What if...? Global COVID policy simulator: improving model performance

Yixin Cheng
321718

Haoyu Sheng
321449

Shijian Xu
327448

***Abstract*—Background: As COVID-19 ravages the world, rationale behind public health epidemic mitigation policies are under intense debate. Aim: This project seeks to improve the predictive performance of the *What if... Interactive Global COVID Policy Simulator*, which simulates the different policies and examines their effects on the reproduction number (R0) prediction. Methods: To achieve the best R0 prediction, we tune the model structure and hyper-parameters. Additionally, we implement *variance inflation factor (VIF)* and correlation based feature selection methods to obtain the best subset of all the features. Results: The best model we had improves the overall MSE score by 25.4% compared with baseline score on multiple countries dataset. Besides, we come out with 2 different feature selection processes. The correlation-based feature selection method achieved 0.07867 MSE on CHE datset. Such models have great potential to help guide evidence based policy making and public understanding on the importance of public health measures**

I. Introduction

A. Problem

Public health policies are essential to contain the spread of COVID-19 pandemic, but incur significant collateral economic and social damage. Policies are a trade-off between lives and livelihoods and have different impacts and efficacy in different populations. To improve the estimations on policy efficacy and find the optimal combinations of policies, the iGH team at EPFL created the – *What if... Interactive Global COVID Policy Simulator*[1] described below.

B. Existing Work

The What if policy simulator is designed as an interactive website platform. People could simulate different combinations of COVID policies in various countries and see the corresponding predicted effects.

Currently, a neural network model has been developed to predict the the basic reproductive number (R0) value based on the collected data of weather, policies, demographics etc. One can understand the concept of R0 by thinking of it as the average number of people infected by one infectious individual[2]. R0 is useful for policymakers since it provides a proxy of how individuals are adhering to policies [3].

There are two main types of features: Time-varying and constant. For each country, the inherent features, like demography and sanitary data, are considered to be constant. The other features, like The evolving statistics about the epidemic, are considered to be variable (time-series). Four main categories of features including features_demography, features_sanitary, features_weather and policies are selected to train the model. Each main category includes several features. The list of main categories and corresponding features used for current model training are listed in Appendix A.

The model consists of two parts, the Long Short-Term Memory (LSTM) part, which is used to handle the variable features, and the multilayer perceptron (MLP) part, which is used to handle the constant features. A fully connected part is appended to concatenate the outputs from these two parts and finally produces the R0 prediction. The detailed structure of the model is shown clearly in the Appendix B.

II. Aim and Objectives

The aim of our project is to further improve the existing model of this policy simulator. Our objectives are following:

1) Tuning the model structure and hyper-parameters in order to make more accurate predictions and reduce the mean square error (MSE).
2) Applying some feature selection methods to remove redundant inputs while optimizing the MSE score.

III. Methods

A. Adjusting Neural Network Structure

In this task, the baseline neural network consists of one LSTM layer and three linear layers. Each linear layer undergoes Monte Carlo dropout[4] after the activation function.

We come up with three possible approaches to adjust the structure:

1) Tuning hyper parameters in the existing neural network. This is the classical approach to find optimal settings. The candidate parameters are
    a) *MC dropout*: The dropout possibility defined in *MC_dropout* layer.
    b) *Neural dropout*: The dropout possibility defined in LSTM or GRU or RNN layer.
    c) *Window size*: Number of past days that will be taken into account. It is defined in model_config file.
    d) *Hidden size*: Sizes of hidden layers for LSTM or RNN or GRU. It is defined in model_config file.
2) Change LSTM layer to other kinds neural networks layers
    a) *Replacing LSTM with RNN*[5]: the RNN layer is the vanilla recurrent network unit, which is essentially a fully connected network, that considers a temporal dimension in its prediction. While

LSTM is usually considered as a better structure than RNN, it also has more parameters and requires heavier computation. Sometimes, like short temporal-sequence prediction, RNN might still be able to get good performance.

b) *Replacing LSTM with GRU*[6]: GRU is a variant of LSTM. It combines the LSTM forget gate and input gate into a single "update gate". It also merges cell states and hidden states. The resulting model is simpler than the standard LSTM model,which means a lower computational complexity and quicker training time.

3) Stacking multiple Layers

Stacked LSTM/RNN/GRU are a more stable technique for solving sequence prediction problems. A stacked architecture can be considered as a model comprised of multiple recurrent layers. One layer above provides a sequence output rather than a single value output to the layer below, which means one output per input time step, rather than one output for all input time steps.

