BeCaked$^+$: An Explainable AI Model to Forecast Delta-Spreading Covid-19 Situations for Ho Chi Minh City

Cuong Nguyen[1,2], Minh Nguyen[1,2], Duc Nguyen[1,2], Duc Nguyen[4], Thinh Nguyen[1,2], Khuong Nguyen-An[1,2], Chon Le[1,2,3], and Tho Quan[1,2](✉)

[1] Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
{cuong.nguyen1605,minh.nguyenhuynh.2257,duc.nguyenquang,ntthinh, nakhuong,ltchon,qttho}@hcmut.edu.vn, chonlt@hcmunre.edu.vn
[2] Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam
[3] Ho Chi Minh City University of Natural Resources and Environment (HCMUNRE), Ho Chi Minh City, Vietnam
[4] Hanoi University of Science and Technology (HUST), Hanoi, Vietnam
duc.nq204876@sis.hust.edu.vn

Abstract. Covid-19 is a global disaster that needs computing power to analyze, predict and interpret. So far, there have been several models doing the job. With a huge amount of daily data, deep learning models can be trained to achieve highly accurate forecasts but their mechanism lacks explainability. Epidemiological models, e.g. SIR, on the other hand, can provide insightful analyses, but they require appropriate parameter values, which might be complicated in certain locations.

The fourth wave of the pandemic in *Ho Chi Minh City* (HCMC), Vietnam in 2021, brought valuable lessons along with accurate and specific data. Hence, we introduce an explainable AI model, known as BeCaked$^+$, to predict and analyze the pandemic situation efficiently from the collected data. BeCaked$^+$ combined deep learning and epidemiological models enhanced by specific parameters related to the policies endorsed by the government. Such a combination makes BeCaked$^+$ so accurate and a tool that provides information for policymakers to respond appropriately. One take a try BeCaked$^+$ at http://www.cse.hcmut.edu.vn/BeCaked.

Keywords: BeCaked$^+$ · Covid-19 · Ho Chi Minh City · Delta-variant · Explainable AI · Epidemic model · Time-series forecasting

1 Introduction

SARS-Cov-2 has spread rapidly through most the countries in the world and has caused an infamous global pandemic (Covid-19 pandemic). In Vietnam, the spread of the *Delta* since the fourth Covid-19 wave in May 2021 has caused significant damages and losses, especially in Ho Chi Minh City (HCMC). Though the government has made every effort, such as quarantine (like previous waves) and vaccination, the city still experienced

K. Daimi and A. Al Sadoon (Eds.): ICR 2022, AISC 1431, pp. 53–64, 2022.
https://doi.org/10.1007/978-3-031-14054-9_6

significant deceased cases. Currently, as the situation in HCMC has been under control, a model needs to be developed to forecast and interpret data from the past to prevent similar sufferings in the future.

Typically, there are two major approaches for pandemic forecasting including *compartmental* and *time forecasting* models. While the latter enjoys good accuracy, the former has the advantage of *explainability*. Hence, in 2020 we introduced a prediction model, which combined the *Susceptible-Infectious-Recovered-Deceased* (SIRD) [1] and deep learning, known as *BeCaked* [2]. BeCaked includes two sub-models, a compartmental model in epidemiology, and a deep learning system. When a new outbreak occurred in HCMC in May 2021, we upgraded BeCaked with a new version named BeCaked$^+$, which offers the following contributions: (1) we propose a new model combining compartmental and learning techniques for Covid-19 forecast interpretation; (2) we introduce a set of specific parameters to address the situation in HCMC during the fourth wave; and (3) we implement the model as a Web application where everyone can visit daily at http://www.cse.hcmut.edu.vn/BeCaked.

2 Background Compartmental Models in Epidemiology

SIR model [3] is a classical model in epidemiology. It assumes that the fixed population N is divided into three *compartments* that can vary over time as below:

- S (susceptible): individuals who have not been infected with the disease,
- I (infectious): individuals who are currently infected,
- R (recovered): individuals who have recovered from the disease and have immunity to it.

