Report: Short Term and Long Term Cases Forecasting of COVID-19 with Varies Neural Networks

Subject: Machine Learning Research Project

Author: Yunfei Luo
Date: Jun 11th, 2020

Abstract

As the novel virus had bothered people all over the world for months, finding the factors that influence the spreading pattern becomes a series issue that people are studying and discussing about. In this project, we explored and developed a general platform for the forecasting of the trends of increasing in the number of confirmed cases, death cases, and recovered cases in different regions with varies models including Linear Models, Recurrent Neural Networks, and Deep Multitask Networks. The experimental results shows that the performance of Recurrent Neural Networks is the most stable one, especially the Long-Short-Term-Memory (LSTM); the performance of Multitask models might not be the best, but they have better ability to catch the periodical features of the time series; and finally, Linear regression shows excellent performance in the scenario of long term prediction. Such platform could have much potential applications, where people could have a rough idea of the epidemic situation at the regions they're living in. The platform offers a significant factor that can be taking into account when people need to going out.

1. Methods

1.1. Problem Setup and Data Preparation

1.1.1. DATASET

Johns Hopkins University (JHU) offers the every day's statistical data of the number of cases across majority regions in the world. With the time series data of the numbers of confirmed cases, death cases, and recovered cases, we are able to train models fed by these series.

1.1.2. DATA PREPROCESSING

For the convenience of analysis and debugging, we need formative data. We first read the time series data for each region offered by JHU. So for each region we have data in shape $(n, 3)$ where n is the number of days, and the three entries of each vector correspond to confirmed cases, death cases, and recovered cases. We then take differences across the vectors, where now each time step t_i becomes $t_i - t_{i-1}$, and the initial time step t_0 remain unchanged. Then each entry now represents the newly increased number.

In order to deal with the imbalance in the ranges of data of different regions, we normalize the data by min-max scaling, where we map the three features (cases of confirmed, death, and recovered) in the data of each region to $[0, 1]$.

We then define a window size of 14. Since this is a forecasting tasks with time series data, if we didn't define the window size and use the entire series for training, there will be too many features and noises but poor number of trained data fed to the models. So for each time step begin at t_{15}, we form a sample by the previous 14 time steps, and set the current time step to be the label. Then for each region, we have sample data in shape $(n - 14, 14, 3)$ where $n - 14$ is the batch size, 14 is the sequence length, and 3 is number of feature, and we also have label data in shape $(n - 14, 3)$.

1.2. Models

1.2.1. LINEAR REGRESSION

An straight forward approach is to use Linear Regression. With the data we've prepared, we need to reshape the samples into a single vector. Conceptually, such model is basically trying to solve "what will be the upcoming number of cases given the number of cases in the past 2 weeks". The objective function has Mean Square Error as loss function. We will run experiments with both regularization term on
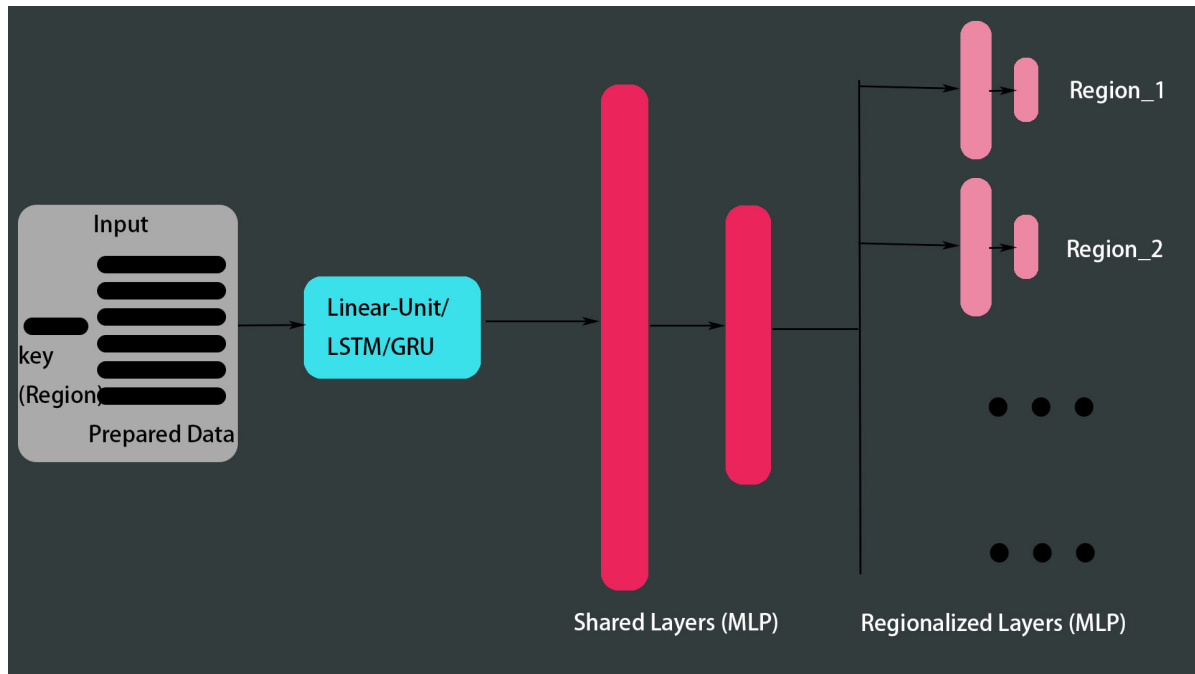
Figure 1. Multitask Network with Regionalized layers