We also modify the number of the LSTM/RNN/GRU layer to test whether this method will get a better performance.

B. Feature Selection

There are a lot of features available in the dataset. When the number of features is large, it is highly possible that there will be multicollinearity. In the simplest cases, multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model. But multicollinearity might also exist even if pairs of variables appear uncorrelated. Fig. 1 shows the correlation matrix of the original features.

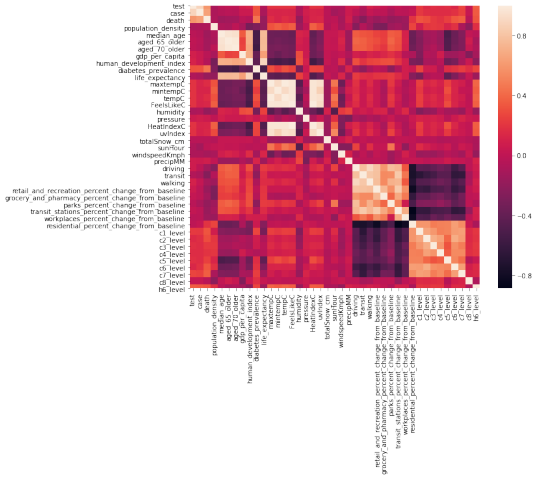

Fig. 1: Correlation matrix of the original features.

It can be clearly seen from Fig. 1 that some features are highly correlated. In this case, feature selection will be helpful. To this end, we refer to use the feature correlations and the variance inflation factors (VIF)[7]. For correlation based selection, the features are selected by setting different correlation thresholds on the Pearson correlation coefficients. For the VIF based selection, the features with VIF scores larger than 5 are considered to be potentially have multicollinearity.

The VIF based feature selection is described as follows. Let X be the design matrix which has p features. VIF is defined as:

$$VIF_j = \frac{1}{1 - R_j^2} \tag{1}$$

where R_j^2 is the coefficient of determination for the regression of feature X_j on all the other features $\{X_1, ..., X_{j-1}, X_{j+1}, ..., X_p\}$, measuring the linear dependence of X_j on the other features of X. Large value of VIF_j indicates that X_j is linear dependent on the other features. In this project, we use the *variance_inflation_factor* function provided in *statsmodels* library [8] to iteratively eliminate those features with large VIF values.

IV. Results and Discussion

Training for all countries to find the best neural network structure will be extremely time consuming. Therefore, we first test on the dataset of one country, then we pick up relatively optimal solutions and apply those models to train on the dataset of all countries. Finally, we get the model that minimizes the overall mean squared error of predictions for all countries. The results in part is based on training for Switzerland (ISO code "CHE").

A. Adjusting Neural Network Dropout Values

MC dropout	NN Types	neural dropout	window size	number of layer	hidden size	MSE
0.5	LSTM (baseline)	0.0	7	1	20	**0.182655**
0.5	LSTM	0.2	7	1	20	0.16902
0.5	LSTM	0.5	7	1	20	0.209217
0.3	LSTM	0.2	7	1	20	0.182655
0.5	GRU	0	7	1	20	0.22617
0.5	GRU	0.2	7	1	20	0.18045
0.3	GRU	0	7	1	20	0.19100
0.3	GRU	0.2	7	1	20	0.16408

TABLE I: Results of Different Dropout Values on CHE. *The best performed dropout combination has been highlighted*

In the existing neural network, dropout exists in two places, one is in the LSTM/RNN/GRU layer, and the other is in the subsequent linear layer. We try to change different dropout values to achieve better prediction results.The results in TableI are based on training for Switzerland(ISO code "CHE").

By adjusting the dropout value, the model improves prediction results and avoids overfitting. In the predictions of multiple countries, the model with a dropout value of not 0 also showed better generalization performance.

B. Change LSTM layer to other kinds neural networks layer

Using one layer default config LSTM as baseline, the training time is shown in Fig.2

Using multi countries' data as input, and the average value of their MSE is used as an indicator to measure the

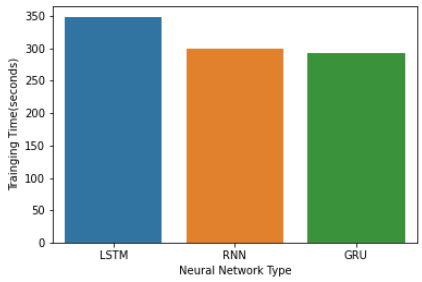

Fig. 2: Comparison of neural network training time in a single country

generalization performance of the network. The results are shown in Fig.3 and Table II.