Initially, most people are susceptible (S), and the others are in the I state. Each person in S interacts with the infectious population and becomes contagious with the effective contact rate of the disease (β). Each infectious individual has the rate of μ of being recovered. These interactions are expressed in Fig. 1(a), assuming that the recovered people cannot be reinfected with the disease (i.e. not get reinfected and not infect others)

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta}{N}SI \\ \frac{dI}{dt} = \frac{\beta}{N}SI - \gamma I \\ \frac{dR}{dt} = \gamma I. \end{cases} \quad (a) \qquad \begin{cases} \frac{dS}{dt} = -\frac{\beta}{N}SI \\ \frac{dI}{dt} = \frac{\beta}{N}SI - \gamma I - \mu I \\ \frac{dR}{dt} = \gamma I \\ \frac{dD}{dt} = \mu I. \end{cases} \quad (b) \tag{1}$$

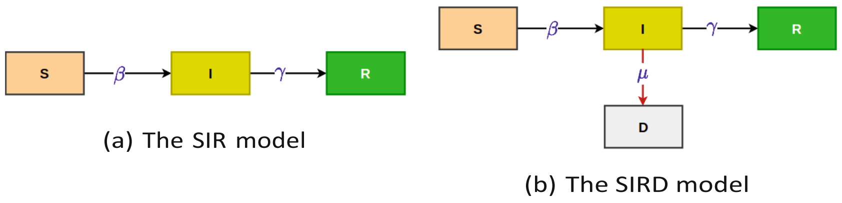
In addition, SIRD is an extension of SIR. Compared to SIR model, SIRD has four states, namely S (susceptible), I (infectious), R (recovered) and D (deceased). The meanings of S, I, and R are the same as those in the SIR model while the D state represents deaths due to the disease. At any given time, t, $S(t)$, $I(t)$, $R(t)$, $D(t)$ denote the number of the individuals in each corresponding compartment, and they satisfy $S(t) + I(t) + R(t) + D(t) = N$. SIRD model is described by the system of equations (1b). Similar to the SIR model, there are relations in this model, which are depicted in Fig. 1(b).

3 Related Works

Pure Time Series Forecasting. In 2020, Hernandez et al. proposed a method [4] that estimated the parameters of *Autoregressive Integrated Moving Average* (ARIMA) [5] to predict the number of infected cases for many countries. Another work by A. Ghany et al. proposed a simple *Multi-layer Perceptron* (MLP) network [6] to forecast the time series data of the epidemic. This method tries to fit the data to the regression models directly without the basis of epidemiological theory. The model returns a beautiful curve fitting with the actual data, but the process is unexplainable. Thus, this direction is a black box, which results in a lack of reliability.

Compartmental Model. Chatterjee et al. [7] improved the SIRD model by reducing the infection rate β over time to reflect the effect of lockdown policies, then they tuned parameters to simulate and analyze the actual situation. Berger et al. [8] proposed a more complex model based on *Susceptible-Exposed-Infectious-Recovered* (SEIR) by considering symptomatic, testing, and vaccination factors. Using models in epidemiology is more explainable since they are built based on epidemiological theory, but the accuracy of this direction does not reach the deep learning models' accuracy as it is less adaptable to new data.

The BeCaked Model. As mentioned in Sect. 1, this work is an improvement of BeCaked that was proposed in 2020. The BeCaked is a model based on Encoder-Decoder architecture. The Encoder is an LSTM network to extract features of epidemic time-series data (i.e number of susceptible cases, infected cases, recovered cases, deceased cases) and return a set of parameters for the SIRD model, then the Decoder takes these parameters and the input to predict the epidemic data in the future. Compared to BeCaked, BeCaked$^+$ upgrades the epidemic model inside Decoder to a more complex model, the Encoder is also integrated with a new network with higher performance. Besides, the method to solve the differential system in Decoder is different, and a new objective function is introduced. The next section will describe the BeCaked$^+$ model in more detail.


$$\begin{tikzcd} S \arrow[r, "\beta"] & I \arrow[r, "\gamma"] & R \end{tikzcd}$$
(a) The SIR model

$$\begin{tikzcd} S \arrow[r, "\beta"] & I \arrow[r, "\gamma"] \arrow[d, "\mu"] & R \\ & D & \end{tikzcd}$$
(b) The SIRD model