L1 (Lasso Regression) and L2 (Ridge Regression) norms.

1.2.2. MULTI-LAYER-PERCEPTRON (MLP)

The most intuitive way to use Neural Networks to perform the forecasting task is to build a Multi-Layer-Perceptron with Linear units in each layers. Conceptually, each unit is performing the similar task as linear regression where they are trying to catch some information from the previous 2 weeks, and contribute to the final prediction with weights determined by the gradient descent algorithm.

1.2.3. RECURRENT NEURAL NETWORKS (RNNS)

Recurrent Neural Networks (RNNs) often indicate better performance in dealing with the time series data, because the information within the series are interpreted recurrently across the time steps. One of the popular RNNs that we will experimenting with is Long-Short-Term-Memory (LSTM), where we have forgot gate, input gate, and output gate within each cell. Another Network that we will experimenting with is the Gated Recurrent Unit (GRU) with reset gate and update gate in each cell.

1.2.4. MULTITASK NETWORKS

In addition, since we have data for multiple regions, and each region have different factors that influence the epidemic situation with different extent. For example, there are

differences in the population density, the advancement of the medical technology, and the ways of how governments and people from varies fields react with the virus. Also some regions already go over almost a complete epidemic period, while some other regions are at the beginning or the middle of one period. Thus, we tried to build "customized" (or "regionalized") models for each region with Multitask Learning.

To build such models, for each of the Neural Networks models mentioned above, including MLP, LSTM, and GRU, we append "regionalized" layers with linear units for each region. Among these models, we denote them as MM-NET, LM-NET, and GM-NET respectively. A brief representation is shown in Figure 1. Though training such models would cost a lot of time and resources, so the hyper-parameters set might not be the best.

2. Result

2.1. Short Term Forecasting

To evaluate the performance of the models in the short term prediction scenario, we use the data from 1/22/2019 to 5/13/2019 for training, and make a fair comparison between these models with their performance on the testing set which is the number of cases in the following week, with Mean Square Error (MSE) as evaluating metric. All the

Table 1. Short-Term Testing Results (MSE) – prediction for 7 days
Train: 1/22/2019 – 5/13/2019, Test: 5/14/2019 – 5/20/2019

MODELS	CONFIRMED	DEATH	RECOVERED
RIDGE REGRESSION	0.264	0.265	0.325
LASSO REGRESSION	0.264	0.265	0.325
MLP	0.250	0.264	0.299
LSTM	**0.240**	**0.259**	**0.297**
GRU	0.241	0.260	0.300
MM-NET	0.362	0.324	0.403
LM-NET	0.392	0.369	0.461
GM-NET	0.420	0.387	0.470

