cost is uniformly distributed between 0 and 5 for each policy. The variation in cost can be attributed to the differences between regions when implementing a certain policy. Figure 4 shows the average cost for closure and healthcare policies across different states.

2.3.1 Scenario generation procedure

However, assuming that no significant difference between the policy costs may be far from the reality. Therefore, we designed a more sophisticated procedure for the remaining scenarios. We took a more precise approach to model the cost relations. For this purpose, first, we assigned each policy to one of the groups below:

- Group 1: policies with low cost
- Group 2: policies with medium cost
- Group 3: policies with high cost

In the next step, we made pairwise comparisons to specify the significance of one group to the other. In other words, we used pairwise comparisons to suggest the ratio between the average cost levels of policies within a group to that of policies in another group. We used a 1–10 scale for the pairwise comparisons. We assigned the value 5 to group 1 [Eq. (9)] as the baseline. Then, we obtained the average costs for other groups based on Eqs. (10)–(11):

$$\mu_1 = 5 \tag{9}$$

$$\mu_2 = \mu_1 r_{21} \tag{10}$$

$$\mu_3 = \min(\mu_1 r_{31}, \mu_1 r_{32} r_{21}) \tag{11}$$

where r_{ij} is the significance, or the ratio of the average costs of group i to group j ($i > j, i, j = 1, 2, 3$), which is obtained from pairwise comparisons. For the third group, we used the minimum function since the cost for group 3 can be obtained either directly by comparing the third group to the first group, or by comparing group 3 with 2 and then comparing group 2 with 1. In this design, we allowed for discrepancies between the ratios; in other words, we allowed to have $r_{31} \neq r_{32} r_{21}$ which may happen in pairwise comparisons. In the next step, to generate the cost for policies in group i ($i = 1, 2, 3$), we assumed costs are normally distributed with mean μ_i and standard deviation σ_i as shown in Eq. (12).

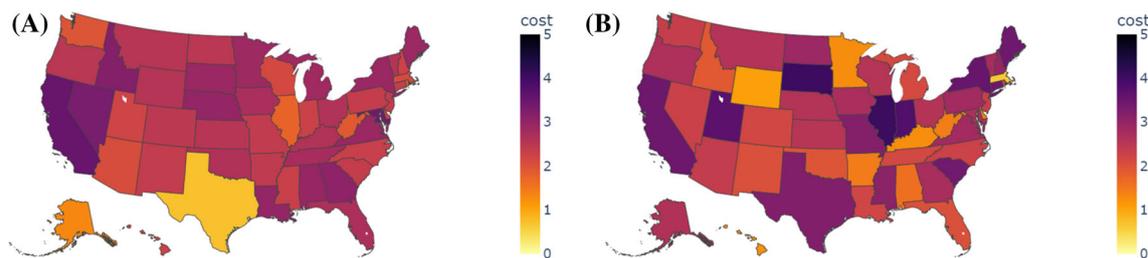


Fig. 4 Average policy cost across different states in US for closure category (A), and healthcare category (B)

$$\theta_{ij} \sim \mathcal{N}(\mu_i, \sigma_i) \quad (12)$$

where θ_{ij} is the cost of policy j in group i . Here, we used $\sigma_i = 2$. We chose this value based on experiments.

Since we have different regions, and each might have different infrastructures, the cost of a policy might be different from one region to another. To take this into account, we assumed the cost of each policy for different regions is uniformly distributed between $\theta_{ij} - \epsilon$ and $\theta_{ij} + \epsilon$ as shown in Eq. (13):

$$\phi_{ijk} \sim \text{Uniform}(\theta_{ij} - \epsilon, \theta_{ij} + \epsilon) \quad (13)$$

where i is the index for groups, j is the policy index within group i , and k shows the region index. Here, we used $\epsilon = 4$, which is selected based on experiments.

As shown earlier, there exists several stringency levels for each policy. Applying a policy at level 1 might not be as costly as applying the most stringent level of the same policy in a region. We used the fourth root of stringency level as the cost multiplier. This is shown in Eq. (14):

$$C_{ijkl} \sim \phi_{ijk} \sqrt[4]{l} \quad (14)$$

where l is the stringency level of policy j within group i for region k .

The aforementioned cost generation procedure is not designed to estimate the true costs and may not be a considered as a tool to illustrate the true relationships between the policies. In fact, the goal is to generate different scenarios for our model to examine its performance in different situations. The assumptions used in this procedure may not hold in the real world, but it provides a systematic approach to validate the model under different circumstances.

2.3.2 Generated scenarios

In the set of scenarios used in this paper, we included four sources of variations: (1) variation between groups, (2) variation between policies within a group, (3) variation between regions for each policy, and (4) variation between the levels of a policy in a region. We used pairwise comparisons to generate variation between groups, normal

distribution for variation between policies within a group, uniform distribution for variation between regions, and non-linear scaling for stringency level variations.

The groupings for scenarios 1–3 are tabulated in Table 3. The numbers in this table represent the groups to which a policy is assigned. The pairwise comparison scores are shown in Table 4. We used a scale of 1–10 for the comparisons.

Scenario 1 is generated according to the policy type, meaning that the logic behind the grouping is that the policies related to the public have more significant impacts than the rest. The policies related to the public lead to direct losses in airfare, lodging, food, and transportation as well as the ancillary impacts to event sponsors, job market and local economy. Thus, we assigned public events cancellation, public transport closure and public information campaigns as policies with high costs. Scenario 2 is based on the GDP influence. It is assumed that the impact of a policy can be quantified as the multiplication of the number of affected people and the potential individual GDP, for workers, or spending power, for consumers. For example, since students have less spending power, the closure of primary/middle schools has less impact than workplace closure. In scenario 3, we considered cost as the implementation cost.

In Fig. 5, the cost breakdown for scenario 1 has been illustrated as an example. In the top level, we have the group base costs, which are obtained using pairwise comparisons. At a level below, we have policy costs within each group that are normally distributed around the base cost of the group. The cost of applying the same policy may differ from region to region; therefore, we assumed the cost of each policy is uniformly distributed for different regions. Finally, we used the fourth root of stringency levels as the multipliers of policy cost for a particular region to differentiate between the regions.

In Fig. 6, we plotted the costs generated from each scenario. On the left side of each plot, the normal distributions from which the policy costs are generated are shown. In this plot, all costs associated with different levels of stringency levels are plotted for each policy. There are

Table 3 Group assignments for scenarios 1–3

Policy	Scenario 1	Scenario 2	Scenario 3
School closure	2	1	2
Workplace closure	2	3	3
Public events cancellation	3	3	1
Restrictions on gatherings	1	3	3
Public transport closure	3	2	2
Stay at home requirements	1	3	3
Restrictions on internal movement	3	3	2
International travel controls	1	2	1
Public information campaigns	3	1	1
Testing policy	1	1	2
Contact tracing	2	2	2
Facial coverings	1	2	1

Table 4 Pairwise comparison scores for scenarios 1–3

Base Cost Ratio	Scenario 1	Scenario 2	Scenario 3
r_{21}	3	3	4
r_{32}	6	3	4
r_{31}	10	7	8

points with zero costs, which are corresponding to the policies for which we do not take any actions (level 0).

3 Proposed model

In this section, we discuss the mathematical model proposed to predict the new daily increased cases and prescribe the optimal NPI policies.

3.1 Overview of the proposed framework

The framework of our work is summarized in Fig. 7. Firstly, the NPI data, the infection data and other information are collected. Through data processing, these data are used as the input of the prediction model. Secondly, based on the prediction results, a prescription model is used to search for the optimal NPI policies. The policies are evaluated in terms of new daily cases and policy cost. Then the policy makers can choose the appropriate ones from the options and implement them in the real world. Finally, the real effectiveness of policies can be observed and be used as indicator to determine how to upgrade the prediction model and the prescription model.