Fig. 1. Standard SIR and SIRD model

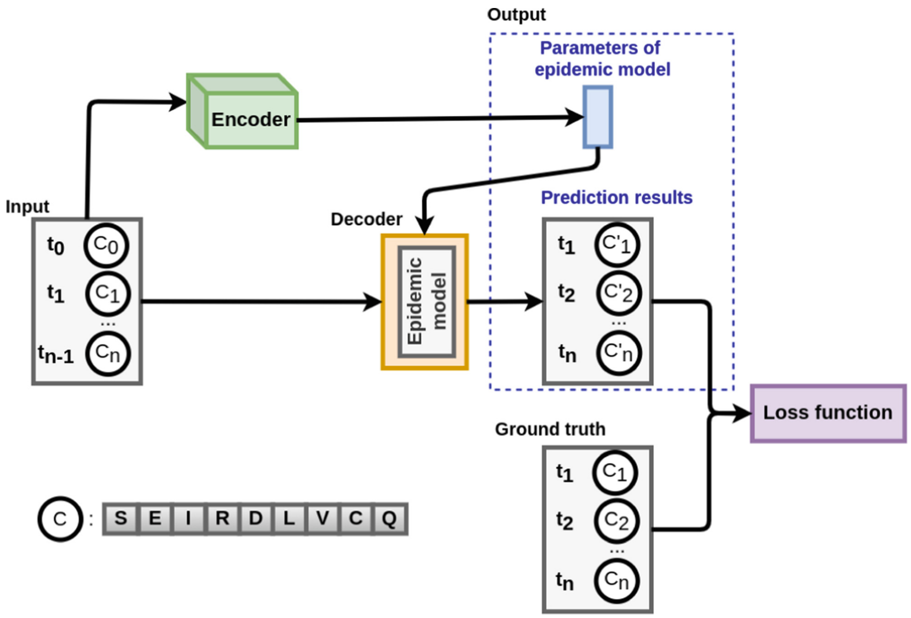


Fig. 2. The overall structure of our method. We use an Encoder block to encode Covid-19 time series data into a set of parameters. This set and the input are then fed to the Decoder block to predict the situations of the next days. These components will be discussed more deeply in later sections.

4 The BeCaked$^+$ Model

4.1 The Overview of BeCaked$^+$

An overview of BeCaked$^+$ is depicted in Fig. 2. Our method comprises an encoder and a decoder, which try to minimize the reconstruction error between ground truth and output from the encoding-decoding process. First, the input data from t_0 day to t_{n-1} day goes through the Encoder to produce the parameters of the epidemic model. The obtained parameters describe the pandemic situation during this period. Then, these parameters and the input data are fed into the Decoder to produce the prediction results from t_1 day to t_n day. Furthermore, the output of Encoder can be analyzed comprehensively by the epidemiologist, which will provide valuable information such as the impact level of each factor.

4.2 BeCaked$^+$ Compartmental Model

This work improves and extends the SIRD model to create a more suitable model that describes the epidemic evolution in HCMC. In addition to the original SIRD model, we also consider other factors such as the effects of restriction, vaccination, quarantine, and critical population. Each individual in the population is assigned to only one of the nine compartments which are S (*Susceptible*), E (*Exposed*), I (*Infected*), R (*Recovered*), D (*Deceased*), L (*Locked*), V (*Vaccinated*), C (*Critical*), and Q (*Quarantined*).

As a compartmental epidemic model, relations between compartments are indispensable, which are depicted in Fig. 3. The dynamics of the pandemic and the constraints between parameters of the model are given in Eq. 2.

$$\begin{cases} M &= S + V + E + R \\ \frac{dS}{dt} &= -\frac{\beta SE}{M} - \tau S - \rho S + \hat{\rho} L - \varphi S + \hat{\varphi} Q \\ \frac{dE}{dt} &= \frac{(\beta S + \zeta V)E}{M} - \eta E - \psi E \\ \frac{dI}{dt} &= \eta E - \gamma_1 I - \theta I + \delta L + \omega Q \\ \frac{dR}{dt} &= \gamma_1 I + \gamma_2 C \\ \frac{dD}{dt} &= \mu C \\ \frac{dL}{dt} &= \rho S - (\delta + \hat{\rho}) L \\ \frac{dV}{dt} &= -\frac{\zeta VE}{M} + \tau S \\ \frac{dC}{dt} &= \theta I - \gamma_2 C - \mu C \\ \frac{dQ}{dt} &= \varphi S + \psi E - (\omega + \hat{\varphi}) Q. \end{cases} \quad (a) \qquad \begin{cases} \gamma_2 + \mu &< 1 \\ \gamma_1 + \theta &< 1 \\ \delta + \hat{\rho} &< 1 \\ \eta + \psi &< 1 \\ \omega + \hat{\varphi} &< 1 \\ \zeta &< \beta. \end{cases} \quad (b) \qquad (2)$$

In short, let $\boldsymbol{x}_t = [S(t), E(t), I(t), R(t), D(t), L(t), V(t), C(t), Q(t)]$, and $\boldsymbol{p}$ be parameters of the epidemic model, we can rewrite the system of Eq. 2 as $\frac{d\boldsymbol{x}_t}{dt} = f(\boldsymbol{x}_t, \boldsymbol{p})$.