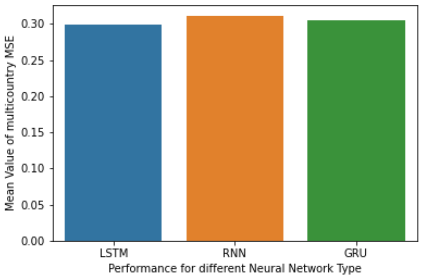

Fig. 3: Comparison of neural network performance in multi country

Neural Network Type	Mean Value of Multi Country MSE
LSTM	0.29854
RNN	0.311294
GRU	0.304491

TABLE II: Results of Different Neural Networks

For the three networks, their performance is similar under the default parameters. Compared with RNN, LSTM increases the filtering of past states, so that the networks can choose which states have more influence on the current, instead of simply choosing the most recent state. LSTM performs better when input contains a large number of sequential features. The advantage of GRU is that GRU has fewer parameters, so training is faster and requires less data to generalize compared to LSTM. The single-layer network LSTM performs best under the default parameters, but this does not mean that the LSTM is the optimal solution. We will continue to discuss it below.

C. Results for Neural Network Structure and Parameters Adjustment

To sum up, by changing the network's neural network type, dropout value, window size, etc, we tried different combinations, and obtained the following results shown in Table III with Swiss data as input. The best result achieved by the current model is 1 layer RNN with 0.3 dropout value.

MC dropout	NN Types	neural dropout	window size	number of layer	hidden size	MSE
0.5	LSTM (baseline)	0.0	7	1	20	**0.182655**
0.3	LSTM	0.5	7	3	20	0.150302
0.3	LSTM	0.2	7	1	20	0.182655
0.3	LSTM	0.2	7	1	10	0.182655
0.3	LSTM	0.5	7	1	20	0.182655
0.3	LSTM	0.2	7	2	20	0.175782
0.3	LSTM	0.5	7	2	20	0.213870
0.3	LSTM	0.2	7	3	20	0.161327
0.3	LSTM	0.2	7	3	10	0.220852
0.3	LSTM	0.5	7	3	10	0.181523
0.3	LSTM	0.5	14	3	10	0.202364
0.5	LSTM	0.5	7	1	20	0.209217
0.5	LSTM	0.5	7	1	10	0.226406
0.5	LSTM	0.0	7	2	20	0.200908
0.5	LSTM	0.2	7	2	20	0.181760
0.5	LSTM	0.5	7	2	20	0.198544
0.5	LSTM	0.0	7	3	20	0.157646
0.5	LSTM	0.2	7	3	20	0.175942
0.5	LSTM	0.5	7	3	20	0.240529
0.5	LSTM	0.5	7	3	10	0.183853
0.3	RNN	0.3	7	1	10	0.11510
0.3	RNN	0.2	7	1	20	0.16155
0.3	RNN	0.2	7	2	20	0.196149
0.3	RNN	0.2	7	3	20	0.287332
0.3	RNN	0.2	7	1	10	0.16112
0.3	RNN	0.2	7	1	5	0.137380
0.3	RNN	0.3	7	1	30	0.148349
0.3	RNN	0.3	7	1	40	0.168066
0.3	RNN	0.3	14	1	10	0.131820
0.5	GRU	0.0	7	2	20	0.163246
0.5	GRU	0.0	7	1	20	0.226170
0.5	GRU	0.2	7	1	20	0.180459
0.5	GRU	0.5	7	1	20	0.180459
0.5	GRU	0.5	7	2	20	0.232609
0.5	GRU	0.2	7	2	20	0.205611
0.5	GRU	0.2	7	3	20	0.172640
0.5	GRU	0.5	7	3	20	0.171242
0.5	GRU	0.0	7	3	20	0.248960
0.3	GRU	0.0	7	1	10	0.191003
0.3	GRU	0.2	7	1	10	0.164081
0.3	GRU	0.5	7	1	10	0.164081
0.3	GRU	0.5	7	2	10	0.175343
0.3	GRU	0.2	7	2	10	0.188547
0.3	GRU	0.2	7	3	10	0.172640
0.3	GRU	0.5	7	3	10	0.171242
0.3	GRU	0.0	7	3	10	0.248960

TABLE III: Results for Neural Network Structure and Parameters Adjustment on CHE Dataset

The MSE value is 0.11510, which is 36.5% higher than the baseline.

