Assessing Covid-19 Situation in Vietnam Using a Data-Driven Epidemiological Compartmental Model

Vo Le Tung
Assessor: Assoc. Prof. Huynh Trung Hieu
Co-Assessor: Assoc. Prof. Nguyen Tuan Duc

Vietnamese - German University
Frankfurt University of Applied Sciences

January 23, 2022

Outline

1. Introduction
2. Literature review
3. Methodologies
4. Results
5. Discussion
6. Conclusion

Outline for Introduction

1. Introduction

The Covid-19 disease

- 230 million infections
- 4.7 million deaths
- Economic impacts
- Mental impacts

Why do we need a model?

- Lack of research on the impacts of Non-Pharmaceutical Intervention (NPI) in Vietnam
- Need of a simple tool for assessing the disease current and future progression

Goals

- A model that works with the data availability in Vietnam
- An explainable model that can be used by experts
- Assessment of the effectiveness of mobility data in predicting future cases

Outline for Literature review

2 Literature review

- Related works
- Artificial Neural Network

Mathematical models I

Compartmental models [1]

Susceptible-Exposed-Infective-Removed (SEIR) model

$$
\begin{aligned}
S' &= -\beta(N)SI \\
E' &= \beta(N)SI - \kappa E \\
I' &= \kappa E - \alpha I \\
R' &= f\alpha I \\
N' &= -(1-f)\alpha I
\end{aligned}
$$

Mathematical models II

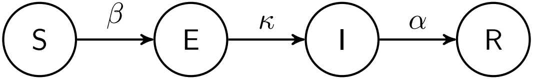


Figure: Graph of transitions between each compartment in the SEIR model

Mathematical models III

Changes made to classical compartmental model for Covid-19 [2]–[6]:

- Inclusion of compartments that represent government interventions
- Inclusion of compartments that specifically represent the behavior of SARS-NCOV-2
- Separation of the infectious compartment into multiple compartments representing the severity of the patients

Mathematical models IV

Agent-based models [7]–[9]

- Model of individualistic behaviors
- Based on population size, age structure, transmission networks

Mathematical models V

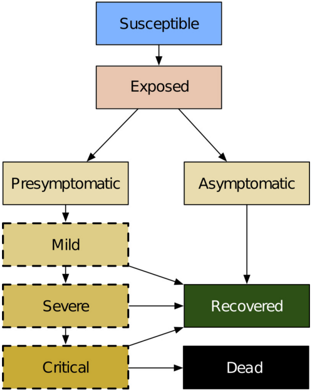


Figure: Agent states used in Covasim [7]

Mathematical models VI

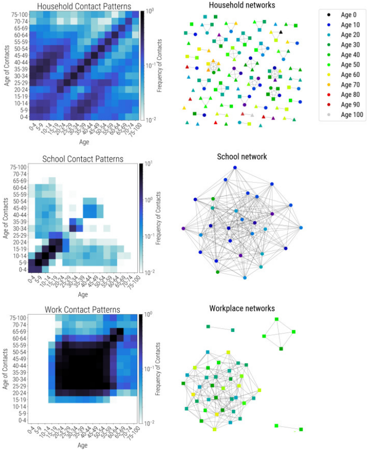


Figure: Transmission networks used in Covasim [7]

Mathematical models VII

Pros

- Explainable
- Based on many years of research
- Easy to understand and implement

Cons

- Low representational capabilities
- The represented dynamics are stationary
- Unrealistic assumptions about real-world scenarios

Data-driven models I

Autoregressive Integrated Moving Average (ARIMA) models [10]–[12]

- Suitable for modeling changing trends, periodic changes, and random noises
- Suitable for different types of data
- Can capture temporal dependency of time series

Data-driven models II

Recurrent Neural Network (RNN) models [13], [14]

- Long Short Term Memory (LSTM)
- Bidirectional Long Short Term Memory (Bi-LSTM)
- Gated Recurrent Unit (GRU)

Explainable Artificial Neural Network (ANN) encoder [15]

Data-driven models III

Pros

- High prediction accuracy
- Allow for modeling without needing domain knowledge

Cons

- Unexplainable
- Relied on large amount of data
- Inability to capture the true disease dynamics