Table 2. Long-Term Testing Results (MSE) – prediction for a month
Train: 1/22/2019 – 5/13/2019, Test: 5/21/2019 – 6/10/2019

MODELS	CONFIRMED	DEATH	RECOVERED
RIDGE REGRESSION	1.028	0.653	0.732
LASSO REGRESSION	**1.025**	**0.652**	**0.731**
MLP	1.899	1.305	1.270
LSTM	1.140	0.976	1.069
GRU	1.410	1.071	1.453
MM-NET	2.943	1.552	2.115
LM-NET	1.440	1.179	1.270
GM-NET	1.540	1.227	1.327

training and testing data remain normalized, and they will only be mapping inversely when we make graphs for the visualization of the performances. The metric are calculated as

$$\frac{\sum_{i=0}^{n} ||y_i - y_i'||_2^2}{n}$$

where n is the number of regions, y_i and y_i' are the true values and predicted values respectively with each vector representing the number of new cases of confirmed, death, and recovered at its corresponding day.

From Table 1 we can see that in the scenario of short term prediction, LSTM has the best performance, and GRU has about the similar error. For the linear regression, the choices of regularization term almost have no effects on the final prediction. Moreover, among the Multitask Models, using Linear unit in the shared layers has the better performance than using LSTM and GRU unit in the shared layers. Therefore, it is obvious to see that Recurrent Neural Networks has the best ability to forecast the time series for the following week.

2.2. Long Term Forecasting

With the same comparison setting we've made for Short Term scenario, we have same training set and evaluat-

ing metric, but the test set is from 5/21/2019 to 6/10/2019.

From Table 2 we can see that in the scenario of short term prediction, we can see that Linear regression with regularization on L1 norm has the best performance. And Regression with regularization on L2 norm has the close performance. Among the Neural Networks, LSTM still shows better performance. One thing to notice is that the Multitask Networks with LSTM in the shared layers is better than using other unit in this long term prediction scenario.

3. Discussion

In order to visualize the performance of different models, we choose results from Lasso Regression, LSTM, and LSTM Multitask Network for comparison. They are the best Linear regression model, Recurrent Neural Network, and Multitask Network we have from the experiments. We then pick some regions, and they are anonymous while graphing. In Figure 2, 3, and 4 as shown below, where it is easy to see the difference among the models. Each data point on the time series represent the new increased cases. The blue line represent the true data; the green line represent the performance on training set; the red line represent the performance on the short term testing set; and the yellow line represent the performance on the long term testing set.

From the figures, there are some observations:
a) Recurrent Neural Network often have the stable prediction;
b) Multitask Network fit very well in the training set, and catch the periodic features in the series, which can be shown in its pattern of predicted series;
c) Linear Regression method rely a little bit heavily on the previous 14 points in the series. If the previous series have the increasing trend, the prediction will have increasing trend; similarly, if the previous series have decreasing trend, the prediction will have decreasing trend.

In practice, although linear regression has lower error in the experiments, it is still reasonable to use Recurrent Neural Network for forecasting. Because Linear Regression could not integrate the information from different part of the time series as appropriate as Recurrent Neural Network. For example, it is highly likely for the regression model to treat the ending of a epidemic period as the beginning of the period.

For the Multitask Networks, it has great potential in the longer term prediction scenario. It could not only catch the useful information within the pattern of the series, but also offering a "customize" prediction for each region. As

a result, the final prediction of regions will not only carry the information from collective features, but also contain information from their own features.