D. Results for Feature Selection

For feature selection, we use two different methods to compare their performance, one is correlation based, the other is VIF based. To verify the effectiveness of the selection methods, we take all the possible features into consideration except *continent* and *weekdays*, which seems to be problematic. These methods are tested on country *CHE*. The results are shown in Table IV.

From Table IV we can see that feature selection can improve the results significantly. The interesting thing is, correlation based feature selection seems more effective and efficient than VIF based feature selection.

E. Results for Combined Methods

Both methods could optimize the MSE score. Therefore, we also tried some combinations of adjustments on structure and feature selection. The results are obtained by training on CHE dataset and are shown in Table V.

VIF threshold	# features left	MSE
12	19	0.18307
11	18	0.13326
10	18	0.13326
9	16	0.12846
8	15	0.14574
7	15	0.14574
6	13	training failed
5	11	0.12890
4	10	0.14138
Correlation threshold	**# of features left**	**MSE**
0.9	34	0.15175
0.8	28	0.12319
0.7	23	0.07867
0.6	16	0.10742
0.5	12	0.12924

TABLE IV: Results for Feature Selection on CHE Dataset

MC dropout	**NN Types**	**neural dropout**	**window size**	**number of layer**	**hidden size**	**MSE score**
0.5	LSTM (baseline)	0.0	7	1	20	**0.182655**
0.5	LSTM+VIF	0.0	7	1	20	0.13326
0.5	LSTM+VIF	0.5	7	3	20	0.214790
0.3	LSTM+VIF	0.0	7	1	20	0.201312
0.3	LSTM+VIF	0.5	7	3	20	0.17895
0.3	LSTM+VIF	0.5	14	3	10	0.182265
0.3	RNN+VIF	0.3	7	1	20	0.12355
0.3	RNN+VIF	0.3	7	1	5	0.17210
0.3	RNN+VIF	0.3	7	1	10	0.13182
0.3	RNN+VIF	0.3	7	1	30	0.14429
0.5	RNN+VIF	0.3	7	1	10	0.13670
0.5	GRU+VIF	0.0	7	1	20	0.13649
0.5	GRU+VIF	0.0	7	2	20	0.16934

TABLE V: Results of Combined Methods on CHE Dataset

F. Test on Multi-countries

To find the best model that optimizes overall MSE score, we picked up several optimal models in part A, B, C, D, E to test on the multiple countries dataset. The results have been shown in Table VI.

According to the mean value of MSE, the best model is the neural network which uses vanilla RNN as time-varying inputs layer and with neural dropout 0.3, MC dropout 0.3, hidden size 10. It achieved 0.200241 average MSE on multi countries dataset, which improved 25.4% compared with baseline.

Since there are only four categories of features can be applied on multiple countries dataset. The feature selection method may not work well with limited number of features. Therefore, the model with feature selection method did not work well as we expected.

V. Conclusion

In this project, we tried to expand "What if. . . Interactive Global COVID Policy Simulator" with optimizing the MSE score of predicted results as well as adding a feature selection process. The best model that minimize the overall mean squared error is vanilla RNN with 0.3 MC dropout probability, 0.3 RNN dropout probability, 10 hidden vanilla RNN layers as well as other defaults parameters. The best feature selection process is the correlation based one with threshold 0.7.

The experience of this project is interesting since we have participated in a project which can indeed help to improve human's life during the COVID-19 outbreak. The team in the iGH Lab also helped us a lot during the project. We also learned how to collaborate with each other from them as well as throughout this project.

MC dropout	**NN Types**	**neural dropout**	**window size**	**number of layer**	**hidden size**	**MSE score**
0.5	LSTM (baseline)	0.0	7	1	20	CHE: 0.182655 ESP: 0.278878 YEM: 0.669299 GTM: 0.072906 AFG: 0.227363 AGO: 0.290883 DEU: 0.156955 **mean: 0.268420**
0.3	LSTM	0.5	7	3	20	CHE: 0.1489056 ESP: 0.281121 YEM: 0.817538 GTM: 0.080640 AFG: 0.195136 AGO: 0.296367 DEU: 0.122222 **mean: 0.277419**
0.3	RNN	0.3	7	1	10	CHE: 0.1150978 ESP: 0.144022 YEM: 0.437803 GTM: 0.080272 AFG: 0.181206 AGO: 0.291873 DEU: 0.1514135 mean:0.200241
0.5	GRU	0.0	7	2	20	CHE 0.213577 ESP 0.183410 YEM 0.846917 GTM 0.094965 AFG 0.232436 AGO 0.316414 DEU 0.223007 **mean: 0.301532**
0.5	LSTM+VIF	0.0	7	1	20	CHE 0.227886 ESP 0.171000 YEM 0.454103 GTM 0.100163 AFG 0.215259 AGO 0.280237 DEU 0.190548 **mean: 0.234171**
0.3	RNN+VIF	0.3	7	1	20	CHE 0.140647 ESP 0.231162 YEM 0.476336 GTM 0.106903 AFG 0.206193 AGO 0.284935 DEU 0.201952 **mean 0.235447**
0.5	GRU+VIF	0.0	7	1	20	CHE 0.197939 ESP 0.194228 YEM 0.4519040 GTM 0.093197 AFG 0.237253 AGO 0.294961 DEU 0.237069 **mean: 0.234172**