Novel compartmental models I

Mitigate the disadvantages of compartmental models by incorporating covariates

- using predefined functions based on domain knowledge
- using neural networks

Novel compartmental models II

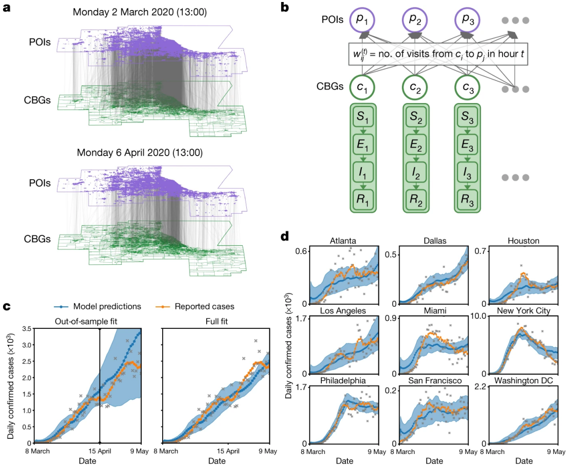


Figure: SEIR model weighted using complex mappings of point-to-point mobility data [16]

Novel compartmental models III

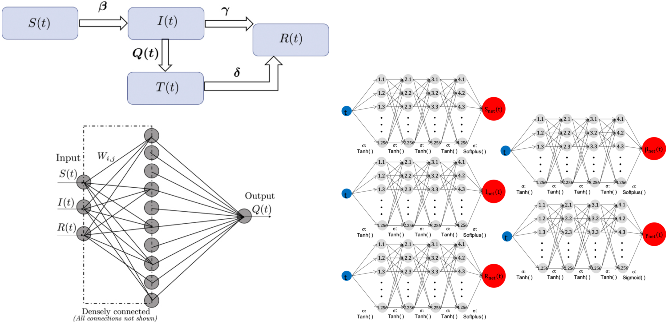


Figure: Using neural networks to predicts the parameters for compartmental models. On the left is the schematic for the QSIR model [17], on the right is the schematic for the time-dependent SIR model [18].

Multi-layer perceptron I

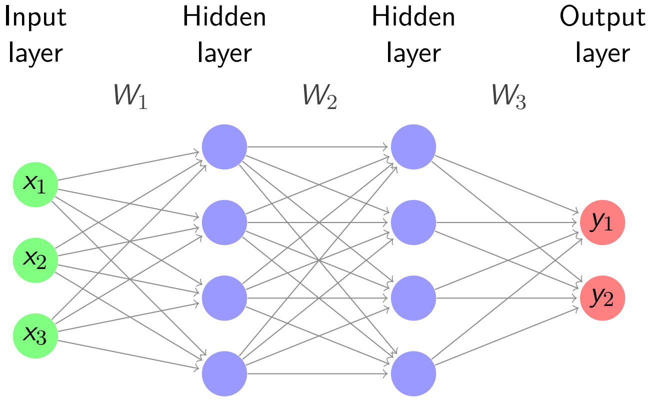


Figure: Graph representation of a multi-layer perceptron with four layers

Multi-layer perceptron II

Perceptron

$$a_i^k = b_i^k + \sum_{j=1}^{r_{k-1}} w_{ij}^k z_j^{k-1}$$

$$z_i^k = \phi(a_i^k)$$

Multi-layer perception

$$g(X) = \phi_n(W_n\phi_{n-1}(\cdots(W_2\phi_1(W_1X + b_1) + b_2) + \cdots) + b_n)$$

Theorem

Given appropriate weights, ANNs can approximate any arbitrary function $f : \mathbb{R}^M \to \mathbb{R}^N$

Multi-layer perceptron III

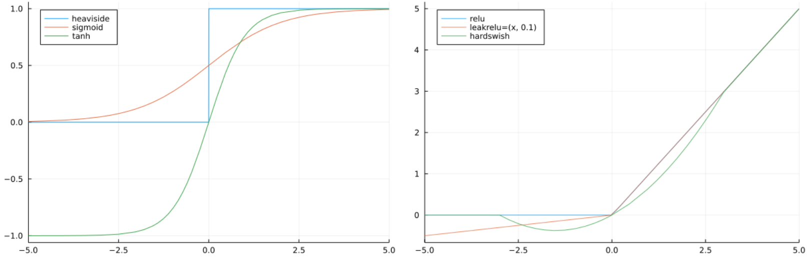


Figure: A visual comparison between the outputs of Heaviside, sigmoid, tanh, ReLU, Leaky ReLU, and hard swish activation functions

Multi-layer perceptron IV

Mean Squared Error (MSE)