4. Conclusion and Future Works

From the experiments, we explored the performances of different models on the forecasting task. LSTM has the best performance in the short term prediction scenario, and Lasso Regression has the best performance in the long term prediction scenario. From the visualized graph, we can see that LSTM has the stable forecasting results, and Multitask Networks has great potentials to provide a generally concrete forecasting results. For now, we only use the raw data of the number of confirmed cases, death cases, and recovered cases. We would like to try training the model with some covariates, such as the weathers, GDP, and any factors that could potentially influence the tendency of the epidemic situation. For more exploration on the Multitask Networks and covariates, we leave them for the future works.
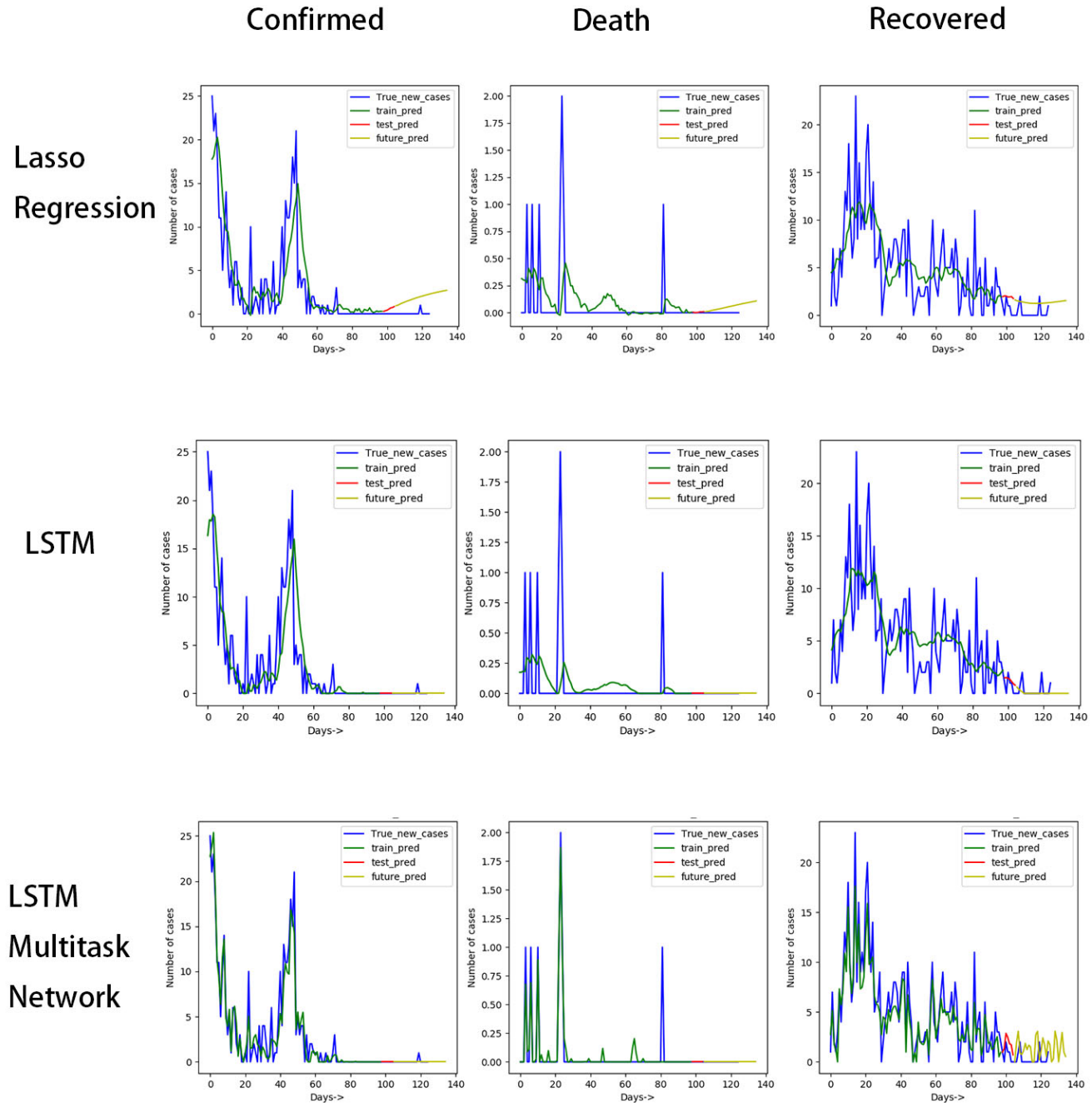
Confirmed   Death   Recovered

