

A Recurrent Neural Network and Differential Equation Based Spatiotemporal Infectious Disease Model with Application to COVID-19

Zhijian Li¹, Yunling Zheng¹, Jack Xin¹, and Guofa Zhou²

¹Department of Mathematics, UC Irvine

²College of Health Sciences, UC Irvine, CA 92697, USA.

Abstract

The outbreaks of Coronavirus Disease 2019 (COVID-19) have impacted the world significantly. Modeling the trend of infection and real-time forecasting of cases can help decision making and control of the disease spread. However, data-driven methods such as recurrent neural networks (RNN) can perform poorly due to limited daily samples in time. In this work, we develop an integrated spatiotemporal model based on the epidemic differential equations (SIR) and RNN. The former after simplification and discretization is a compact model of temporal infection trend of a region while the latter models the effect of nearest neighboring regions. The latter captures latent spatial information. We trained and tested our model on COVID-19 data in Italy, and show that it out-performs existing temporal models (fully connected NN, SIR, ARIMA) in 1-day, 3-day, and 1-week ahead forecasting especially in the regime of limited training data.

Keywords: COVID-19, Recurrent Neural Network, Discrete Epidemic Model, Spatiotemporal Machine Learning.

1 Introduction

Susceptible-Infected-Removed (SIR) is a classical differential equation model of infectious diseases [2]. It divides the total population into three compartments and models their evolution by the system of equations

$$\begin{aligned}\frac{dS}{dt} &= -\beta I S \\ \frac{dI}{dt} &= \beta I S - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

where β and γ are two positive parameters. SIR is a simple and efficient model of temporal data for a given region, see also [3] for related compartment models with social structures.

Yet the infectious disease data are often spatio-temporal as in the case of COVID-19, see [5]. A natural question is how to extend SIR to a space time model of suitable complexity so that it can be quickly trained from the available public data sets and applied in real-time forecasts. See [8] for temporal model real-time forecasts on cumulative cases of China in Feb 2020.

In this paper, we explore spatial infectious disease information to model the latent effect due to the in-flow of the infected people from the geographical neighbors. The in-flow data is not observed. To this end, machine learning tools such as regression and neural network models are more convenient. Auto-regressive model (AR) and its variants are linear statistical models to forecast time-series data. The Long Short Term Memory (LSTM) neural networks, originally designed for natural language processing [4], have more representation power and can be applied to disease time-series data as well. With spatial structures added, the graph-structured LSTM models can achieve state-of-the-art performance on spatiotemporal influenza data [6], crime and traffic data [10, 9]. However, they require a large enough supply of training data. For COVID-19, we only have limited daily data since the outbreaks began in early 2020. Applying space-time LSTM models [6, 9] directly to COVID-19 turns out to produce poor results. In view of the limited COVID-19 data, we shall propose a hybrid SIR-LSTM model.

2 Related Work

In [11], the authors designed a variant of AR, the AutoRegression with Google search data (ARGO), that utilizes external feature of google search data to forecast influenza data from Centers for Disease

Control and Prevention (CDC). Based on google search trend data that correlated to influenza as external feature, ARGO is a linear model that processes historical observations and external features. The prediction of influenza activity level at time t , defined as \hat{y}_t , is given by:

$$\hat{y}_t = u_t + \sum_{j=1}^{52} \alpha_j y_{t-j} + \sum_{i=1}^{100} \beta_i X_{i,t}.$$

ARGO is optimized as:

$$\begin{aligned} \min_{\mu_y, \vec{\alpha}, \vec{\beta}} & \left(y_t - u_t - \sum_{j=1}^{52} \alpha_j y_{t-j} - \sum_{i=1}^{100} \beta_i X_{i,t} \right)^2 \\ & + \lambda_a \|\vec{\alpha}\|_1 + \eta_a \|\vec{\beta}\|_1 + \lambda_b \|\vec{\alpha}\|_2^2 + \eta_b \|\vec{\beta}\|_2^2 \end{aligned}$$

where $\vec{\alpha} = (\alpha_1, \dots, \alpha_{52})$ and $\vec{\beta} = (\beta_1, \dots, \beta_{100})$. The y_{t-j} 's, $1 \leq j \leq 52$, are historical observations of previous 52 weeks and $X_{i,t}$ are the google search trend measures of top 100 terms that are most correlated to influenza at time t . Essentially, ARGO is a linear regression with regularization terms. In [11], ARGO is shown to outperform standard machine learning models such as LSTM, AR, and ARIMA.

In [6], graph structured recurrent neural network (GSRNN) further improved ARGO in the forecasting accuracy of CDC influenza activity level. The CDC partitions the US into 10 Health and Human Services (HHS) regions for reporting. GSRNN treats the 10 regions as a graph with nodes v_1, \dots, v_{10} , and E be the collection of edges (i.e $E = \{(v_i, v_j) | v_i, v_j \text{ are adjacent}\}$). Based on the average history of activity levels, the 10 HHS regions are divided into two groups, relatively active group, \mathcal{H} , and relatively inactive group, \mathcal{L} . There are 3 types of edges, $\mathcal{L} - \mathcal{L}$, $\mathcal{H} - \mathcal{L}$, and $\mathcal{H} - \mathcal{H}$, and each edge type has a corresponding RNN to train the edge features. There are also two node RNNs for each group to output the final prediction. Given a node (region) v , suppose $v \in \mathcal{H}$. GSRNN generates the edge features of v at time t , $e_{v,\mathcal{H}}^t$ and $e_{v,\mathcal{L}}^t$, by averaging the history of neighbors of v in the corresponding groups. Next, the edge features are fed into the corresponding edge RNNs:

$$f_v^t = \text{edgeRNN}_{\mathcal{H}-\mathcal{L}}(e_{v,\mathcal{L}}^t), \quad h_v^t = \text{edgeRNN}_{\mathcal{H}-\mathcal{H}}(e_{v,\mathcal{H}}^t)$$