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(g(X_i) - Y_i)^2$$

Multi-layer perceptron V

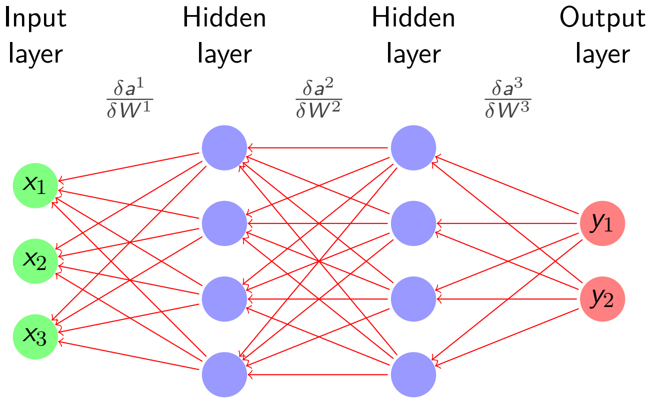

Input layer
Hidden layer
Hidden layer
Output layer
$\frac{\delta a^1}{\delta W^1}$
$\frac{\delta a^2}{\delta W^2}$
$\frac{\delta a^3}{\delta W^3}$
x_1
x_2
x_3
y_1
y_2

Figure: Graph representation of the back-propagation algorithm on a multi-layer perceptron with four layers

Physics-informed neural networks I

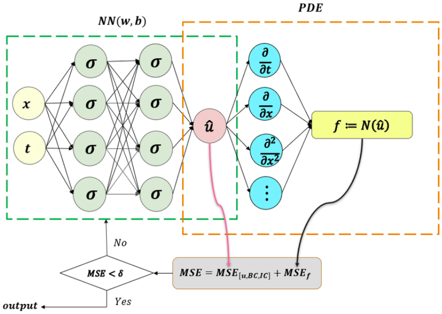


Figure: The schematic of PINNs for solving PDEs [19].

Neural ordinary differential equations I

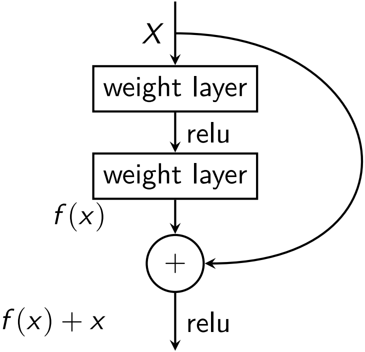


Figure: Example of a skip connection in residual network

Neural ordinary differential equations II

Sequence of modifications to a hidden state

$$h_{t+1} = h_t + f(h_t, \theta_t)$$
$$\frac{dh(t)}{dt} = f(h(t), t, \theta).$$

NeuralODE output and loss

$$\begin{aligned} L(z(t_1)) &= L\left(z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta) dt\right) \\ &= L(ODESolve(z(t_0), f, t_0, t_1, \theta)). \end{aligned}$$

Neural ordinary differential equations III

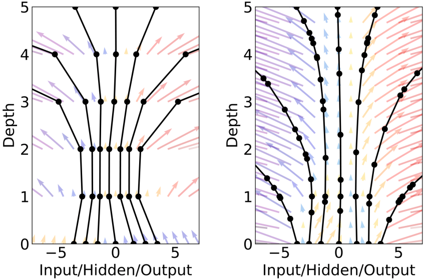


Figure: *Left*: A residual network defines discrete sequence of transformations. *Right*: A ODE network defines a vector field, which continuously transform the state. *Both*: Circles represent evaluation locations. [20])

Neural ordinary differential equations IV

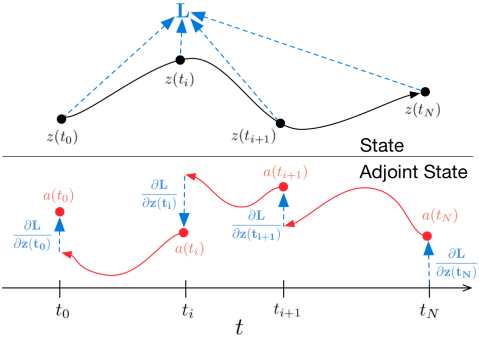

L
$z(t_0)$
$z(t_i)$
$z(t_{i+1})$
$z(t_N)$
State
Adjoint State
$a(t_0)$
$\frac{\partial L}{\partial z(t_0)}$
$\frac{\partial L}{\partial z(t_i)}$
$a(t_i)$
$a(t_{i+1})$
$\frac{\partial L}{\partial z(t_{i+1})}$
$a(t_N)$
$\frac{\partial L}{\partial z(t_N)}$
t_0
t_i
t_{i+1}
t_N
t