Lasso Regression

LSTM

LSTM Multitask Network

Figure 2. Performance of models in Region 1

Confirmed | Death | Recovered

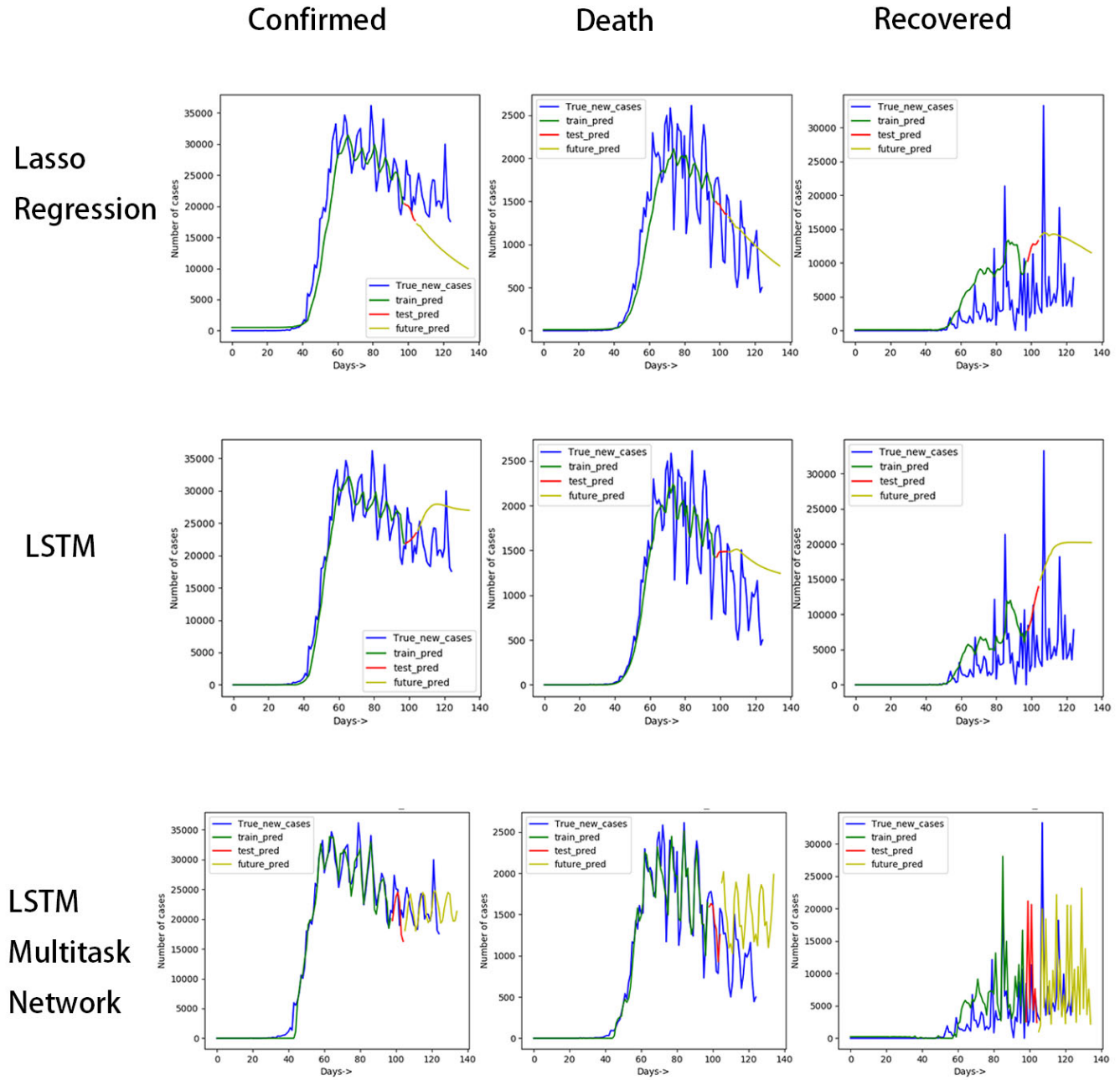Lasso Regression