Then, the outputs of edgeRNNs are fed into the nodeRNN of group \mathcal{H} together with the node feature of v at time t , denoted as v^t , to output the prediction of the activity level of node v at time $t+1$, or y_v^{t+1} :

$$y_v^{t+1} = \text{nodeRNN}_{\mathcal{H}}(v^t, f_v^t, h_v^t).$$

3 Our Contribution: IeRNN model

We propose a novel spatiotemporal model integrating LSTM [4] with a discrete time I-equation derived from SIR differential equations. The LSTM is utilized to model latent spatial information. The I-equation models the observed temporal information. Our model, named IeRNN, differs from [6, 10, 9] in that a difference equation with 3 parameters (the I-equation) fits the limited temporal data, which is far more compact than LSTM.

3.1 Derivation of Discrete-Time I-Equation

Based on SIR model, we add an additional feature I_e that represents the external infection influence from the neighbors of a region. Then the SIR nonlinear system with I_e as external forcing becomes

$$\frac{dS}{dt} = -\beta_1 S I - \beta_2 S I_e \quad (1)$$

$$\frac{dI}{dt} = \beta_1 S I + \beta_2 S I_e - \gamma I \quad (2)$$

$$\frac{dR}{dt} = \gamma I \quad (3)$$

which conserves the total mass (normalized to 1): $S + I + R = 1$. It follows from (3) that

$$R(t) = R(t_0) + \int_{t_0}^t \gamma I d\tau$$

Hence,

$$S(t) = 1 - I(t) - R(t_0) - \gamma \int_{t_0}^t I d\tau$$

Substituting $S(t)$ into (2) we have:

$$\frac{dI}{dt} = (\beta_1 I + \beta_2 I_e) \left(1 - I(t) - R(t_0) - \gamma \int_{t_0}^t I(\tau) d\tau \right) - \gamma I$$

Combining forward Euler method and Riemann sum approximation of the integral, we have a discrete approximation:

$$I(t+1) = (1 - \gamma)I(t) + (\beta_1 I(t) + \beta_2 I_e(t)) \cdot \left(1 - I(t) - R(t_0) - \gamma \frac{t - t_0}{p + 1} \sum_{j=0}^p I(t-j) \right)$$

As we model $I(t)$ from the beginning of the infection, we have $t_0 = 0$ and $R(t_0) = 0$. We arrive at the following discrete time I-equation:

$$I(t+1) = (1-\gamma)I(t) + (\beta_1 I(t) + \beta_2 I_e(t)) \cdot \left(1 - I(t) - \gamma \frac{t}{p+1} \sum_{j=0}^p I(t-j)\right) \quad (4)$$

Note that if we let $I_e(t) = 0$, then we have an approximation of $I(t)$ for the original SIR model, which is a solely temporal model (named I model):

$$I(t+1) = (1 - \gamma - \beta)I(t) - \beta I^2(t) - \beta \gamma \frac{t}{p+1} I(t) \sum_{j=0}^p I(t-j) \quad (5)$$

In reality, it is hard to know how a population of a region interacts with populations of neighboring regions. As a result, $I_e(t)$ is a latent information that is difficult to model by a mathematical formula or equation. In order to retrieve latent spatial information, we employ recurrent neural networks made of LSTM cells [4], see Fig. 2.

3.2 Generating Edge Feature and Computing Latent I_e

We utilize the spatial information basing on the Italy region map, Fig. 1. In order to learn the latent information I_e of a region v , we first generate the edge feature of v . Let C be the collection of neighbors of v . Then, the edge feature of v at time t is formulated as:

$$f_e^t = \frac{1}{|C|} \sum_{i: v_i \in C} [I_i(t-1), \dots, I_i(t-p)]$$

where $I_i(t)$ is the infection population percentage of region v_i at time t . Then, we feed f_e^t into an Edge RNN, an RNN with 3 stacked LSTM cells (see Fig. 3), followed by a dense layer for computing I_e . The activation function of the dense layer is hyperbolic tangent function. Figure 4 illustrates the procedure of computing I_e for Lazio as an illustration. We hence call our model *IeRNN* due to its integrated design of I-equation and edge RNN.