3.2 Prediction model

GRU is a variant of LSTM since both use the gating mechanism in recurrent neural networks [14]. The

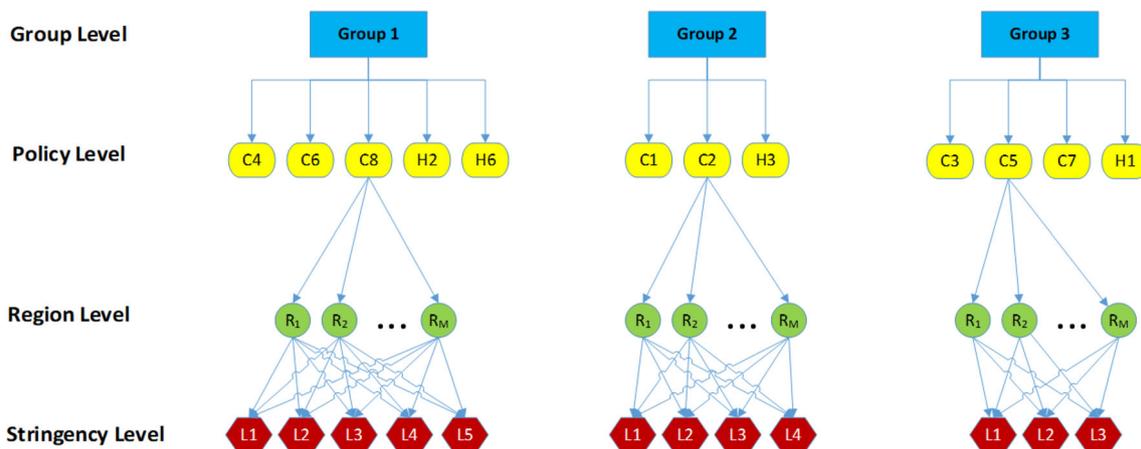


Fig. 5 Cost breakdown for scenario 1

Fig. 6 Cost distribution for different policies in scenarios 1–3

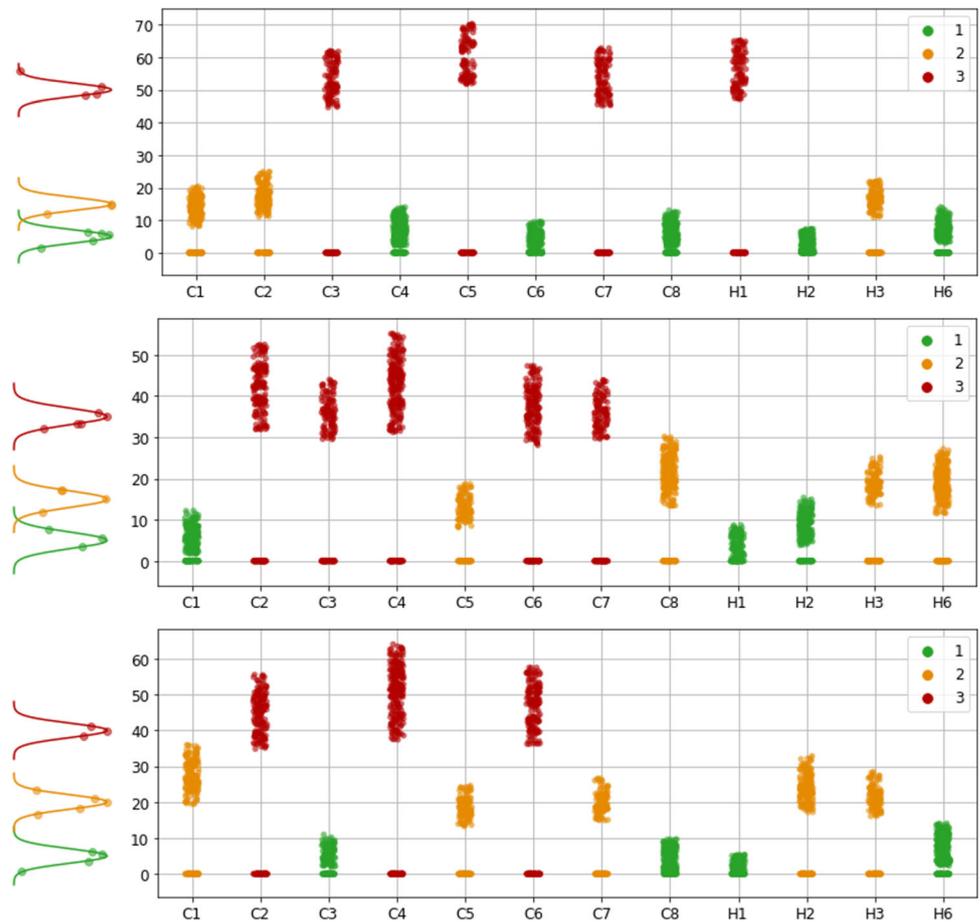
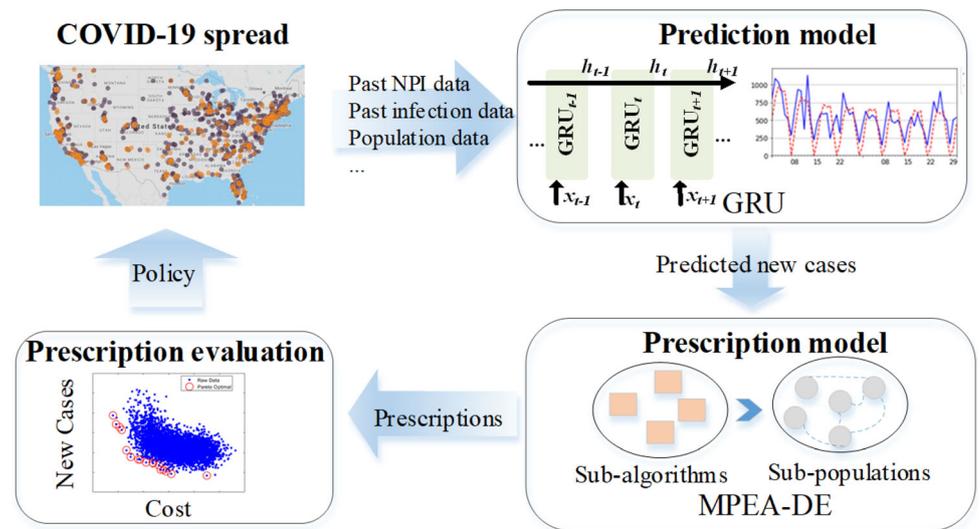


Fig. 7 The flowchart of policy prescription procedure



performance of GRU on tasks, such as speech recognition, was found to be comparable to that of LSTM [12]. GRU also converges faster than LSTM because it has fewer parameters.

A GRU cell consists of two gates: reset gate r and update gate z . A reset gate is used to determine which part of

information should be reset. The value of reset gate at time t , i.e., r_t , is calculated based on the previous output h_{t-1} , and the current input x_t as presented in Eq. (15).

$$r_t = \sigma(W_r[h_{t-1}, x_t]) \tag{15}$$

where σ is a sigmoid function, W_r is the parameter matrix

of the reset gate. The update gate is used to update the output of the GRU, h_t . The value of update gate at time t , i.e., z_t , is computed using previous output h_{t-1} and the current input x_t as presented in Eq. (16).

$$z_t = \sigma(W_z[h_{t-1}, x_t]) \tag{16}$$

where W_z is the parameter matrix of the update gate. Then the candidate hidden layer is calculated according to Eq. (17).

$$h'_t = \tan h(W[r_t h_{t-1}, x_t]) \tag{17}$$

where W is parameter matrix of the candidate hidden layer. Finally, the current output can be obtained according to Eq. (18). The gates, namely, z_t and r_t , and parameters, namely, W_z , W_r and W will be updated in the training process.

$$h_t = (1 - z_t)h_{t-1} + z_t h'_t \tag{18}$$

The GRU-based method proposed in this paper considers three type of input features, i.e., time-series NPI data, time-series prediction ratio and daily changes of NPIs.