Almost all people are in S, E, or I state at first. M denotes the current number of people that can make interaction in the community. The people in S can be vaccinated, quarantined, or moved to restricted areas; in other words, they are transferred into V, Q, or L with the probability of τ, φ, ρ, respectively (corresponding to τ S, φS, ρS in Eq. 2a reflecting population changes). It should be noticed that minus marks indicate decreases and plus signs show increases. Besides, the corresponding probability that a restrictcted area comes back to normal is $\hat{\rho}$ (given in terms of $\hat{\rho} L$). The vaccinated population (V) and susceptible population (S) interacting with the exposed (E) becomes exposed with the rate of ζ and β (corresponding to $\frac{\zeta VE}{M}$ and $\frac{\beta SE}{M}$ where $\frac{E}{M}$ denotes the probability that one interaction between a person in E with another, ζ and β are the probability that a person is exposed after interacting with exposed people). We assume that ζ is lower than β because of the vaccine effectiveness. If ζ is zero, the vaccine offers immunizable protection against the disease. Each exposed person has a rate of ψ of staying in quarantine zones, while the quarantined ones have the rate of $\hat{\varphi}$ that they finish the quarantine period without being positive for Covid-19 (given in terms of ψE and $\hat{\varphi} Q$). The probability that the exposed, locked, and quarantined populations are confirmed to have Covid-19 are η, δ, and ω, respectively (corresponding to ηE, δL and ωQ). Each currently infected case (I) has a rate of $\vartheta\theta$ of getting serious (C) (the term ϑI). The recovery of an infected person and a critical patient is determined by the γ_1 and γ_2 rate, respectively (corresponding to $\gamma_1 I$ and $\gamma_2 C$). The population of critical patients (C) has a death rate of μ and then moves to the D state (the term μC).

4.3 Encoder

We design the Encoder based on *Temporal Convolutional Network* (TCN) [9] to capture the features of all input data (i.e epidemic data of 10 days in the past, which 9 elements per day corresponding to 9 compartments of the compartmental model that will be described in the Sect. 4.2. The way to preprocess the raw data in the real life to obtain this input

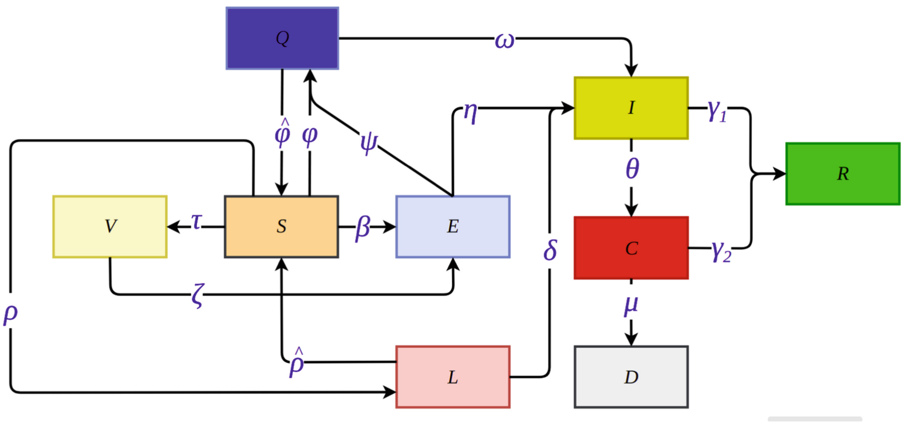


Fig. 3. The relations between the compartments

will be described in Sect. 5.1). TCN applies the *Causal convolution* rather than the usual convolution. Causal convolution is normally used for temporal data to ensure that the prediction at any time step does not depend on the future time steps. Since the input and output of TCN have the same shape, an average-pooling layer is used to reduce the size of the window. Besides, we also need a flattened layer and a fully-connected layer to get the desired output from the Encoder. We propose to use some well-known techniques to improve the performance, which are residual connections and attention mechanisms. Moreover, a regularization, concretely Dropout is applied to prevent overfitting. The output of the Encoder is a set of parameters per day mentioned in Sect. 4.2. Formally, with the input