4 Experiment

To calibrate our model IeRNN, we use the Italy COVID-19 data [5] for training and testing. Although the US has the most infected cases, the recovered cases are largely missing. On the other hand, the Italian COVID-19 data is more accurately reported and better



Figure 1: Italian Region Map

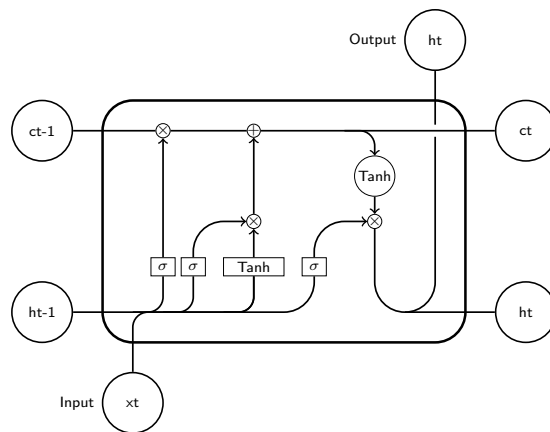


Figure 2: LSTM cell: input x_t , output: h_t ; σ is a sigmoid function.

maintained, reflecting a nearly complete duration of the rise and fall of infection. We collect the data of daily new (current) cases from 2020-02-24 to 2020-06-18 of 20 Italian regions. We set $p = 3$ in (4) based on experimental performance. As a result, we have the

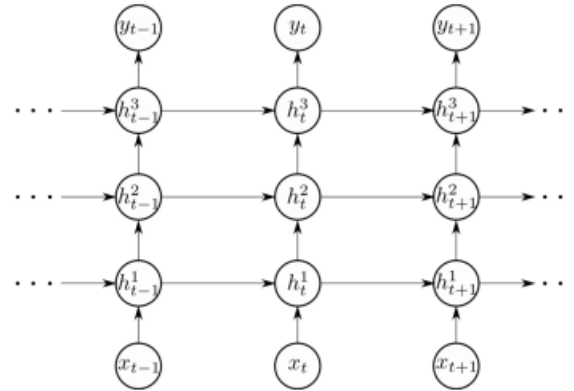


Figure 3: Edge RNN consisting of 3 stacked LSTM cells.

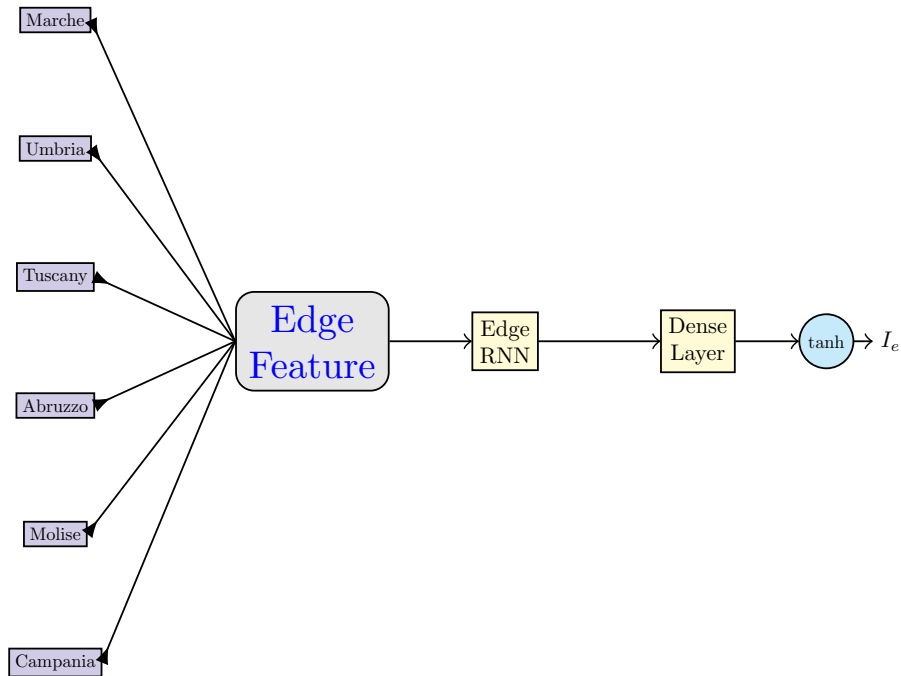


Figure 4: Computing I_e of Lazio region. Edge RNN is as shown in Fig. 3. Dense layer is fully connected (see Fig. 5).

current data for 113 days, with 81 days to train our model and 32 days to test our model (or 70%/30% training/testing data split).

Our training loss function is the mean squared error of the model output and training data:

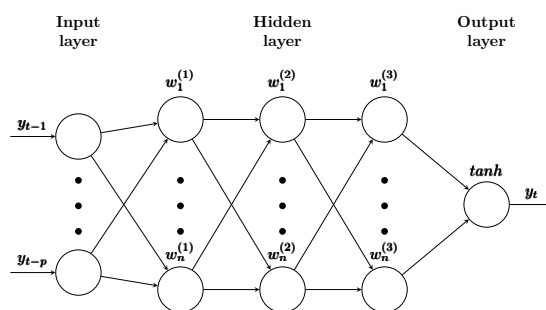
$$\hat{y}(t) = (1 - \gamma)y(t - 1) + (\beta_1 y(t - 1) + \beta_2 I_e(t - 1)) \cdot \left(1 - y(t - 1) - \gamma \frac{t}{p + 1} \sum_{j=1}^{p+1} y(t - j)\right)$$

$$Loss = \frac{1}{T - p - 1} \sum_{t=p+1}^T (y(t) - \hat{y}(t))^2$$

Since the training is minimization of the above loss function over parameters in both I-equation and RNN, the two components of IeRNN are coupled while learning from data. We use Adam gradient descent to learn the weights of LSTM and the dense layer, as well as I-equation parameters β_1 , β_2 , and γ .