How to extract information from the time-series NPI data is the key of building the predictor. A GRU layer is used to convert the time-series NPIs to one output. This serves as the major body of the model. It is assumed that the increase of NPI levels can suppress the spread of the COVID-19. Thus, all parameters of the GRU layer are set to be non-negative to ensure the monotonic influence of NPI levels. However, without the context information (daily new cases in the past), only knowing the influence of NPIs is not enough to make prediction. Thus, the time-series prediction ratio, which considers the infected cases and population of the region, is used as the context input of the model. In this paper, a GRU layer is used to extract the context features (epidemiological, cultural, and economic factors) from time-series prediction ratio. It is assumed that the new infection ratio is proportional to the context input. For example, if the infected cases increase fast in the past several days, it may suggest the prediction ratio in the next days would remain high. Thus, all parameters of the GRU layer are set to be non-negative. Also, it is reasonable to assume that the changes of NPIs can influence the volatility of the number of infected cases. For example, if there are no changes of NPIs in the past days and the levels of NPIs are very low, the daily new cases may increase dramatically in future. If the NPI levels keep dropping in the past several days, it may suggest that the infection is in a downtrend since there is no need for NPIs. Thus, daily changes can be used as an auxiliary input to improve the prediction accuracy. In this paper, the daily changes of NPIs are flattened to one-dimensional input then converted to one node using a fully connected layer. Through

different sub-models, each type of input features will generate one extracted feature. Then the three extracted features are used to obtain the final prediction using a simple formula as shown in Eq. (19).

$$\varphi = f_{context}(1 - f_{NPI}) + f_{NPI\,change} + \xi \tag{19}$$

where φ is the prediction ratio, f_{NPI} is the extracted feature of time-series NPI data, $f_{context}$ is the extracted feature of time-series prediction ratio, $f_{NPI\,change}$ is the extracted feature of NPI daily changes and ξ is the bias.

For multi-step time-series prediction, the rolling prediction method is used to get the future prediction ratios. In the first iteration, given the time-series inputs, the model can output the prediction ratio of the next time point. In the next iterations, the output of the model will be added to the past time-series data and used as the input to forecast the prediction ratio of the next time point. By repeating this process, the model can obtain the prediction results of the coming month.

3.3 Prescription model

The prediction model can provide more guidance and information for policymakers by evaluating the outcome of policies. After predicting the new cases for future, we use the outputs to feed into the prescriptive model so the optimal set of policies for a state can be identified. However, since the search space of NPIs is huge, setting the NPIs manually by experts is still limited. Thus, an automated algorithm that can identify the optimal mitigation strategies with less cost in the large search space is needed. Due to the non-linear complexity of the optimization problem, evolutionary algorithms which can evolve prescriptions through population-based search is more effective than other methods.

The second policy generator works based on an optimization problem. We were interested to know if we wanted to activate N levels of intervention plans for a country from 34 available levels, what would be those policies? In other words, we wanted the best solution for the following problem:

$$\min_{i=1}^{12} C_{i,X_i} \tag{20}$$

$$\min P(X_1, X_2, \dots, X_{12}) \tag{21}$$

Subject to:

$$X_i \in \{0, 1, \dots, d_i\} \tag{22}$$

where X_i is the level of intervention plan i , C_{i,X_i} is the unit cost of implementing X_i , P is the predicted daily new cases using the NPIs, and d_i is the maximum level for that intervention plan.

On the basis of the traditional population-based algorithm, this paper uses MPEA in which multiple algorithms are added into the framework to ensure the adaptability of algorithm. Aimed to improve the exploitation ability of MPEA, the mechanism of DE is introduced to strengthen the communication between different populations in this paper. It is assumed that there are m algorithms available for the proposed model. The components of MPEA-DE are as follows.

Algorithm 1 MPEA-DE

```

1: Generate  $m$  populations, pop 1, pop 2, . . . , pop  $m$ 
2: while stopping conditions are not satisfied do
3:   for  $i = 1 : m$  do
4:     Evolve pop  $i$  using algorithm  $A_i$ 
5:   end for
6:   if a migration interval is reached then
7:     for  $i = 1 : m$  do
8:       Using DE to update the worst individuals
9:     end for
10:  end if
11: end while

```

- *Initialization* Generate m populations, one for each algorithm.
- *Encoding and evaluation* The individuals, i.e., NPIs prescriptions, are encoded using real integer value encoding method. To improve the consistency of the policy, we assumed that NPIs can only change every 10 days. Since the number of NPIs considered is 12, the total number of decision variables can be calculated by dividing the number of days to prescribe by 10 and then multiplying by 12. For example, if the number of days to prescribe is 60, the number of decision variables is 72. In each iteration, the two objectives of each individual, i.e., new cases and cost, will be evaluated. The individual is used as the input of the predictor to get the predicted new cases. The social and economic cost is used based on the cost matrix.
- *Evolution* Based on the objective values of individuals, the corresponding algorithm of each population will determine the evolution direction. The old individuals will be replaced by the new generated individuals that have dominated objective values.
- *Migration* After several iterations, the diversity of individuals in each population will be reduced dramatically. To improve the evolution efficiency of each population, migration mechanism is introduced. When the number of iteration reaches the predefined number, the worst individuals in the current population are

evolved towards the best ones in the other populations, using the DE scheme as expressed in Eq. (23).

$$I_{worst}^{new} = I_{worst}^{old} + F(I_{best} - I_{worst}^{old}) \quad (23)$$

where I_{worst}^{old} is the worst individual in the current population, I_{best} is the best individual in other populations, F is the scaling factor in DE.

The pseudo-code for the MPEA-DE is as follows.

Through the result analysis of the predictive models, it can be found out that the time-series predictive models can get better accuracy than the non-time series predictive model. This means the past NPI measurements will influence the daily new cases in the present. Therefore, the long-term influence of prescription should be considered in the decision-making. For example, the prescriptions for the next 10 days should also consider the possible new cases in the next one month or two months. However, the performance of predictive models decays as the length of prediction period increases. To address the above concerns, the objective is defined in Eq. (24):

$$D = \sum_{k=0}^T \gamma^k d_k \quad (24)$$

where D is weighted daily new cases, k is the number of prediction period, T is the total number of prediction periods considered, γ is the coefficient to measure the longer-term influence of NPIs and the uncertainty of the predictive model, d_k is the predicted daily new cases in the k th prediction period.

4 Numerical results

In this section, both the prediction model and the prescription model will be tested under different scenarios to show the validity of the methodology.

4.1 Performance of the prediction model

The prediction accuracy of the prediction model determines whether the generated prescriptions are consistent with the real world. Two indices including mean absolute error (MAE) and mean absolute percentage error (MAPE), as calculated in Eqs. (25) and (26) were measured to evaluate the performance of models.

$$MAE = \frac{1}{n} \sum_n |e_i| \tag{25}$$

$$MAPE = \frac{1}{n} \frac{\sum_n |e_i|}{\sum_n d_i} \tag{26}$$

where n is the total number of observations, e_i is the error between the real daily cases and predicted daily cases of i th observation, and d_i is the real daily new cases of i th instance. MAE_{1M} represent the mean absolute error per 1 million people.