$$X = \begin{bmatrix} S_0\ E_0\ I_0\ R_0\ D_0\ L_0\ V_0\ C_0\ Q_0 \\ S_1\ E_1\ I_1\ R_1\ D_1\ L_1\ V_1\ C_1\ Q_1 \\ \vdots \\ S_9\ E_9\ I_9\ R_9\ D_9\ L_9\ V_9\ C_9\ Q_9 \end{bmatrix},$$

the Encoder will generate the output

$$P = \begin{bmatrix} \beta_0\ \tau_0\ \rho_0\ \hat{\rho}_0\ \varphi_0\ \hat{\varphi}_0\ \zeta_0\ \eta_0\ \psi_0\ \eta_0\ \gamma_{1_0}\ \gamma_{2_0}\ \theta_0\ \delta_0\ \omega_0\ \mu_0\ \theta_0 \\ \beta_1\ \tau_1\ \rho_1\ \hat{\rho}_1\ \varphi_1\ \hat{\varphi}_1\ \zeta_1\ \eta_1\ \psi_1\ \eta_1\ \gamma_{1_1}\ \gamma_{2_1}\ \theta_1\ \delta_1\ \omega_1\ \mu_1\ \theta_1 \\ \vdots \\ \beta_9\ \tau_9\ \rho_9\ \hat{\rho}_9\ \varphi_9\ \hat{\varphi}_9\ \zeta_9\ \eta_9\ \psi_9\ \eta_9\ \gamma_{9_9}\ \gamma_{2_9}\ \theta_9\ \delta_9\ \omega_9\ \mu_9\ \theta_9 \end{bmatrix}$$

The overall Encoder architecture is shown in Fig. 4.

ResNeXt. To improve the performance of TCN, we suggest using ResNeXt blocks [10] in hidden layers. The main idea of ResNeXt blocks is to perform a set of transformations with the input, and then the intermediate outputs are summed. Each transformation is a different path from the input to the output. Thus, many routes will extract various features, helping the model learn better.

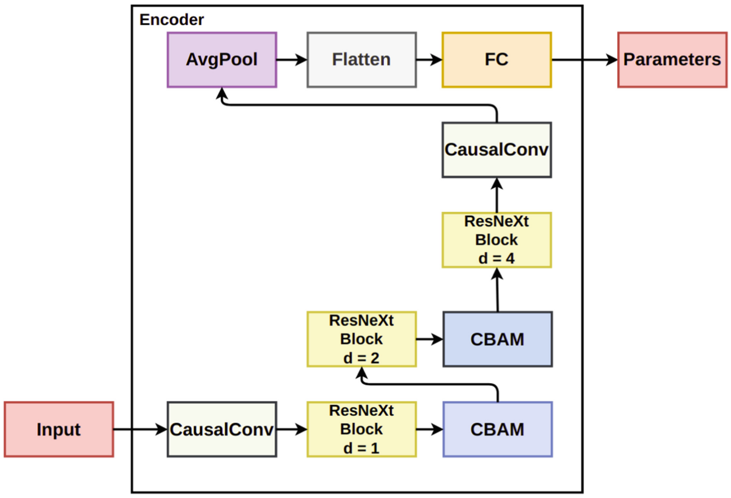


Fig. 4. The overview of the Encoder. In the beginning, we used a Causal Convolution Block to process the input. This block ensures that the output only depends on the previous input (i.e., the number of infectious in the following days can not affect the number of current infections). Next, a ResNeXt Block is used to extract features, and then we apply CBAM to focus on both temporal and channel dimensions. We repeat this process three times with dilation of 1, 2, and 4, respectively, to capture all of the 10-day input based on TCN architecture.

Convolutional Block Attention Module (CBAM). CBAM was proposed by Woo et al. in 2018 [11]; they apply CBAM to improve CNN architectures by inferring attention maps along two separate dimensions, channel and spatial, sequentially. To apply this idea to our model, we use 1D filters instead of 2D ones, and simultaneously two dimensions correspond to channel and temporal. This way, we can concentrate on the prominent features in both dimensions. We briefly summarize this idea as below.