To evaluate the performance of our model, we compare IeRNN, I-model (5), a fully-connected neural network (fcNN, Fig. 5) with hyperbolic tangent activation function, and auto-regression model (ARIMA). As the standard setting of ARIMA is 1-day ahead prediction, we shall only compare with it in such a very short-term case. Since infectious disease evolution is intrinsically nonlinear, we shall compare nonlinear models for 3-day and 1-week ahead forecasting. Based on experimental performance, we set the number of hidden units to be 100, 150, and 100 for the three layers of fcNN respectively.

Figure 5: Schematic of fcNN for modeling time series.



4.1 One-Day Ahead Forecast

As we see in Fig. 6, fcNN can perform poorly. This is not a surprise, as both [6] and [11] relied on hundreds of historical observations to train their models. The I-model based on only sequential data in time of one region merely follows the trend of the true data but cannot provide accurate predictions. Our IeRNN model, with the help of additional spatial information, is able to make accurate predictions and outperform other models. We also test the IeRNN with training data reduced to 40% (46 days). IeRNN is still able to track the general trend of the infected population percentage.

We measure the test accuracy with the Root Mean Square Error (RMSE) averaged over a few trials in training. In Tables 1 and 2 on 1-day ahead forecast, IeRNN achieves the smallest RMSE errors, and I-model has the largest errors. The compact I-model with 2 parameters cannot do 1-day ahead prediction as accurately. ARIMA outperforms I-model and does better on Emilia-Romagna and Lazio regions than fcNN. ARIMA, a linear model, has simpler structure than fcNN whose nonlinearity does not play out in such a short time task. Fig. 8 shows 1-day ahead forecast of IeRNN model on other regions with the learned latent external forcing I_e in Fig. 9.

Table 1: RMSE test errors in 1-day ahead forecast trained with 70 % of data. E-R= Emilia-Romagna.

| Model | Lombardy | E-R | Lazio |
|---------|-----------|-----------|-----------|
| IeRNN | 1.027e-04 | 6.333e-05 | 3.251e-05 |
| I-model | 1.175e-03 | 3.284e-04 | 2.439e-04 |
| fcNN | 1.580e-04 | 4.614e-04 | 2.294e-04 |
| ARIMA | 9.789e-04 | 3.627e-04 | 4.365e-05 |

Table 2: RMSE test errors in 1-day ahead forecast trained with reduced (40 % of) data. E-R= Emilia-Romagna.

| Model | Lombardy | E-R | Lazio |
|---------|-----------|------------|-----------|
| IeRNN | 9.850e-05 | 1.778e-04 | 3.617e-05 |
| I-model | 1.871e-03 | 1.252 e-03 | 5.443e-04 |
| fcNN | 3.364e-04 | 6.204e-04 | 8.030e-04 |
| ARIMA | 1.277e-03 | 1.082e-03 | 4.018e-05 |

Table 3: RMSE test errors in 7-day ahead forecast trained with 70 % of data. E-R= Emilia-Romagna.

| Model | Lombardy | E-R | Lazio |
|---------|-----------|-----------|-----------|
| IeRNN | 3.513e-04 | 4.423e-04 | 1.161e-04 |
| I-model | 2.004e-03 | 6.627e-04 | 5.586e-04 |
| fcNN | 6.608e-04 | 4.804e-04 | 4.508e-04 |

Table 4: RMSE test errors in 7-day ahead forecast trained with reduced (40 % of) data. E-R= Emilia-Romagna.

| Model | Lombardy | E-R | Lazio |
|---------|-----------|-----------|-----------|
| IeRNN | 3.061e-04 | 4.324e-04 | 7.754e-05 |
| I-model | 2.196e-03 | 1.167e-03 | 6.011e-04 |
| fcNN | 2.224e-03 | 6.889e-04 | 1.851e-04 |

Table 5: RMSE test errors in 3-day ahead forecast trained with 70 % of data. E-R= Emilia-Romagna.

| Model | Lombardy | E-R | Lazio |
|---------|-----------|-----------|-----------|
| IeRNN | 2.479e-04 | 3.668e-04 | 5.979e-05 |
| I-model | 5.609e-04 | 1.724e-04 | 1.383e-04 |
| fcNN | 8.165e-04 | 6.757e-04 | 1.689e-04 |

Table 6: RMSE test errors in 3-day ahead forecast trained with reduced (40 % of) data. E-R= Emilia-Romagna.

| Model | Lombardy | E-R | Lazio |
|---------|-----------|-----------|-----------|
| IeRNN | 1.987e-04 | 3.256e-04 | 5.297e-05 |
| I-model | 1.114e-03 | 7.337e-04 | 3.507e-04 |
| fcNN | 8.611e-04 | 1.374e-03 | 5.290e-04 |

Table 7: Average model training (tr) and inference (inf) times in seconds on Macbook Pro with Intel i5 CPU. The first two columns are for 70 % training (tr70) data and the last two columns are for 40 % training (tr40) data.

| Model | tr70 | inf70 | tr40 | inf40 |
|---------|-------|--------|-------|--------|
| IeRNN | 0.58s | 0.018s | 0.51s | 0.02s |
| I-model | 0.14s | 0.004s | 0.11s | 0.004s |
| fcNN | 0.09s | 0.003s | 0.09s | 0.003s |
| ARIMA | 0.23s | 0.014s | 0.19s | 0.015s |

4.2 Multi-Day Ahead Forecast

In model training for multi-day ahead forecast, the training loss function is modified so that the model input comes from multiple days in the past. In 7-day ahead forecast, IeRNN leads the other two nonlinear models especially in the 40% training data case, by as much as a factor of 7 in Lombardy. In the 3-day ahead forecast, IeRNN leads fcNN by a factor of 4 in the 40% training data case, as much as a factor of 10 in Lazio. Figs. 10-13 show model comparison in training and forecast phases for Lombardy and Lazio.