4.1.1 Baseline models

Four baseline methods were investigated in this paper.

- (1) Non-time series least absolute shrinkage and selection operator (LASSO). LASSO is a kind of linear regression models that use L1 norm to perform both variable selection and regularization. It is widely used in different research areas owing to its simplicity and interpretability. The goal of the model is to minimize:

$$\sum_{i=1}^n (y_i - \sum_j x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j| \tag{27}$$

where n represents the total number of samples, y_i is the ground truth of the i th sample, x_{ij} is the value of the j th variable of i th sample, β_j is the coefficient of the j th variable, λ is the penalty coefficient, and p is the total number of input variables. In this paper, LASSO takes the NPIs and the prediction ratio of the previous day as the input.

- (2) Time-series LASSO. LASSO can also be used for time-series prediction. The NPIs and the prediction ratio of multiple days can be flattened as one-row inputs. For example, assuming that there are p variables in total, the LASSO model that considers the past t days will have $p \times t$ inputs coefficients.
- (3) Fully connected neural network (FCNN). FCNN is a class of methods that use multiple layers to extract information from the input data [15]. The basic layers are a fully connected layer and an activation layer. The fully connected layer consists of multiple neurons. Each neuron in a fully connected layer

connects to all neurons in the next layer. The output of a fully connected layer is calculated as Eq. (28).

$$y = Wx + b \tag{28}$$

where W is the weight vector, and b is the bias for the node in the next layer. The fully connected layer can only deal with a linear problem. To add the non-linear characteristic to the model, the concept of activation layers was introduced. Some widely used activation functions include sigmoid function, hyperbolic tangent function (Tanh) and Rectified Linear Unit (ReLU) function. In this paper, sigmoid function is used as Eq. (29).

$$S(x) = 1/(1 + e^{-x}) \tag{29}$$

The structure of FCNN is the same as that of the GRU-based method. The three types of input features are converted to one node by a fully connected layer, separately. Then the final prediction is made according to Eq. (29).

- (4) Convolutional neural network. CNN is a class of deep, feed-forward artificial neural networks. It was adopted widely for its fast deployment and high performance on image classification tasks. However, it is also a popular architecture for time series prediction since time-series data can also be viewed as 2-dimensional (2D) data. CNNs are usually composed of convolutional layers, pooling layers, batch normalization layers and fully connected layers.
 - *Convolutional layer* The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters, which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2D activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.
 - *Maxpooling layer* The Maxpooling Layer is applied to perform downsampling operations, i.e. shrinking the feature maps along both width and height by a factor of two. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer.
 - *Full connected layer* Fully connected layers connect every neuron in one layer to every neuron in another

Table 5 Setting of the five experiments

No.	I	II	III	IV	V
Rolling prediction length	6 days	12 days	18 days	24 days	30 days
Train start date	April 1st, 2020				
Train end date (excluded)	The first day of each month (from August 2020 to March 2021)				
Test start date (included)	The first day of each month (from August 2020 to March 2021)				
Test end date	Test start date+Rolling prediction length				

In each experiment, the models were tested using 8 time windows

Table 6 Comparisons among GRU and baseline models

Nos.	Methods	MAE_{1M}	MAPE (%)
I	GRU	101.72	29.62
	NT-LASSO	125.20	25.92
	T-LASSO	124.39	25.71
	FCNN	117.37	23.40
	CNN	119.98	24.38
II	GRU	117.63	23.04
	NT-LASSO	152.26	31.83
	T-LASSO	151.07	31.45
	FCNN	143.71	30.19
III	GRU	127.98	26.18
	NT-LASSO	167.26	36.25
	T-LASSO	166.09	35.84
	FCNN	158.94	34.66
	CNN	159.94	32.80
IV	GRU	135.98	29.43
	NT-LASSO	176.66	30.07
	T-LASSO	175.55	39.70
	FCNN	168.89	38.64
	CNN	169.53	36.62
V	GRU	143.34	32.72
	NT-LASSO	183.22	43.32
	T-LASSO	182.06	42.85
	FCNN	176.07	39.95
	CNN	176.65	41.97

NT-LASSO non-time series LASSO, T-LASSO time-series LASSO

layer. In principle, it is the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to one node.

In this paper, the CNN-based method uses three small CNN modules to convert the three types of input (i.e., time-series NPI data, time-series prediction ratio and daily changes of NPIs) to final output. Since it is assumed the influence of the NPI levels on the prediction ratio is monotonic, NPI levels are used as real-value input of the CNN-based

method. The parameters of the first two CNN modules are set to be non-negative.

4.1.2 Model parameters and experiment design

For the GRU-based method, the numbers of units of the two GRU sub-modules are set at 32. For the LASSO methods, the L1 norm coefficient is set to 0.9. The number of neurons of the FCNN sub-modules is set at 32. For the FCNN-based method, the numbers of neurons of the three sub-modules is set at 32. For the CNN-based method, three convolutional layers with 32 channels and 3*3 kernel size are used. Each convolutional layer is attached with a Maxpooling layer.

The length of the time-series is set to 21, meaning only the policies and cases of the past 21 days are considered to make the prediction. The batch size is set to 1000. The optimizer is Adam [16].

To test the prediction performance of the models, five experiments of different rolling prediction length were designed. As shown in Table 5, in each experiment, the four models were tested using 8 different time windows, i.e. August, September, October, November and December in 2020, and January, February and March in 2021. The final measurement values for each experiment are the average results of the 8 test windows and 50 regions.

4.1.3 Results and comparisons

The results are shown in Table 6. The indexes values were the average values across 50 regions. In the five experiments, the MAE_{1M} and $MAPE$ of the proposed GRU-based method ranged from 101.72–143.34 and 19.62–32.72%, separately. And GRU performed much better than others. In Experiment I, for 6-day prediction, the MAE_{1M} of GRU is 13–19% less than the other methods. The $MAPE$ of GRU is 4–6% than that of others. In Experiment V, for 30-day prediction, the MAE_{1M} and $MAPE$ of GRU were 19–22 and 7–11% less than that of other methods, separately. As the length of the prediction widow increased from 6 days to 30 days, the MAE_{1M} and $MAPE$ of all models increased. The reason is that the prediction error was accumulated when only using the predicted values as the input.

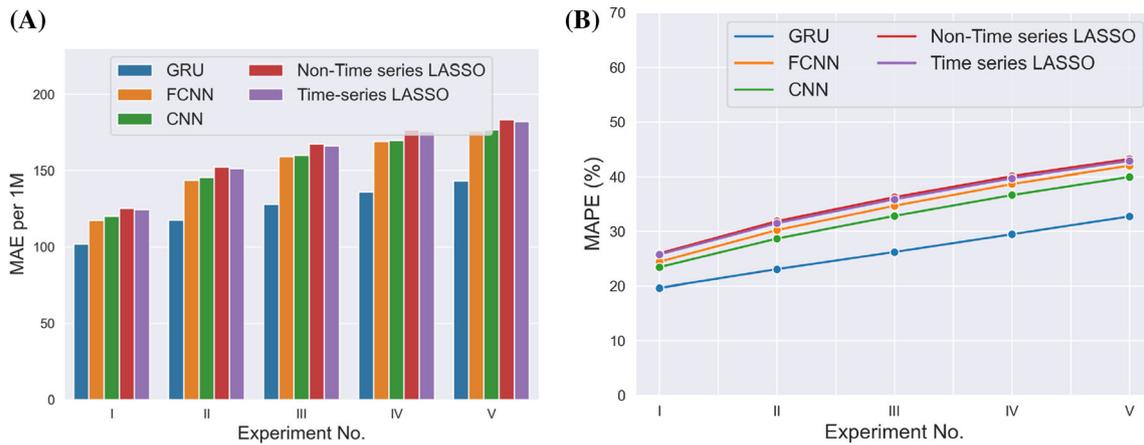


Fig. 8 Comparisons among the four predictive models for different prediction periods **A** MAE_{1M} , **B** $MAPE$