Let F be an intermediate feature map, $F \in \mathrm{R}^{C \times L}$. Our modified CBAM first calculates a channel attention map $M \in \mathrm{R}^{\mathrm{C} \times 1}$, then a temporal attention map $M_t \in \mathrm{R}^{1 \times \mathrm{L}}$.

In the channel attention module, it receives the input F, then squeezes temporal dimension by using both global max-pooling and global average-pooling parallelly. Each branch continues to go through a shared MLP. An addition operation aggregates the outputs of two branches. We receive the channel attention map after the gained output is forwarded to the sigmoid function. Formally, the channel attention operation can be written as

$$\begin{aligned} M_c(F) &= \sigma(MLP(MaxPooling(F)) + MLP(AvgPooling(F))) \\ &= \sigma(MLP(MaxPooling(F) + AvgPooling(F))). \end{aligned}$$

In the temporal attention module, to prepare the input, the feature map F is multiplied by M_c element by element. We also use both global max-pooling and average-pooling operations to aggregate across the channel dimension, then concatenate them to produce temporal descriptors. We apply a causal convolutional layer and sigmoid activation to temporal descriptors to generate a final attention map. The temporal attention can be written as

$$F' = M_c(F) \odot F$$
$$M_t\big(F'\big) = \sigma\big(Conv\big(\big[MaxPooling\big(F'\big); AvgPooling\big(F'\big)\big]\big)\big).$$

where σ is the sigmoid activation and *Conv* is a 1D causal convolutional layer. For brevity, the overall process is equivalent to

$$\begin{aligned} F' &= M_c(F) \odot F, \\ F'' &= M_t\big(F'\big) \odot F', \end{aligned} \tag{3}$$

where $\odot$ is Hadamard product and F'' is the refined output.

4.4 Decoder

The Decoder receives the input data in the interval from t_0 day to t_{n-1} day (X) as well as the output of the Encoder (P) to produce the prediction results from t_1 day to t_n day ($\hat{Y}$). The nature of the Decoder is to solve ordinary differential equations, which is done by the semi-implicit Euler [12] method to avoid the stiffness of the system.

4.5 Parameter Estimation

We use *Weighted Mean Square Error* to compute the gap between the predicted output and the actual data, each compartment has a different weight

$$\mathcal{L}_1 = \frac{1}{n}\sum_{t=1}^{n} w\big(x_t - \hat{x}_t\big)^2,$$

with $\boldsymbol{w}$ is a vector including weights for each compartment while $\boldsymbol{x}_t$, $\hat{\boldsymbol{x}}_t$ are vectors from actual data and predicted output respectively:

$$\begin{aligned} \boldsymbol{x}_t &= [S(t), E(t), I(t), R(t), D(t), L(t), V(t), C(t), Q(t)], \\ \hat{\boldsymbol{x}}_t &= [\hat{S}(t), \hat{E}(t), \hat{I}(t), \hat{R}(t), \hat{D}(t), \hat{L}(t), \hat{V}(t), \hat{C}(t), \hat{Q}(t)]. \end{aligned}$$

However, $_1$ only considers the relation between the input data and the predicted result according to the system of equations (2a). We adopt another loss based on the idea of *Physics-informed neural networks* [13] to constrain the relation between prediction at time-step t and one at time-step $t + 1$ tightly according to the system (2a).

$$\mathcal{L}_2 = \frac{1}{n-1}\sum_{t=1}^{n-1} w\left(\frac{dx_t}{dt} - f(x_t, p)\right)^2.$$

Besides, to satisfy the conditions (2b), we consider an L2 regularization loss L_3 to penalize whenever these constraints are violated. In summary, the final loss is

$$L = L_1 + \lambda_1 L_2 + \lambda_2 L_3,$$

where λ_1 and λ_2 are adjustable weights.