4.3 Model Size and Computing Time

IeRNN (fcNN) has about 16400 (1800) parameters. The optimized $(\beta_1, \beta_2, \gamma) = (0.685, 0.158, 0.044)$ in Lombardy, similarly in other regions. Table 7 lists average model training and inference times.

5 Conclusions and Future Work

We developed a novel spatiotemporal infectious disease model consisting of a discrete epidemic equation for the region of interest and RNNs for interactions with nearest geographic regions. Our model can be trained under 1 second. Its inference takes a fraction of a second, suitable for real-time applications. Our model out-performs temporal models in one-day and multi-day ahead forecasts in limited training data regime. In future work, we shall consider social and control mechanisms [1, 7] to strengthen the I-equation, as well as traffic data to expand interaction beyond nearest neighbors.

6 Acknowledgements

The work was partially supported by NSF grants IIS-1632935, and DMS-1924548. JX would like to thank Prof. Fred Wan for helpful communications on disease modeling.

References

- [1] G. Albi, L. Pareschi, and M. Zanella. Control with uncertain data of socially structured compartmental epidemic models. *arXiv preprint arXiv*, 2004.13067v1:1–26, 2020.
- [2] R. Anderson and R. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, Oxford, 1992.
- [3] H. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42:599 – 653, 2000.
- [4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [5] Italian region. *Italian COVID-19 Data (accessed March-July, 2020)*. <https://github.com/Akaza994/COVID-19-Data/blob/master/covid-19/italy.csv>.
- [6] Z. Li, X. Luo, B. Wang, A. Bertozzi, and J. Xin. A study on graph-structured recurrent neural networks and sparsification with application to epidemic forecasting. In *World Congress on Global Optimization*, pages 730–739. Springer, 2019, https://doi.org/10.1007/978-3-030-21803-4_80.
- [7] D. Morris, F. Rossine, J. Plotkin, and S. Levin. Optimal, near-optimal, and robust epidemic control. *arXiv preprint arXiv*, 2004.02209v2:1–28, 2020.
- [8] K. Roosa, Y. Lee, R. Luo, A. Kirpich, R. Rothenberg, J.M. Hyman, P. Yan, and G. Chowell. Real-time forecasts of the COVID-19 epidemic in China from February 5th to February 24th, 2020. *Infectious Disease Modelling*, 5:256 – 263, 2020.
- [9] B. Wang, X. Luo, F. Zhang, B. Yuan, A. Bertozzi, and P. Brantingham. Graph-based deep modeling and real time forecasting of sparse spatio-temporal data. *arXiv preprint arXiv:1804.00684*, 2018.
- [10] B. Wang, P. Yin, A. Bertozzi, P. Brantingham, S. Osher, and J. Xin. Deep learning for real-time crime forecasting and its ternarization. *Chinese Annals of Mathematics, Series B*, 40(6):949–966, 2019.
- [11] S. Yang, M. Santillana, and S. Kou. Accurate estimation of influenza epidemics using Google search data via ARGO. *Proceedings of the National Academy of Sciences*, 112(47):14473–14478, 2015.

Figure 6: Training and 1-day ahead forecast of 4 models (IeRNN, fcNN, I-model, and ARIMA) in 4 rows respectively.