Figure: Reverse-mode differentiation of an ODE solution. [20]

Neural ordinary differential equations V

- Memory efficiency
- Adaptive computation
- Scalable and invertible normalizing flows
- Continuous time series models

Outline for Methodologies

3 Methodologies

- Model definition
- Parameters estimation
- Datasets
- Experiments
- Evaluation metrics
- Software and hardware

Universal differential equation I

Universal Differential Equation (UDE) [21] is based on Neural Ordinary Differential Equation (NeuralODE)

Generalized UDE

$$u' = f(u, t, U_\theta(u, t))$$

Universal differential equation II

UDE system

$$
\begin{aligned}
S' &= -\frac{\beta(t)SI}{N} \\
E' &= \frac{\beta(t)SI}{N} - \gamma E \\
I' &= \gamma E - \lambda I \\
R' &= (1 - \alpha(t)) * \lambda * I \\
D' &= \alpha(t) * \lambda * I \\
N' &= -\alpha * \lambda * I \\
C' &= -C + \gamma * E \\
T' &= \gamma * E
\end{aligned}
$$

Universal differential equation III

UDE system

$$\beta(t) = \mathcal{N}\mathcal{N}_{\theta_1}(\mathcal{F})$$

$$\alpha(t) = \mathcal{N}\mathcal{N}_{\theta_2}(\frac{t}{t_{\text{max}}}, \frac{I(t-1)}{N(t-1)}, \frac{R(t-1)}{N(t-1)}, \frac{D(t-1)}{N(t-1)})$$

- Hidden layers activation: mish [22]
- Output layer activation $\nu_i = \nu_{i,L} + (\nu_{i,U} - \nu_{i,L}) * \sigma(z_i)$

Universal differential equation IV

Set of covariates 01

$$\mathcal{F}_1(t) = \left\{ \frac{t}{t_{\text{max}}}, \frac{S(t-1)}{N(t-1)}, \frac{E(t-1)}{N(t-1)}, \frac{I(t-1)}{N(t-1)} \right\},$$

Set of covariates 02

$$\mathcal{F}_2(t) = \mathcal{F}_1(t) \cup \{\text{MovementRange}(t)\},$$

Set of covariates 03

$$\mathcal{F}_3(t) = \mathcal{F}_2(t) \cup \{\text{SocialProximityToCases}(t)\},$$

Loss function

Trainable parameters: $(\gamma', \lambda', \theta_1, \theta_2)$

Box constraints for system parameters

$$\gamma = \gamma_L + (\gamma_U - \gamma_L) * \sigma(\gamma'),$$
$$\lambda = \lambda_L + (\lambda_U - \lambda_L) * \sigma(\lambda').$$

Regularized MSE with scaled outputs

$$\mathcal{L}(\hat{y}, y) = \frac{1}{T} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \left[e^{\zeta t} \left(\frac{\hat{y}_{i,t} - y_{i,t}}{max(y_i) - min(y_i)} \right)^2 \right] + \frac{\lambda}{2T} (\|\theta_1\|_2^2 + \|\theta_2\|_2^2)$$

Training process I

- Partially observed data for $\{D, C, T\}$
- *Tsit5* solver (Tsitouras 5/4 Runge-Kutta method [23])
- *InterpolatingAdjoint* sensitivity algorithm for automatic differentiation

Training process II

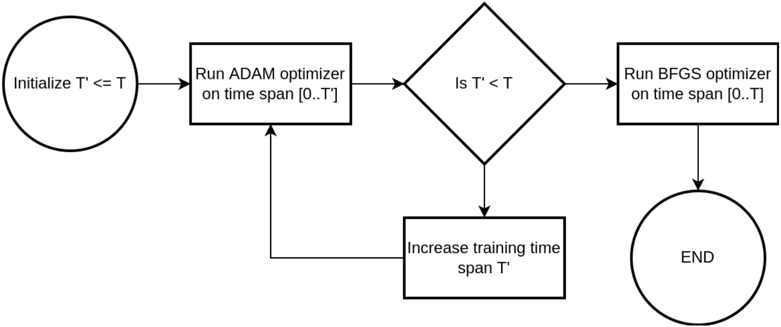

Initialize T' <= T
Run ADAM optimizer on time span [0..T']
Is T' < T
Run BFGS optimizer on time span [0..T]
Increase training time span T'
END

Figure: Model training procedure

Covid-19 cases time series

date	infective	confirmed	recoveries	deaths
1/22/20	0	0	0	0
1/23/20	2	2	0	0
1/24/20	2	2	0	0
...	...	...	...	...