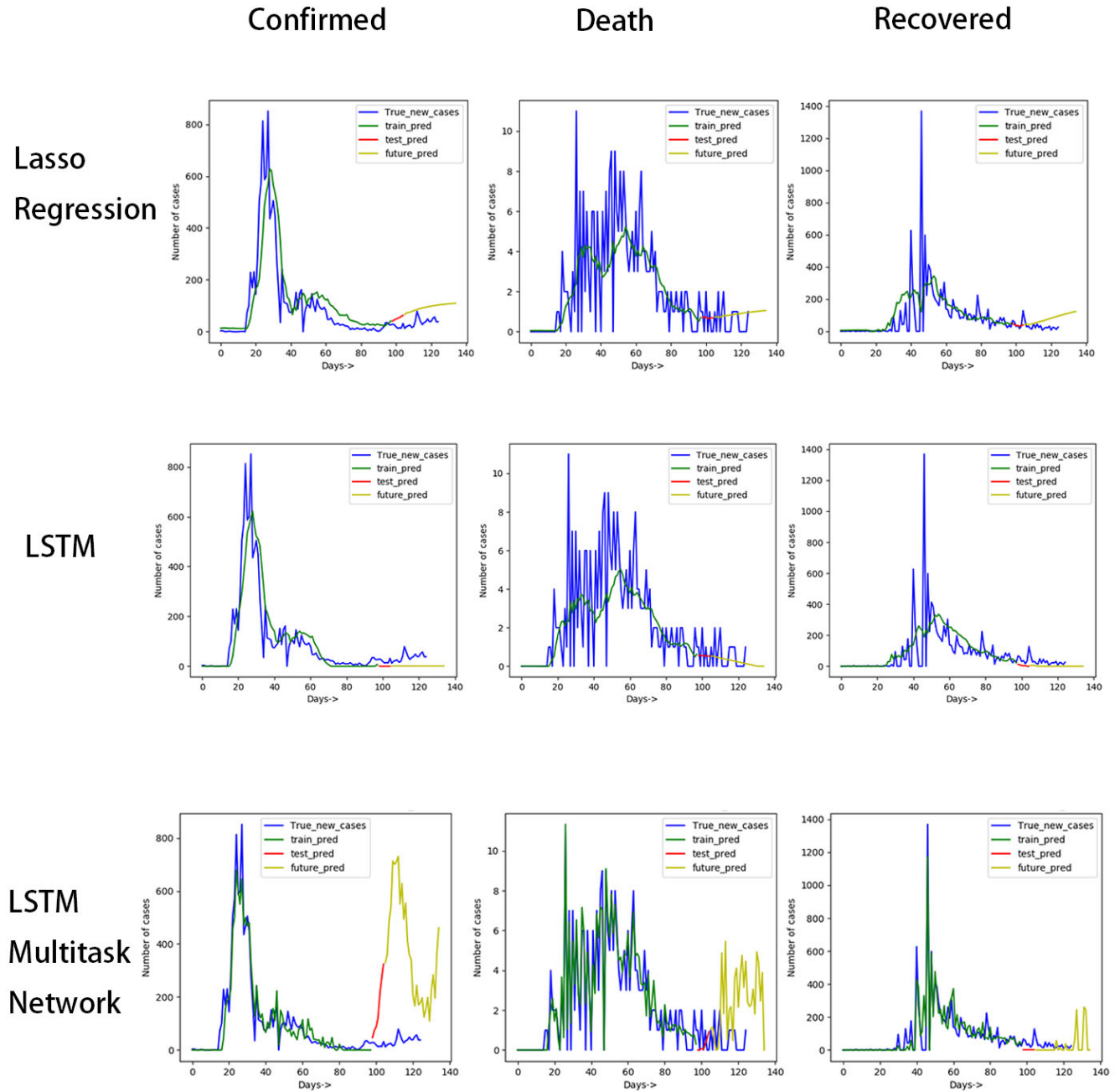LSTM

LSTM Multitask Network

Figure 3. Performance of models in Region 2

Figure 4. Performance of models in Region 3