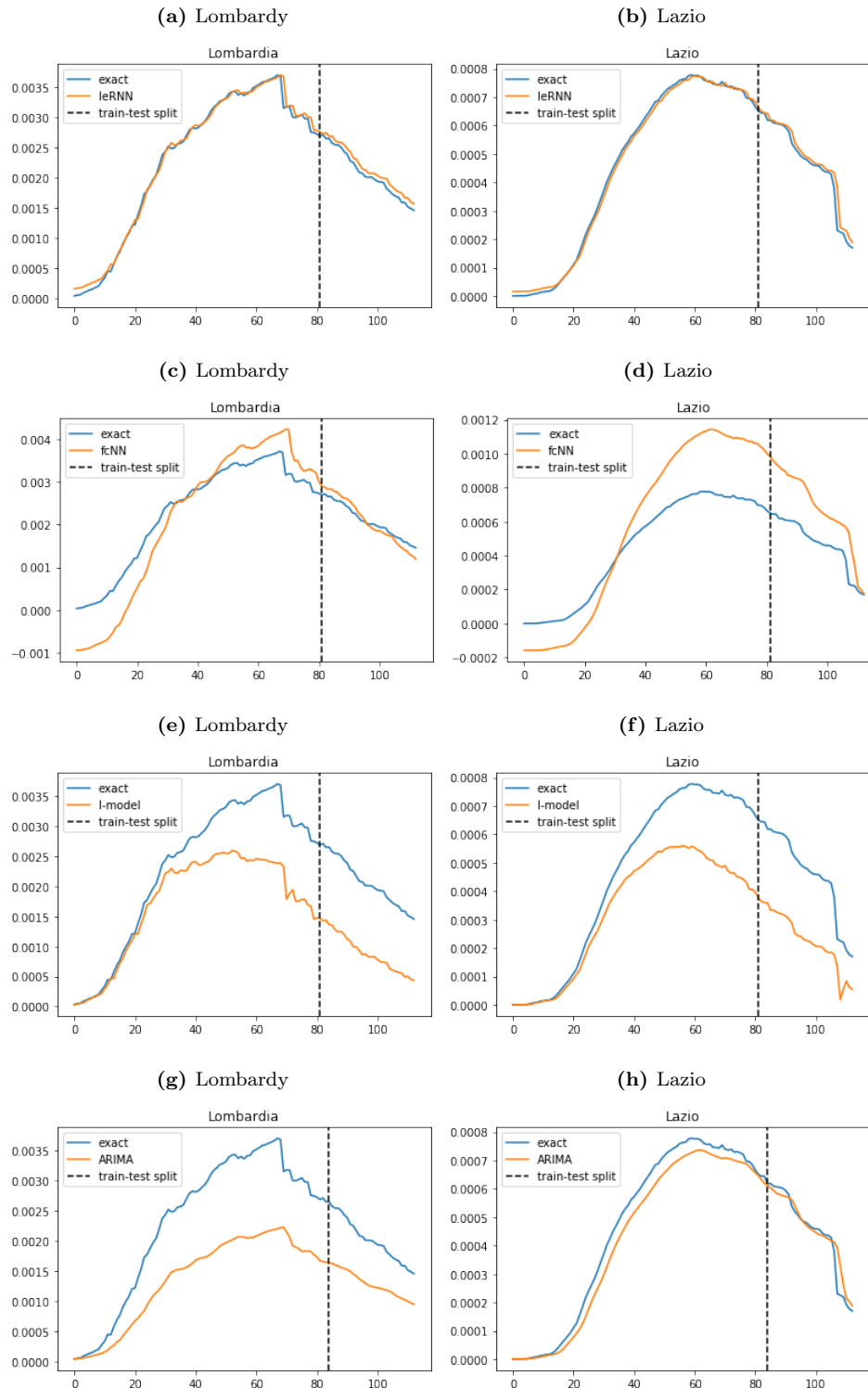


Figure 7: Training and 1-day ahead forecast of 4 models with reduced (40%) training data. The 4 rows are IeRNN, fcNN, I-model, and ARIMA respectively.

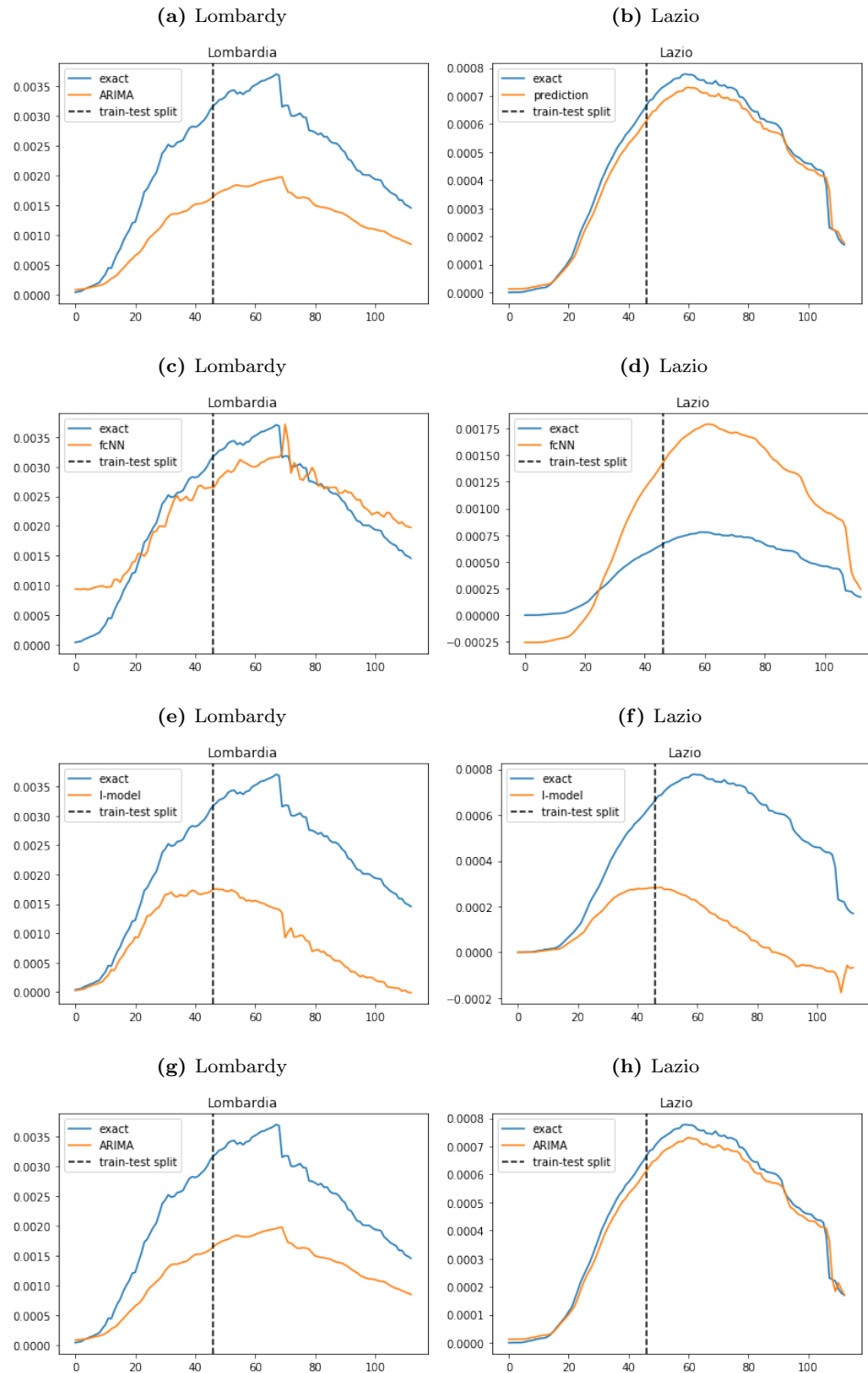


Figure 8: IeRNN training and 1-day ahead forecast on four additional regions.