TABLE VI: Results of Training on Multi Countries Dataset

VI. Limitations and Future Work

Our method for optimizing MSE score mainly focus on replacing the layer which deal with the time-varying variables. In the future, more attempts could be taken to deal with those constant inputs such as adding bias to constant variables etc. Due to the limitation of the time, we tried some combinations of neural network structure and feature selection on the single country dataset. Based on the current result, the best model still remained to be the RNN model without feature selection. In the future experiments, more possible combinations of those two methods could be tested to find the optimal model.

At the same time, due to time constraints, we did not have enough time to modify and try other models, such as Transformer[9]. The Transformer using the self-attention mechanism avoids the recurrent model structure, and its structure is more flexible or versatile than LSTM. If trying the transformer model for this task, it may get better predictions.

APPENDIX A
FEATURES FOR TRAINING

Feature Type	***features_demography***	***features_sanitary***	***features_weather***	***policies***
Features	- population_density - median_age - aged_65_older - aged_70_older - gdp_per_capita - human_development_index	- diabetes_prevalence - life_expectancy	- maxtempC - mintempC - tempC - FeelsLikeC - humidity - pressure - HeatIndexC - uvIndex - totalSnow_cm - sunHour - windspeedKmph - precipMM	- c1_level school closing - c2_level workplace closing - c3_level cancel public events - c4_level gathering size restrictions restrictions - c5_level close public transport - c6_level home confinement orders - c7_level internal movement restrictions - c8_level international travel restrictions - h6_level facial covering

APPENDIX B
MODEL STRUCTURE

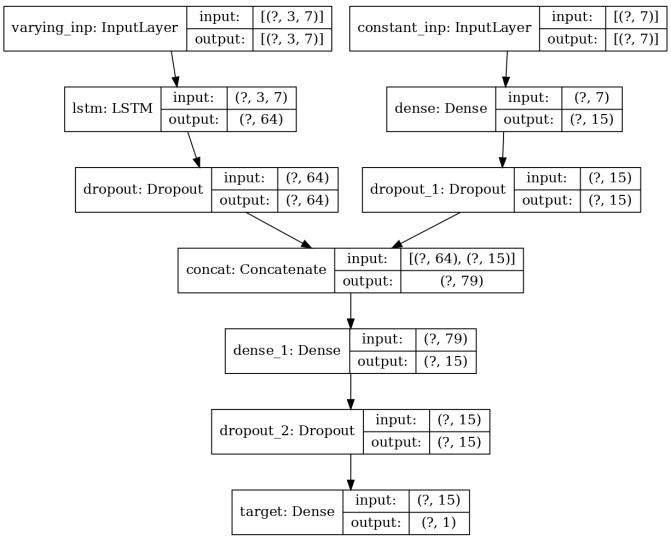

Fig. 4: Model Structure[1]

REFERENCES

[1] T. Bossy, "model.png," 2020, [Online; accessed December 15, 2020]. [Online]. Available: https://github.com/epfl-iglobalhealth/What-if-sandbox/blob/master/modeling/model.png

[2] G. N. Milligan and A. Barrett, *An Essential Guide*. Wiley Online Library, 2015.

[3] B. Ridenhour, J. M. Kowalik, and D. K. Shay, "Unraveling r 0: Considerations for public health applications," *American journal of public health*, vol. 108, no. S6, pp. S445–S454, 2018.

[4] S. Wang and C. Manning, "Fast dropout training," in *international conference on machine learning*, 2013, pp. 118–126.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[7] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, 1998, vol. 326.

[8] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, pp. 5998–6008, 2017.