Table: Structure of the processed Covid-19 time series datasets

.

- John Hopkins University (JHU) Covid-19 public datasets
- *VNExpress* Covid-19 public dashboard
- *VnCDC* Covid-19 public dashboard

Facebook's mobility data I

ds	country	polygon	rel. change	stay put ratio
...	...	...	...	...
2021-01-01	VNM	VNM.1.10_1	0.125	0.270
2021-01-02	VNM	VNM.1.10_1	0.052	0.259
2021-01-03	VNM	VNM.1.10_1	0.185	0.269
...	...	...	...	...

Table: Structure of Facebook's Movement Range Maps dataset.

Facebook's mobility data II

Social connectedness index

$$\mathrm{SCI}_{i,j} = \frac{\text{FB connections}_{i,j}}{\text{FB users}_i * \text{FB users}_j},$$

Social proximity to cases [24]

$$\mathrm{SPC}_{i,t} = \sum_{j}^{C} \text{Cases per 10k}_{j,t} \frac{\mathrm{SCI}_{i,j}}{\sum_h^C \mathrm{SCI}_{i,h}},$$

Population data

- Vietnam: Vietnam General Statistics Office (GSO)
- United States: JHU Covid-19 datasets

ID_1	NAME_1	AVGPOPULATION
3	Ha Noi	8.2466e6
62	Vinh Phuc	1.1712e6
16	Bac Ninh	1.4191e6
. . .	. . .	. . .
1001.0	Autauga, Alabama, US	55869
1003.0	Baldwin, Alabama, US	223234
1005.0	Barbour, Alabama, US	24686
. . .	. . .	. . .

Table: Structure of the processed average population datasets

Settings I

- Training period: 48 days
  - Vietnam: First date when the total confirmed cases passed 500
  - United States: 1st July 2021
- Testing period: 28 days
- ADAM optimizer
  - Learning rate 0.05
  - Weight decay rate 0.5 for each 1000 iterations until 0.00001
  - 1000 iterations on each time span
  - Initial time span of 4 days and increased by 4 after each session
- BFGS optimizer
  - *Initial stepnorm*: 0.01
  - 1000 iterations on the full time span

Settings II

- Apply 7-day moving average to all datasets
- Apply min-max scaling on the Movement Range Maps (MRMs) dataset and Social Proximity to Cases (SPC) index

Initial conditions and parameters I

Location	$S(0)$	$E(0)$	$I(0)$	$R(0)$
Vietnam	9.75e7	25	5	2817
Ho Chi Minh city	9.22e6	173.5	34.7	360.1
Binh Duong	2.58e6	206.4	41.2	314.7
Dong Nai	3.17e6	295	59	194.8
Long An	1.71e6	217.8	43.5	300.5
United States	2.99e8	71890	14378	3.31e7
Los Angeles, California	8.78e6	2500	500	1.22e6
Cook, Illinois	4.59e6	615	123	546508
Harris, Texas	4.30e6	930	186	396430
Maricopa, Arizona	3.92e6	1325	265	549899

Table: Initial conditions for the system of ODEs

Initial conditions and parameters II

Parameter	Value	Lower bound	Upper bound
β	N/A	0.05	1.67
γ	$1/4$	$1/4$	$1/4$
λ	$1/14$	$1/14$	$1/14$
α	N/A	0.005	0.05
θ_1	*glorot_normal*	N/A	N/A
θ_2	*glorot_normal*	N/A	N/A

Table: Initial parameters for the system of ODEs

Error functions

Mean absolute error

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

Mean absolute percentage error

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|$$

Root mean squared error

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

Software and hardware

Julia programming language

- *DifferentialEquations* package
- *DiffEqFlux* package

Linux systems

- Google Cloud Compute *n2-standard-8* instance
- Personal laptop with 2-core *Intel(R) Core(TM) i5-4260U* CPU 1.40GHz, and 4Gb of memory.