Table 7 Weights of each policy in the non-time series LASSO trained using data from April 1st, 2020 to March 1st, 2021

Policy	Description	Weight
C1	School closure	32.73
C2	Workplace closure	26.14
C3	Public events cancellation	33.12
C4	Gathering restriction	20.37
C5	Public transport closure	1.30
C6	Stay at home requirements	52.31
C7	Internal movement restrictions	1.29
C8	International travel controls	3.03
H1	Public information campaigns	44.41
H2	Testing policy	23.58
H3	Contact tracing	1.98
H6	Facial coverings	31.87

As shown in Fig. 8, compared with the time-series models, non-time series LASSO had greater MAE_{1M} and $MAPE$. This proved the necessity of time-series prediction. It can be also noticed that the performance of time-series LASSO was worse than that of FCNN, CNN and GRU. This reason is that there are some complex interactions between the input variables. Thus, it is necessary to construct high-level features using models that are more complex than linear regression. However, LASSO has better interpretability than neural network-based methods. It can be used for some basic analysis. Table 7 listed the weights of each policy in the non-time series LASSO. Since all policies are at the same scale, the weight value can reflect the influence of each policy. The importance of each feature can be calculated by multiplying the weight by the variance of the feature. Therefore, based on the weights of policies, the top-4 polices are stay at home requirements,

public information campaigns, public events cancellation, and school closure. It can be seen from Fig. 2 that the policy stringency of public information campaigns did not change since March 2020. Thus, the most important polices indicated by non-time series LASSO are stay at home requirements, public events cancellation, and school closure.

In Fig. 9, six regions, i.e., Iowa, West Virginia, Pennsylvania, Massachusetts, South Carolina, and Utah, were selected to show the difference between test ground truth and test predicted cases in 8 test time windows of Experiment V. In general, the predictive model is following the true daily new cases closely and has captured the pattern. However, in some situations, the gap between ground truth and predicted daily cases was large. For example, in the test time window of November 2020, the predictive model thought the daily new cases of Iowa would go down, but on the contrary, the daily new cases increased dramatically in the real world. For other five regions, although the predictive model captured the rising trend, the predicted numbers were much smaller than the ground truth numbers. This was caused by the cumulative error of rolling out prediction.

4.2 Performance of the prescription model

Prescription is the key to identifying the optimal solution and suppressing the spread of COVID-19 using the minimum cost. As discussed in Eq. (20)–(22), two objectives, i.e., cost and average daily new cases, are used to evaluate the quality of generated prescriptions.

4.2.1 Baseline models

Three strategies are used as baselines:

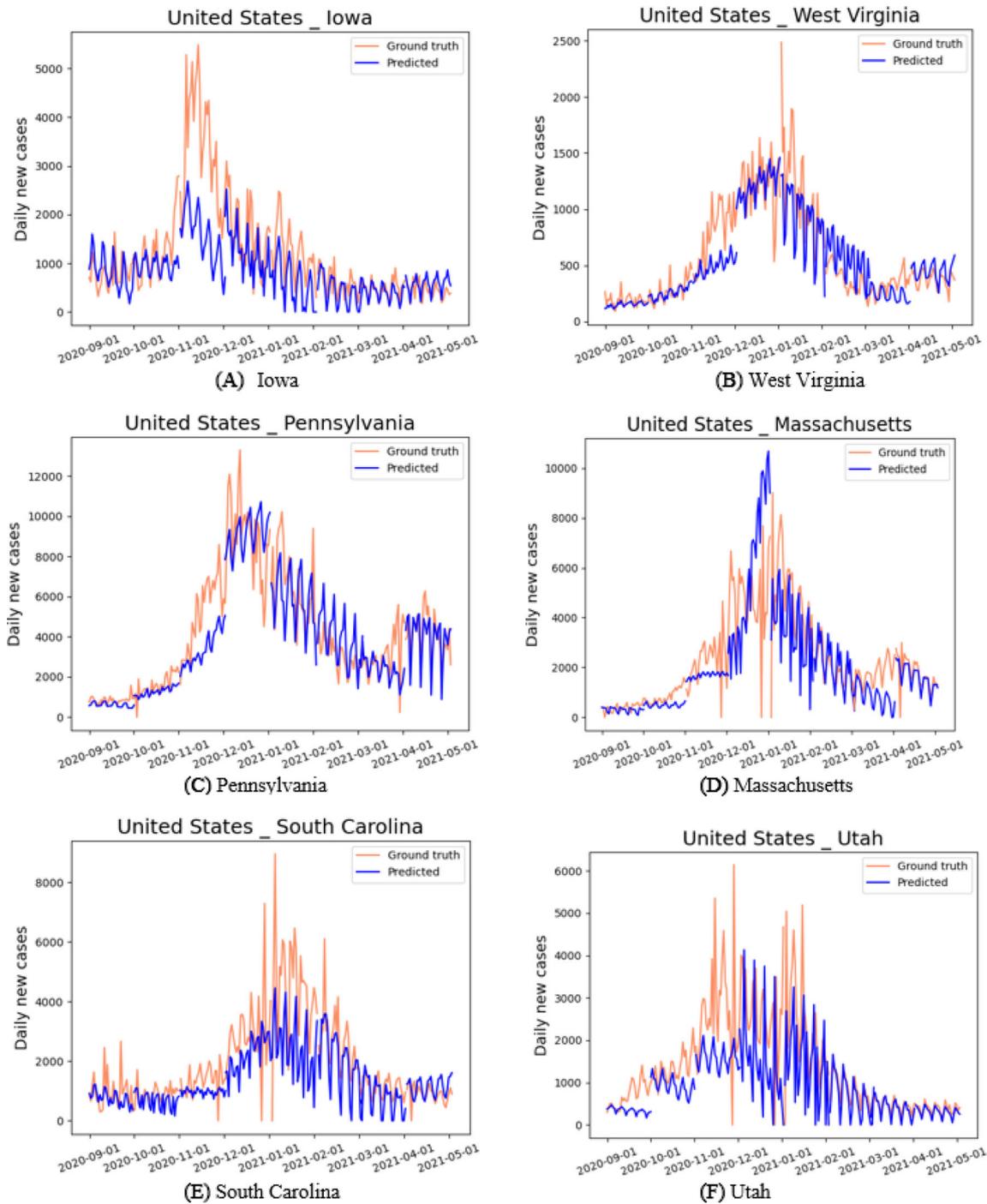


Fig. 9 The ground truth and predicted daily new cases of 8 testing time windows using the GRU-based method in Experiment V

- *Random* Random strategy is to generate solutions by randomly selecting a level for each intervention indicator. In each iteration, new individuals are generated by perturbing the old ones and replacing the individuals of bad quality.
- *Blind greedy* This approach works by adding the maximum level of each policy based on the order of cost. It is a logical assumption that adding more

restrictive policies will lower the number of predicted cases. However, these policies are more expensive to implement. The blind greedy search strategy starts with all NPIs as zeros and then iteratively set the NPI that has the least cost to the maximal level.