5 Empirical Evaluation

5.1 Dataset, Metrics, and Setting

We use the actual data provided by Ho Chi Minh Center for Disease Control (HCDC). The raw data includes daily reports about the epidemic in HCMC in particular, and we then process them to match the model's input format. In more detail,

– $R(t)$, $D(t)$, $V(t)$ are the total recovered cases, total deceased cases, total vaccinated human at the time t, respectively. They are calculated by accumulating the daily data.
– $I(t)$ is the current infected cases, calculated by the total number of infections, excluding the total number of recovered cases and the number of deceased.
– $L(t)$, $Q(t)$, $C(t)$ are the current number of people living inside lockdown regions, number of people living inside quarantine zones, and emergency cases, respectively. They are reported clearly and do not need to be preprocessed.
– $E(t)$ is the current number of people in latent time (exposed) and is estimated to be four times $I(t)$, based on the result [14] revealing the exposure time of the Delta variant approximate four days.

From our dataset, we split it into training and testing sets by the rates of 80_20, respectively, and the testing set consists of the actual data over four consecutive weeks. We use Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) as metrics to evaluate on four indices, namely I, R, D and C. Our source code is implemented in PyTorch. To train the model, we use Adam optimizer [15] with a cosine annealing scheduler [16]. The initial learning rate is 0.0001, and the minimum one is 0.0000001.

5.2 Baseline Models

The typical ARIMA model is used to compare with our method. ARIMA is a well-known model for time-series forecasting. Nevertheless, ARIMA is a univariate model that can only receive and predict one variable, so we build a different ARIMA model for each index. In addition, we also adopt an RNN-based baseline that is a two-layer LSTM and a multivariate single-step model (named Pure-LSTM, which is similar to the method of A Ghany mentioned in the Sect. 3). It forecasts all four indices simultaneously. To measure the effectiveness of the Encoder based on TCN, we also consider *L-BeCaked*, a model that is quite similar to BeCaked$^+$, but its encoder is based on LSTM.

5.3 Quantitative Results

Both MSE and MAPE indicate that the higher an error, the further the distance between the prediction and the actual data. As shown in Table 1, our method performs better than the others. Though Pure-LSTM is more complex than ARIMA, it is less effective. Nevertheless, there is a considerable gap between ARIMA and our method. Especially when HCMC has attempted to reduce severe cases per day, predicting the number of critical cases is essential, for which our approach is well qualified.

5.4 Model Analysis

To better understand the advantages of our method, it should be noted again that BeCaked$^+$ is explainable as further discussed.

Table 1. Performance of BeCaked$^+$ is consistently better than other baselines.

Model	MAE				MAPE			
	I	*R*	*D*	*C*	*I*	*R*	*D*	*C*
ARIMA	9707.786	12519.643	458.571	704.036	0.062	0.050	0.027	0.615
Pure-LSTM	10239.042	21853.690	431.653	765.448	0.066	0.088	0.026	0.661
L-BeCaked	9103.990	**2320.005**	1852.554	954.378	0.058	**0.009**	0.111	0.822
BeCaked$^+$	**3663.312**	6481.097	**309.719**	**122.656**	**0.023**	0.026	**0.019**	**0.104**

Explainability of BeCaked$^+$. BeCaked$^+$ can serve not only as a regression system with good fit curves for compartments of the epidemic model (i.e Infected, Exposed, etc.) but also as an analytical system to generate the parameters of the epidemic model with an explanation of the regressed value at the same time. Therefore, the parameters of the epidemic model can be used by epidemiologists to comprehensively analyze the factors influencing the evolution of the pandemic. For instance, if the number of infected people becomes alarming at present, we can study and figure out the major reasons causing such a situation. Hence, we have solid bases to alleviate the pandemic.

As mentioned above, one key reason why our approach can be explainable is thanks to the parameters of the epidemic model. Given the input data, the Encoder not only generates the parameters reflecting the pandemic situation but also makes sure these parameters are highly accurate and reliable. The Encoder captures the relationships between days in the input data, e.g, increases or decreases in states, and encodes them in terms of the parameters of the epidemic model.

To explore the parameters, we can apply a method, called sensitivity analysis. It is usually used to figure out how changes in input variables affect target variables. In this work, we use the LHS-PRCC method to estimate the sensitivity of the model's parameters. In particular, a high sensitivity value indicates that the corresponding parameter has a significant impact on figures of the pandemic. Thus, we focus more on solutions that can mitigate the effects of the parameters which are making the pandemic worse. The obtained results are shown in Fig. 5.