Julia example

```julia
function SEIRD!(du, u, p, t)
    @inbounds begin
        S, E, I, _, _, N, C, _ = u
        β, γ, λ, α = p
        du[1] = -β * S * I / N
        du[2] = β * S * I / N - γ * E
        du[3] = γ * E - λ * I
        du[4] = (1 - α) * λ * I
        du[5] = α * λ * I
        du[6] = -α * λ * I
        du[7] = -C + γ * E
        du[8] = γ * E
    end
    return nothing
end
```

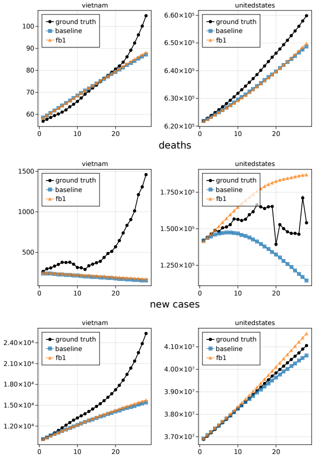
Figure: Implementation of the system of ODEs in Julia

Outline for Results

4 Results

- Model's outputs for Vietnam and the United States
- Model's outputs for counties in the United States
- Model's outputs for provinces in Vietnam

Forecasts for country-level data I



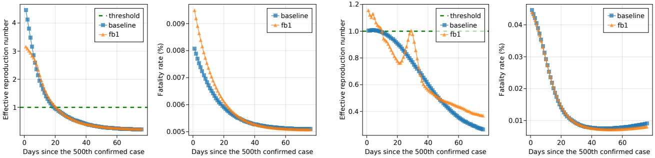
deaths

new cases

total cases

Figure: Forecasts made by all versions of the model

Forecasts for country-level data II



(a) Viet nam

(b) United States

Figure: The effective reproduction number and the fatality rate learned by different versions of the model

Forecasts for counties in the US I

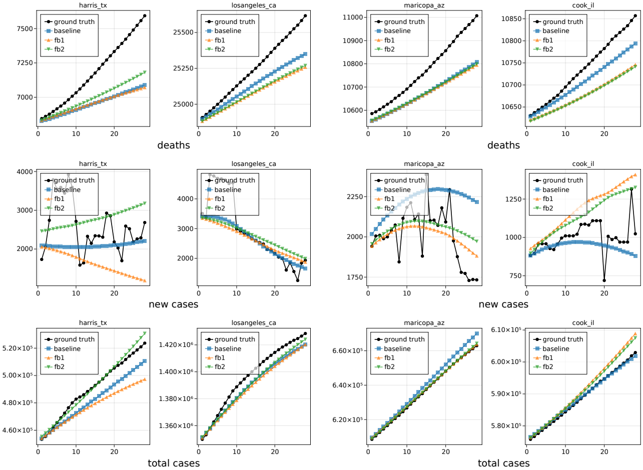
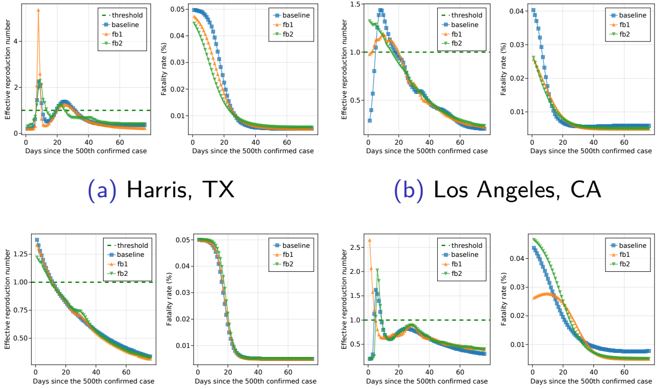


Figure: Forecasts made by all versions of the model

Forecasts for counties in the US II



(a) Harris, TX

(b) Los Angeles, CA

(c) Maricopa, AZ

(d) Cook, IL

Figure: The effective reproduction number and the fatality rate learned by different versions of the model