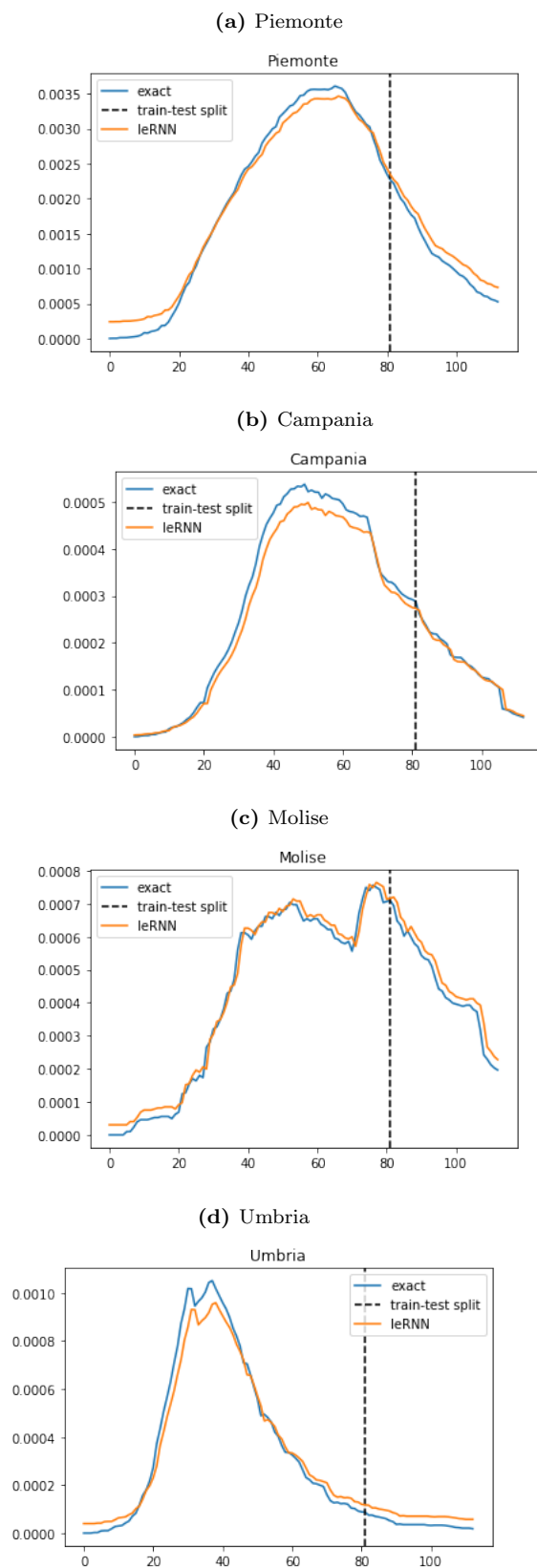


Figure 9: Visualization of the latent information I_e in Fig. 8 learned by IeRNN.



Figure 10: Training and 7-day ahead forecast of 3 models (IeRNN, fcNN, and I-model) in 3 rows respectively.