Our work shows that the number of infected cases mostly depends on quarantine (corresponding to ω, φ, $\hat{\varphi}$), vaccination (corresponding to τ), and lockdown (corresponding to ρ). The significant sensitivities of ω and φ with the number of infected cases indicate that most people in quarantine areas are more likely to be infected. Significantly, the contact rates (corresponding to β and ζ) do not significantly affect the number of infected cases. This can be explained by the city enforcing a curfew during the period

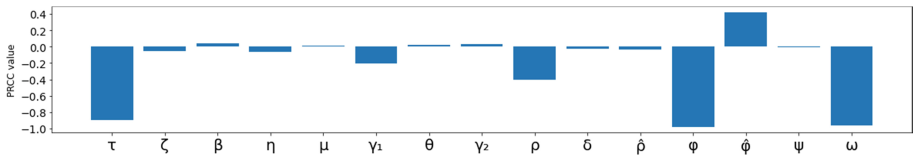


Fig. 5. PRCC scores of parameters with I compartment. The closer the absolute value of this point is to one, the more sensitive the corresponding parameter is to the model.

we apply the model, people having fewer interactions with each other, or the vaccination has probably reduced the risk of infection by contact.

Looking at this result, we can suggest to policymakers that our city can release the lockdown carefully and slowly while also improving the quarantine strategy.

6 Conclusion

We proposed BeCaked$^+$, an extended model of the SIRD integrated with Deep Learning to estimate the parameters describing the evolution of the pandemic. The results provided by our method are more accurate and liable than other baselines. Another advantage of BeCaked$^+$ is the explainability, which is illustrated to explain the situation of HCMC during the fourth wave starting from May 2021. Furthermore, BeCaked$^+$ can give results displayed visually for end-users and can be used by epidemiologists to analyze the evolution of the pandemic.

Acknowledgement. This research is funded by the Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, under grant number **SVKSTN-2021-KH&KTMT-41**. We acknowledge the support of time and facilities from HCMUT, VNU-HCM for this study. We also would like to express our sincerest gratitude to Mr. Hong Son Bui, Prof. Hung Son Nguyen, Dr. Truong-Minh Vu, Dr. Thu Anh Nguyen, Dr. Viet-Cuong Nguyen, and Mr. Minh Canh Khuu for giving us the useful and valuable actual data, servers, and advice.

References

1. Bailey, N.T., et al.: The Mathematical Theory of Infectious Diseases and Its Applications, 2nd edn., Charles Griffin & Company Ltd., High Wycombe (1975)
2. Nguyen, D., et al.: BeCaked: an explainable artificial intelligence model for COVID-19 forecasting. Sci. Rep. (2022). https://doi.org/10.21203/rs.3.rs-454474/v1
3. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proc. R. Soc. London. Ser. A, Contain. Pap. Math. Phys. Character **115**(772), 700–721 (1927)
4. Hernandez-Matamoros, A., et al.: Forecasting of COVID-19 per regions using ARIMA models and polynomial functions. Appl. Soft Comput. **96**, 106610 (2020)
5. Saboia, J.L.M.: Autoregressive integrated moving average (ARIMA) models for birth forecasting. J. Am. Stat. Assoc. **72**(358), 264–270 (1977)
6. Ghany, K.K.A., et al.: COVID-19 prediction using LSTM algorithm: GCC case study. Inform. Med. Unlocked **23**, 100566 (2021)

7. Chatterjee, S., Sarkar, A., Chatterjee, S., Karmakar, M., Paul, R.: Studying the progress of COVID-19 outbreak in India using SIRD model. Indian J. Phys. **95**(9), 1941–1957 (2020)
8. Berger, D., et al.: Testing and reopening in an SEIR model. Rev. Econ. Dyn. **43**, 1–21 (2020)
9. van den Oord, A., et al.: WaveNet: a generative model for raw audio. SSW **125**, 2 (2016)
10. Xie, S., et al.: Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987–5995 (2017)
11. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
12. Hu, Y.: Semi-implicit Euler-Maruyama scheme for stiff stochastic equations. In: Körezlioğlu, H., Øksendal, B., Üstünel, A.S. (eds) Stochastic Analysis and Related Topics V. Progress in Probability, vol. 38, pp. 183–202. Birkhäuser Boston, Boston (1996). https://doi.org/10.1007/978-1-4612-2450-1_9
13. Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving non- linear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
14. Li, B., et al.: Viral infection and transmission in a large, well-traced outbreak caused by the SARS-CoV-2 Delta variant. medRxiv (2021)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
16. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. Learning, vol. 10, p. 3