Forecasts for provinces in Vietnam I

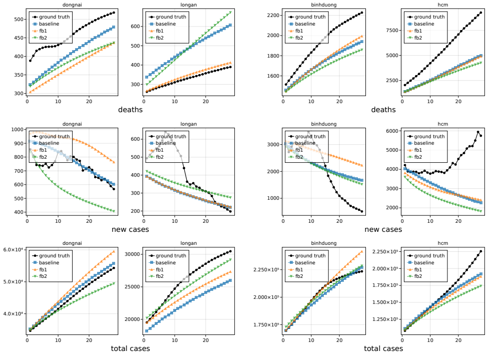
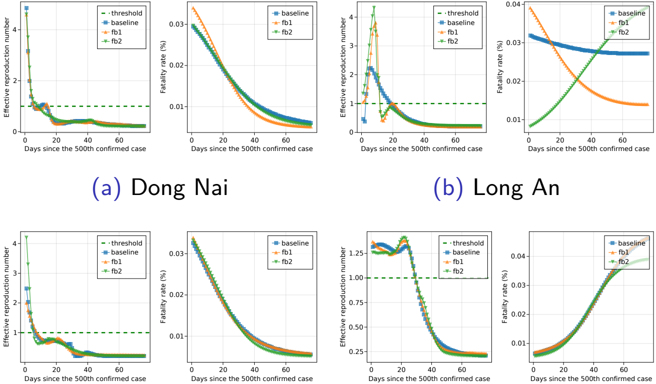


Figure: Forecasts made by all versions of the model

Forecasts for provinces in Vietnam II



(a) Dong Nai

(b) Long An

(c) Binh Duong

(d) Ho Chi Minh city

Figure: The effective reproduction number and the fatality rate learned by different versions of the model

Outline for Discussion

5 Discussion

- Model's convergence and generalizability
- Model's limitations

About the results

- Could capture the disease trends with high accuracy
- Accuracy get lower as we extrapolated further into the future
- Bad fit and low forecast accuracy when applied to data with high fluctuation
- No improvement in the forecast accuracy when mobility data was incorporated

Model's convergence and generalizability

- The model could get stuck in bad local minima, due to
  - Ordinary Differential Equations (ODEs) *stiffness*
  - High frequency changes
- The model was separately trained at each location and only capable of making forecast for the location that it was trained with
- Forecast can only be made for a short period after the training period

Model's limitations

- Issues with ground truth data
- Inability to capture rapid trend changes
- Bad local minima
- Compartmental model bias
- Covariates bias
- Inability to capture real dynamics of the disease
- Neural network interpretability

Outline for Conclusion

What has been done?

Goals: An explainable Covid-19 model that can work with the low data availability in Vietnam

- Collect and process Covid-19 related data for various locations
- Implement a model an explainable Covid-19 model using UDE
  - Progression of all compartments can be given by the model
  - Metric about the disease, e.g. $\mathcal{R}_0$, can be derived
- Add mobility data to the model and show that ineffectiveness of the approach

Future improvements

- Increase the model complexity with additional compartments and parameters
- Include additional time-varying covariates
- Implement better methods and algorithms for training UDE
- Use methods such as Sparse Identification of Nonlinear Dynamical System (SINDy) to find a governing equation and improve the model interpretability

References I

[1] F. Brauer, "Compartmental Models in Epidemiology," in *Mathematical Epidemiology*, ser. Lecture Notes in Mathematics, F. Brauer, P. van den Driessche, and J. Wu, Eds., red. by J. .-. Morel, F. Takens, and B. Teissier, vol. 1945, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 19–79.

[2] S. Zhao and H. Chen, "Modeling the epidemic dynamics and control of COVID-19 outbreak in China," *Quantitative Biology*, vol. 8, no. 1, pp. 11–19, Mar. 2020.

[3] S. He, Y. Peng, and K. Sun, "SEIR modeling of the COVID-19 and its dynamics," *Nonlinear Dynamics*, vol. 101, no. 3, pp. 1667–1680, Aug. 1, 2020.

References II

[4] F. Ndaïrou, I. Area, J. J. Nieto, and D. F. Torres, “Mathematical modeling of COVID-19 transmission dynamics with a case study of Wuhan,” *Chaos, Solitons, and Fractals*, vol. 135, p. 109 846, Jun. 2020.

[5] S. B. Bastos and D. O. Cajueiro, “Modeling and forecasting the early evolution of the Covid-19 pandemic in Brazil,” *Scientific Reports*, vol. 10, no. 1, p. 19 457, Dec. 2020.

[6] K. Sarkar, S. Khajanchi, and J. J. Nieto, “Modeling and forecasting the COVID-19 pandemic in India,” *Chaos, Solitons, and Fractals*, vol. 139, p. 110 049, Oct. 2020.