- *Blind greedy with random search* In blind greedy with random search, blind greedy is used to generate initial

Table 8 Percentage of states claimed by MPEA-DE for scenario 1

Test start points	Prescription period	Considered period	Claim Percentage
September 1st, 2020	10 days	30 days	94.0
November 1st, 2020	10 days	30 days	92.0
January 1st, 2021	10 days	30 days	94.0

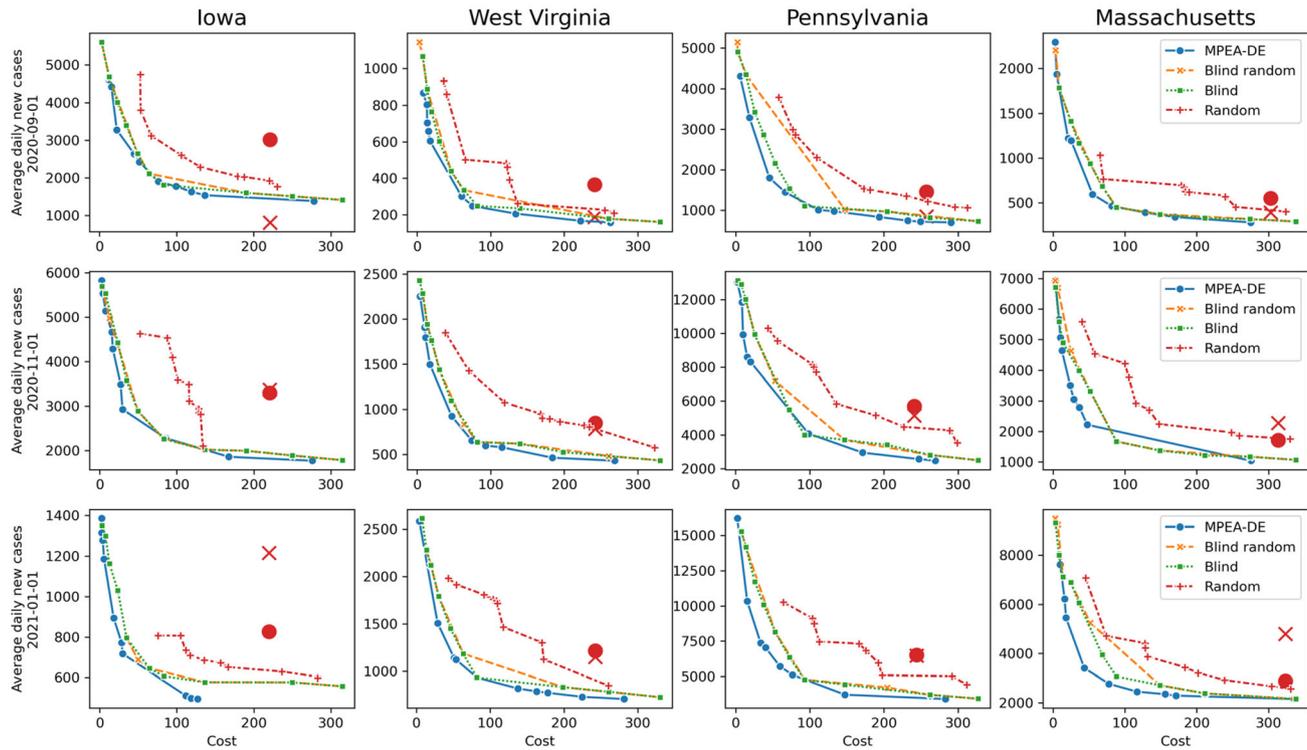


Fig. 10 Pareto frontier of prescriptive models for Iowa, West Virginia, Pennsylvania, and Massachusetts, for three test points under scenario 1. (x-axis represents the cost, y-axis shows the average

number of new cases, a red X indicates the objectives of the prescription in real world and a red dot the predicted objectives of the prescription in real world) (color figure online)

Table 9 Percentage of states claimed by MPEA-DE for scenario 2

Test start points	Prescription period	Considered period	Claim Percentage
September 1st, 2020	10 days	30 days	94.0
November 1st, 2020	10 days	30 days	98.0
January 1st, 2021	10 days	30 days	90.0

solutions in the first iteration and then the random search is performed for the remaining iterations.

4.2.2 Simulation

In this study, we used three baseline/benchmark models, i.e., random search, blind greedy and blind greedy with random search. In the random search strategy, a population consisting of 30 individuals are generated randomly. In each iteration, new individuals are generated by

perturbating the old ones and replacing the individuals of bad quality. The number of iterations is set to 30. The blind greedy search strategy starts with all NPIs as zeros and then iteratively set the NPI that has the least cost to the maximal level. The blind greedy with random search is to use blind greedy search first to generate an initial population consisting of 30 individuals. Then the random search strategy is used to update the population for 30 iterations.

For MPEA-DE, two populations have been generated ($m = 2$). Each population consists of 30 individuals. The first 60% of the individuals in each population is generated

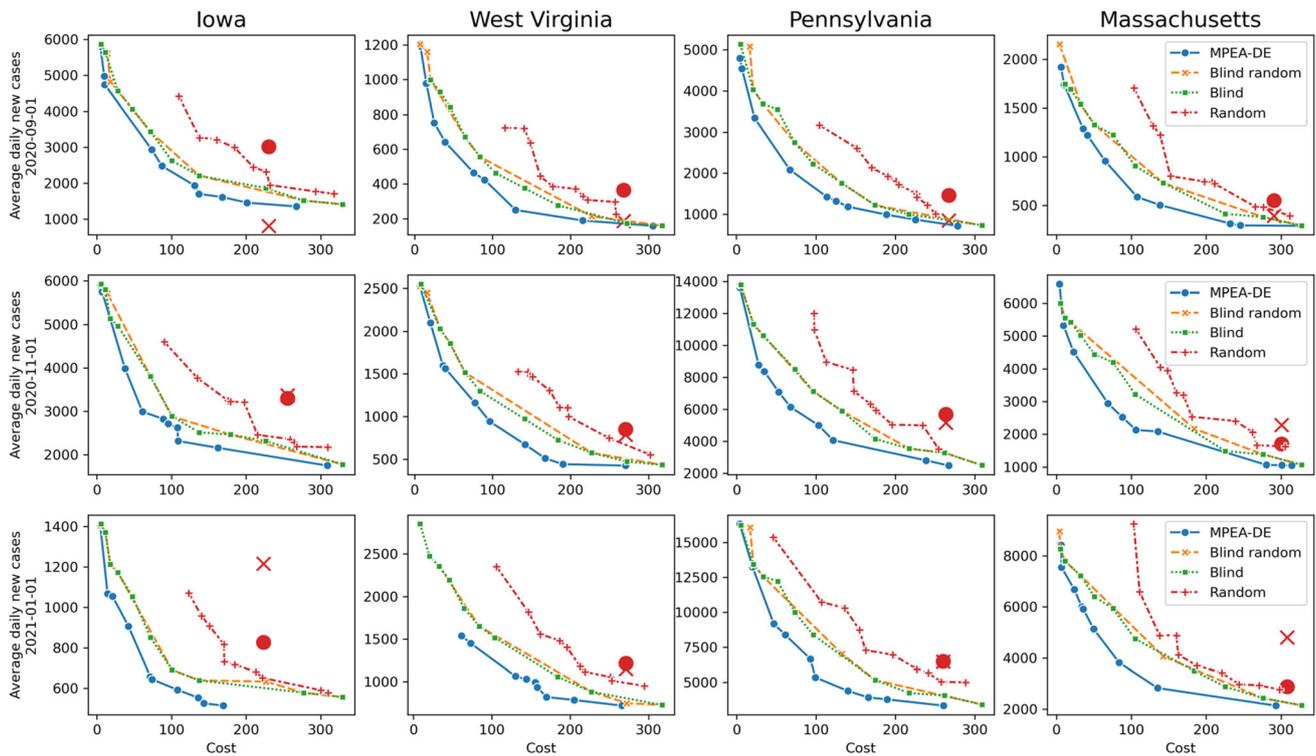


Fig. 11 Pareto frontier of prescriptive models for Iowa, West Virginia, Pennsylvania, and Massachusetts, for three test points under scenario 2. (*x*-axis represents the cost, *y*-axis shows the average

number of new cases, a red X indicates the objectives of the prescription in real world and a red dot the predicted objectives of the prescription in real world) (color figure online)