[7] C. C. Kerr, R. M. Stuart, D. Mistry, *et al.*, “Covasim: An agent-based model of COVID-19 dynamics and interventions,” *PLOS Computational Biology*, vol. 17, no. 7, e1009149, Jul. 26, 2021.

References III

[8] P. C. Silva, P. V. Batista, H. S. Lima, M. A. Alves, F. G. Guimarães, and R. C. Silva, "COVID-ABS: An agent-based model of COVID-19 epidemic to simulate health and economic effects of social distancing interventions," *Chaos, Solitons, and Fractals*, vol. 139, p. 110 088, Oct. 2020.

[9] N. Hoertel, M. Blachier, C. Blanco, *et al.*, "A stochastic agent-based model of the SARS-CoV-2 epidemic in France," *Nature Medicine*, vol. 26, no. 9, pp. 1417–1421, 9 Sep. 2020.

[10] Z. Ceylan, "Estimation of COVID-19 prevalence in Italy, Spain, and France," *Science of The Total Environment*, vol. 729, p. 138 817, Aug. 10, 2020.

References IV

[11] R. K. Singh, M. Rani, A. S. Bhagavathula, *et al.*, "Prediction of the COVID-19 Pandemic for the Top 15 Affected Countries: Advanced Autoregressive Integrated Moving Average (ARIMA) Model," *JMIR Public Health and Surveillance*, vol. 6, no. 2, e19115, May 13, 2020.

[12] M. H. D. M. Ribeiro, R. G. da Silva, V. C. Mariani, and L. d. S. Coelho, "Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil," *Chaos, Solitons & Fractals*, vol. 135, p. 109 853, Jun. 1, 2020.

[13] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons, and Fractals*, vol. 135, p. 109 864, Jun. 2020.

References V

[14] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons, and Fractals*, vol. 140, p. 110 212, Nov. 2020.

[15] A. Ramchandani, C. Fan, and A. Mostafavi, "DeepCOVIDNet: An Interpretable Deep Learning Model for Predictive Surveillance of COVID-19 Using Heterogeneous Features and Their Interactions," *IEEE Access*, vol. 8, pp. 159 915–159 930, 2020.

[16] S. Chang, E. Pierson, P. W. Koh, *et al.*, "Mobility network models of COVID-19 explain inequities and inform reopening," *Nature*, vol. 589, no. 7840, pp. 82–87, 7840 Jan. 2021.

References VI

[17] R. Dandekar, C. Rackauckas, and G. Barbastathis, "A Machine Learning-Aided Global Diagnostic and Comparative Tool to Assess Effect of Quarantine Control in COVID-19 Spread," *Patterns*, vol. 1, no. 9, p. 100 145, Dec. 11, 2020.

[18] S. Y. Jung, H. Jo, H. Son, and H. J. Hwang, "Real-World Implications of a Rapidly Responsive COVID-19 Spread Model with Time-Dependent Parameters via Deep Learning: Model Development and Validation," *Journal of Medical Internet Research*, vol. 22, no. 9, e19907, Sep. 9, 2020.

[19] Y. Guo, X. Cao, L. Bainian, and M. Gao, "Solving Partial Differential Equations Using Deep Learning and Physical Constraints," *Applied Sciences*, vol. 10, p. 5917, Aug. 26, 2020.

References VII

[20] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural Ordinary Differential Equations,” (Dec. 13, 2019), [Online]. Available: http://arxiv.org/abs/1806.07366 (visited on 09/26/2021).

[21] C. Rackauckas, Y. Ma, J. Martensen, *et al.*, “Universal Differential Equations for Scientific Machine Learning,” (Aug. 6, 2020), [Online]. Available: http://arxiv.org/abs/2001.04385 (visited on 09/11/2021).

[22] D. Misra, “Mish: A Self Regularized Non-Monotonic Activation Function,” (Aug. 13, 2020), [Online]. Available: http://arxiv.org/abs/1908.08681 (visited on 12/06/2021).

References VIII

[23] C. Tsitouras, “RungeKutta pairs of order 5(4) satisfying only the first column simplifying assumption,” *Computers & Mathematics with Applications*, vol. 62, no. 2, pp. 770–775, Jul. 2011.

[24] T. Kuchler, D. Russel, and J. Stroebel, “The Geographic Spread of COVID-19 Correlates with the Structure of Social Networks as Measured by Facebook,” National Bureau of Economic Research, Working Paper 26990, Apr. 2020.