Table 10 Percentage of states claimed by MPEA-DE for scenario 3

Test start points	Prescription period	Considered period	Claim percentage
September 1st, 2020	10 days	30 days	96.0
November 1st, 2020	10 days	30 days	94.0
January 1st, 2021	10 days	30 days	96.0

by the blind greedy search strategy. To increase the diversity of the population, the remaining 40% of the individuals is generated by the random search strategy. The two populations are evolved using standard genetic algorithm (GA) [17] and DE [18], respectively. The total number of iterations is 30. Every 5 iterations, the DE scheme will be performed to strengthen the communication between two populations.

Each model generated 10 prescriptions/mitigation strategies for each region. The score of a prescription is calculated as the number of prescriptions it dominates (i.e., have better new cases and cost). For example, if a prescription of model A dominates 5 prescriptions of model B and 2 prescriptions of model C, the score of this prescription is 7. Thus, the performance of a prescriptive model can be calculated as the sum of the scores of its prescriptions. For each region, the prescriptive model that has the highest cumulative score is selected as the winner of the region. To measure the performance of prescriptive

models across all regions, the percentage of regions claimed by each model is calculated.

In this study, there are three test points: September 1st, 2020, November 1st, 2020, and, January 1st, 2021. To impose policy stability and reduce the searching space, the NPIs can only change every ten days. The models are used to prescribe for the next 10 days. As discussed in Eq. (24), the long-term influence should be considered in decision making. Thus, the prescription calculated the weighted daily new cases over the next 30 days. The new cases and stringency level averaged a day were used as the two objectives.

4.2.3 Result analysis

Scenario 1: policy type. In the first scenario, the logic behind the grouping is that the policies related to the public have more significant impacts than the rest. The policies related to the public lead to direct losses in airfare, lodging,

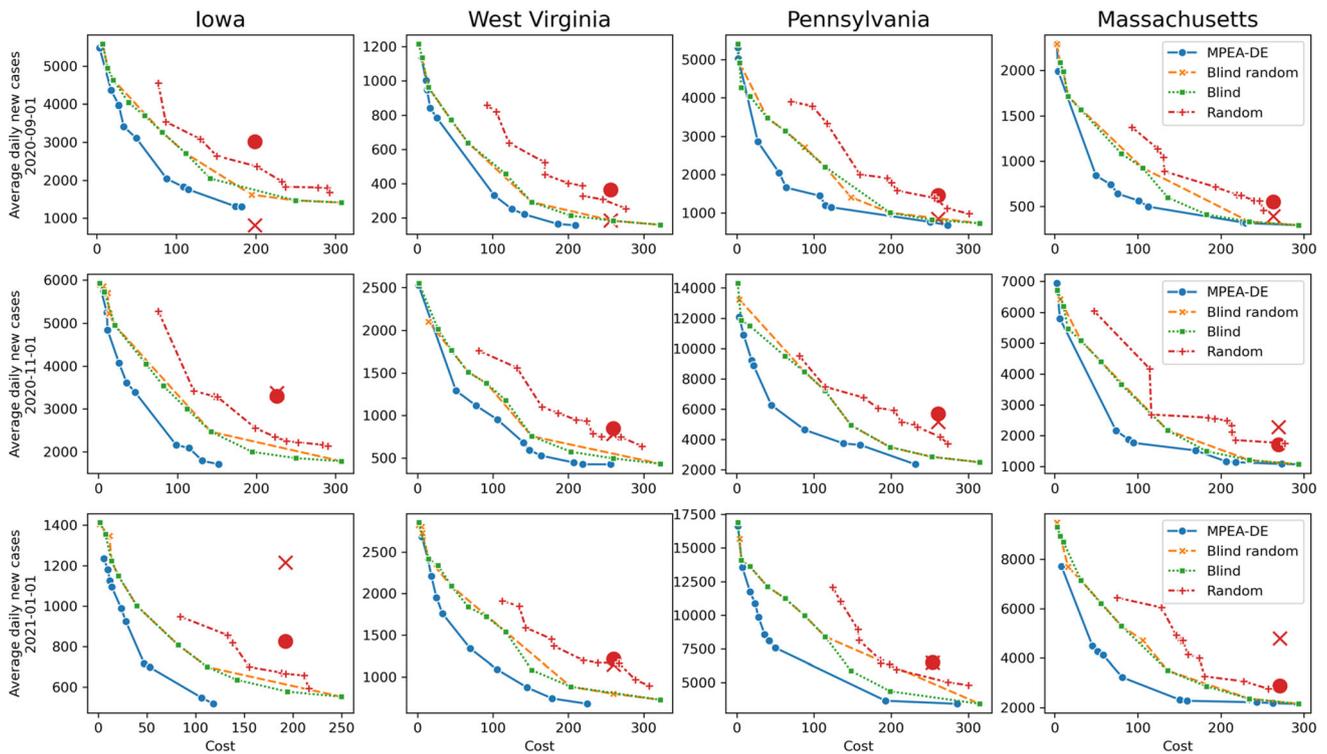


Fig. 12 Pareto frontier of prescriptive models for Iowa, West Virginia, Pennsylvania, and Massachusetts, for three test points under scenario 3. (*x*-axis represents the cost, *y*-axis shows the average

number of new cases, a red X indicates the objectives of the prescription in real world and a red dot the predicted objectives of the prescription in real world) (color figure online)

food, and transportation as well as the ancillary impacts to event sponsors, job market and local economy. Thus, we assigned public events cancellation, public transport closure and public information campaigns as policies with high costs. In Table 8, the percentage of states claimed by the prescriptive model (MPEA-DE) is shown. As we can see the proposed method is completely superior to the other three baseline models.

In Fig. 10, the Pareto frontier generated by each algorithm has been plotted for four states. On the *x*-axis, we have the average cost across the whole test window, and on the *y*-axis, we have the average of new COVID-19 cases. Each point represents a policy suggested by one of the algorithms. The goal is to design policies with both small cost and small number of new cases. As we can see the proposed method is outperforming the baseline models since, on average, for the majority of the policies, MPEA-DE dominated the other two models in both objectives.

The objectives of real-world NPIs have also been evaluated. The predicted objectives and ground-truth objectives were compared to show the reliability of the proposed prescriptions. For most regions and test time windows, the red dot was close to the red X, meaning that the predictive model can provide accurate prediction for the evaluations of prescriptions. Under this context, it can be observed that when compared with the real-world NPIs, the proposed

prescriptions can reduce 50–70% of the cost while having the same or less daily new cases. The proposed prescriptions can also reduce 5–50% of the daily new cases at the same cost of the real-world NPIs.

Scenario 2: GDP influence. In the second scenario, as assumed that the impact of a policy can be quantified as the multiplication of the number of affected people and the potential individual GDP, for workers, or spending power, for consumers. For example, since students have less spending power, the closure of primary/middle schools has less impact than workplace closure.

In Table 9, the percentage of states claimed by the prescriptive model (MPEA-DE) is shown. As we can see the proposed method is completely superior to the other three baseline models. When compared with the real-world NPIs (as shown in Fig. 11), the proposed prescriptions can reduce 38–72% of the cost while having the same or less daily new cases. The proposed prescriptions can also reduce 34–63% of the daily new cases at the same cost of the real-world NPIs.

Scenario 3: implementation cost. In the third scenario, we considered cost as the implementation cost. In Table 10, the percentage of states claimed by the prescriptive model (MPEA-DE) is shown. As we can see, the proposed method is completely superior to the other three baseline models. When compared with the real-world NPIs

(as shown in Fig. 12, the proposed prescriptions can reduce 53–68% of the cost while having the same or less daily new cases. The proposed prescriptions can also reduce 11–62% of the daily new cases at the same cost of the real-world NPIs.

5 Discussion

The economic and social disruptions caused by COVID-19 have been significant. This paper presents a framework to identify optimal non-pharmaceutical intervention solutions based on prediction of future development of the virus. The goal is to provide more scientific and time-effective intervention policies for the decision makers.

In the prediction component, instead of using an existing neural network to predict daily increase numbers, we design an explainable formula [i.e., Eq. (19)] that considers increasing ratio, population, NPIs influence and volatility as the skeleton of the prediction model. Then for each component of Eq. (19), GRU modules are employed to transform the time-series input into the components of the formula. In the numerical experiment/case study, the proposed model was tested on the 50 regions of the United States. Compared to the non-time series LASSO, time-series LASSO, FCNN and CNN, the proposed model can get the highest prediction accuracy. It has also been verified that the increase of NPI stringency levels can effectively reduce the progression of COVID-19, especially for home requirements, public information campaigns, public events cancellation, and school closure.

In the prescription component, a multi-population evolutionary algorithm has been proposed to search for the optimal prescriptions that can minimize the comprehensive cost and suppress the spread of COVID-19. To simulate the real world, the concept of cost matrix is presented to generate reasonable scenarios. Compared to random search and blind greedy search, the proposed algorithm is more efficient in searching the pareto frontier due to the intelligent local evolution strategies and migration mechanism. The proposed algorithm can dominate the solutions of others on over 94% of the 50 regions. The gap between the objectives of pareto frontier and the real-world NPIs have emphasized the importance of NPI policy optimization. Besides, the high prediction accuracy of the proposed prediction model can guarantee the effectiveness of the generated optimal NPI policies.

Although most of states are easing some of its COVID-19 restrictions due to the virus becomes less deadly, the proposed framework can still provide valuable instructions for the prediction and prevention of the future pandemics.

6 Conclusion

Accurate forecasting of infected cases and the right mitigation strategies are key to reducing the spread of COVID-19. In this study, we proposed a framework to identify sets of superior policies from which a decision-maker can choose according to the goals and budgets.

In this paper, a GRU-based model is proposed to predict the spread of COVID-19 using time-series infection data and NPIs. The results have shown that the predictive model can predict the spread of COVID-19 accurately. The prediction results are then employed to identify which policies can be applied to reduce the number of new cases while minimizing the overall costs. To search for the optimal intervention policy, a multi-population evolutionary algorithm named MPEA-DE is proposed. We compared the prescriptive model to three baseline models: random selection strategy and blind greedy search method. The performance of the proposed prescriptive models is evaluated based on the dominance of the generated prescriptions over the other two models. The experiments have shown that in terms of prescription quality, MPEA-DE, which claimed at least 95% of the regions, performs better than other methods. Based on our approach, the authorities can have a better recognition of the outcome of their policies and make policy shifts in time.

This study is subject to a few limitations, which suggest a few research directions. First, in the forecasting model, the vaccination rate is not considered. Since the vaccination period is relatively short at the time of the study, it is better to consider the vaccination factor after collecting sufficient data. Second, in the prescription phase, we used random costs for the policies generated from a distribution. However, the social and economic impacts of intervention plans in the real world are more complicated than what has been assumed in this study. For this purpose, we consider testing a variety of more complicated cost matrices as a part of the robustness check and use models based on real data to predict the social and economic impacts of NPIs for future studies.

Acknowledgements In the context of the pandemic response challenge, codes and data were provided by XPRIZE to build predictors and prescriptions to achieve better intervention support to mitigate between people safety and opening back the economy. We thank XPRIZE for sharing the ideas and codes.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Mathieu E, Ritchie H, Ortiz-Ospina E, Roser M, Hasell J, Appel C, Giattino C, Rod s-Guirao L (2021) A global database of COVID-19 vaccinations. *Nat Hum Behav* 5(7):947–953
2. Miikkulainen R, Francon O, Meyerson E, Qiu X, Sargent D, Canzani E, Hodjat B (2021) From prediction to prescription: evolutionary optimization of nonpharmaceutical interventions in the COVID-19 pandemic. *IEEE Trans Evol Comput* 25(2):386–401
3. Kermack WO, McKendrick AG (1927) A contribution to the mathematical theory of epidemics. *Proc R Soc Lond. Ser A, Contain Pap Math Phys Charact* 115(772):700–721
4. Whitley D (1994) A genetic algorithm tutorial. *Stat Comput* 4(2):65–85
5. Ardabili SF, Mosavi A, Ghamisi P, Ferdinand F, Varkonyi-Koczy AR, Reuter U, Rabczuk T, Atkinson PM (2020) COVID-19 outbreak prediction with machine learning. *Algorithms* 13(10):249
6. Yin R, Tran VH, Zhou X, Zheng J, Kwok CK (2018) Predicting antigenic variants of h1n1 influenza virus based on epidemics and pandemics using a stacking model. *PLoS One* 13(12):0207777
7. Agarwal N, Koti SR, Saran S, Kumar AS (2018) Data mining techniques for predicting dengue outbreak in geospatial domain using weather parameters for New Delhi, India. *Curr Sci* 00:2281–2291
8. Liang R, Lu Y, Qu X, Su Q, Li C, Xia S, Liu Y, Zhang Q, Cao X, Chen Q et al (2020) Prediction for global African swine fever outbreaks based on a combination of random forest algorithms and meteorological data. *Transbound Emerg Dis* 67(2):935–946
9. Hu Z, Ge Q, Li S, Boerwinkle E, Jin L, Xiong M (2020) Forecasting and evaluating multiple interventions for COVID-19 worldwide. *Front Artif Intell* 3:41
10. Zandavi SM, Rashidi TH, Vafae F (2020) Forecasting the spread of COVID-19 under control scenarios using LSTM and dynamic behavioral models. *arXiv preprint [arXiv:2005.12270](https://arxiv.org/abs/2005.12270)*
11. Chandra R, Jain A, Singh Chauhan D (2022) Deep learning via LSTM models for COVID-19 infection forecasting in India. *PLoS One* 17(1):0262708
12. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555)*
13. Peng X, Liu K, Jin Y (2016) A dynamic optimization approach to the design of cooperative co-evolutionary algorithms. *Knowl-Based Syst* 109:174–186
14. Cho K, Van Merri nboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches. *arXiv preprint [arXiv:1409.1259](https://arxiv.org/abs/1409.1259)*
15. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the 13th international conference on artificial intelligence and statistics. JMLR workshop and conference proceedings*, pp 249–256
16. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)*
17. Murata T, Ishibuchi H et al (1995) Moga: multi-objective genetic algorithms. In: *IEEE international conference on evolutionary computation*, vol 1. IEEE, Piscataway, pp 289–294
18. Xue F, Sanderson AC, Graves RJ (2003) Pareto-based multi-objective differential evolution. In: *The 2003 congress on evolutionary computation*, 2003. CEC'03, vol 2. IEEE, pp 862–869

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.