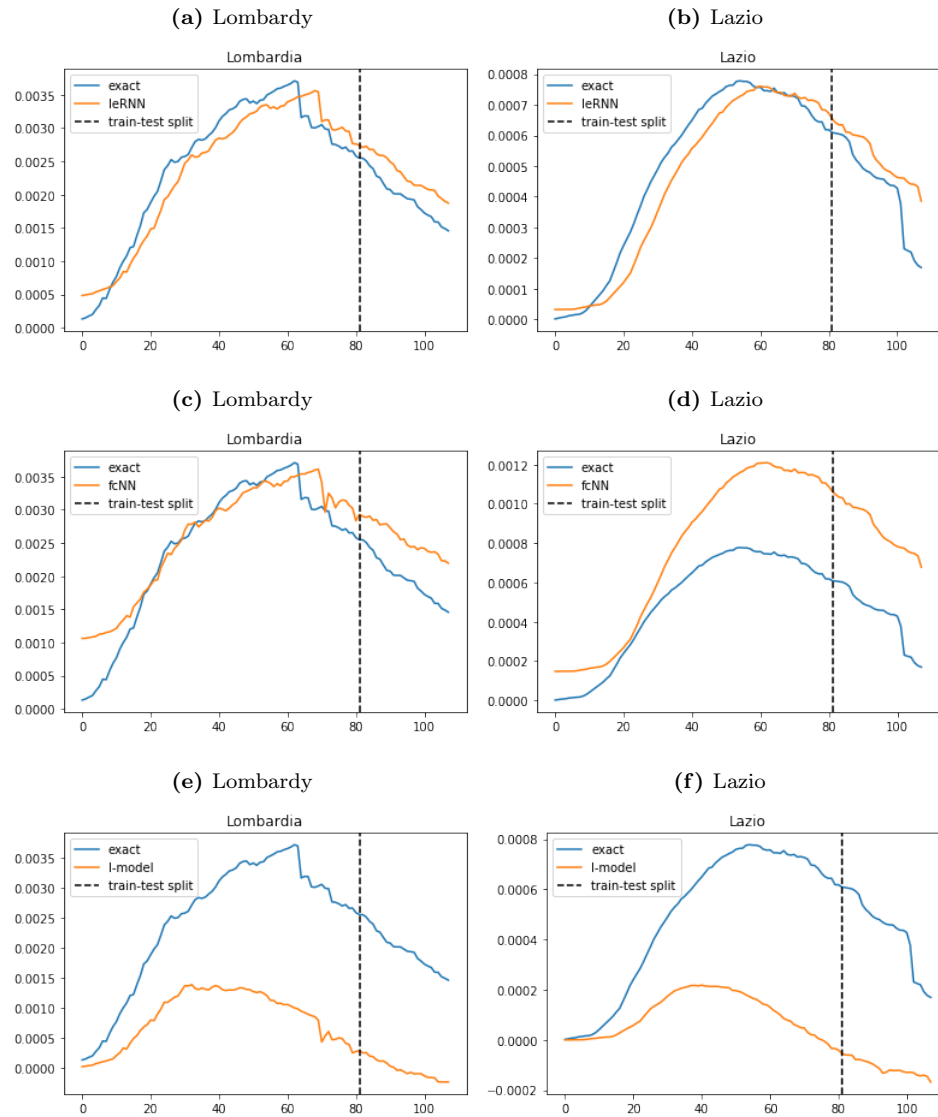


Figure 11: Training and 7-day ahead forecast of 3 models (IeRNN, fcNN, and I-model) with reduced (40%) training data in 3 rows respectively.

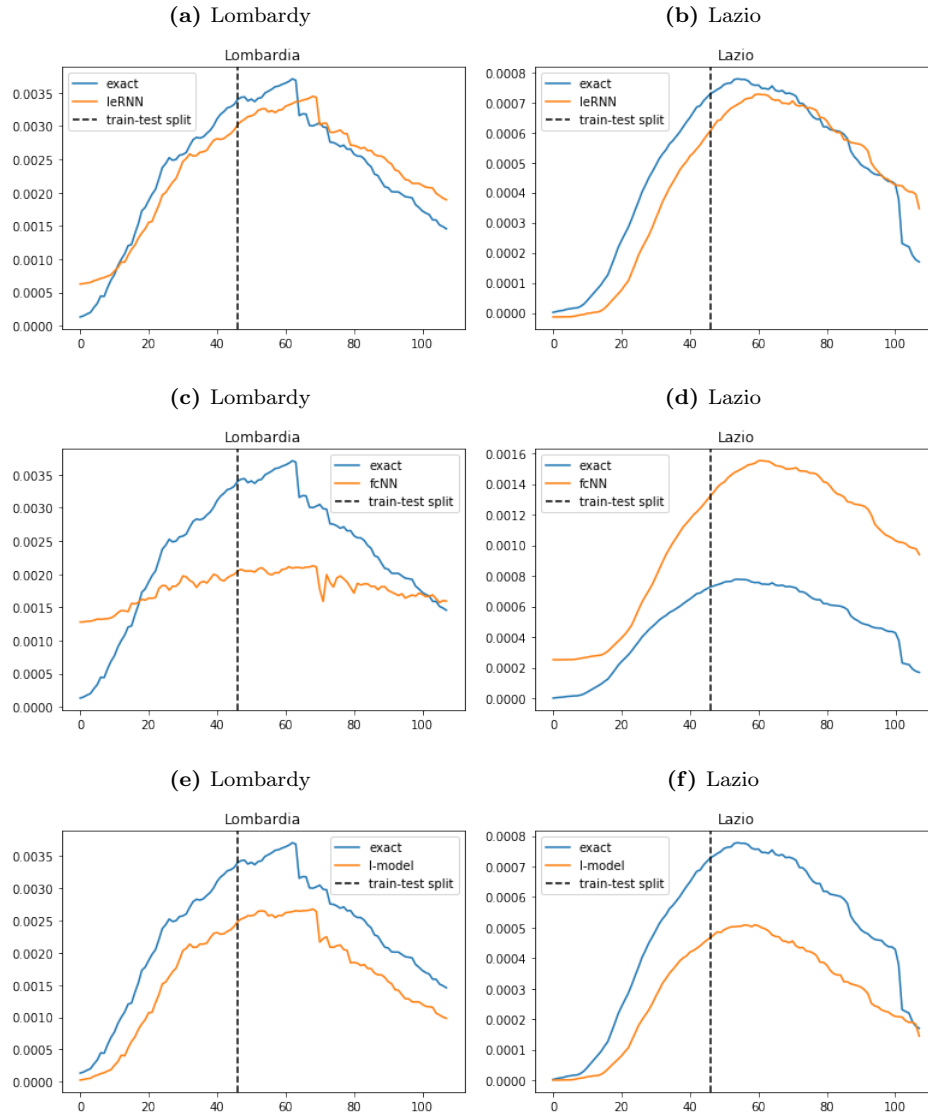


Figure 12: Training and 3-day ahead forecast of 3 models (IeRNN, fcNN, and I-model) in 3 rows respectively.

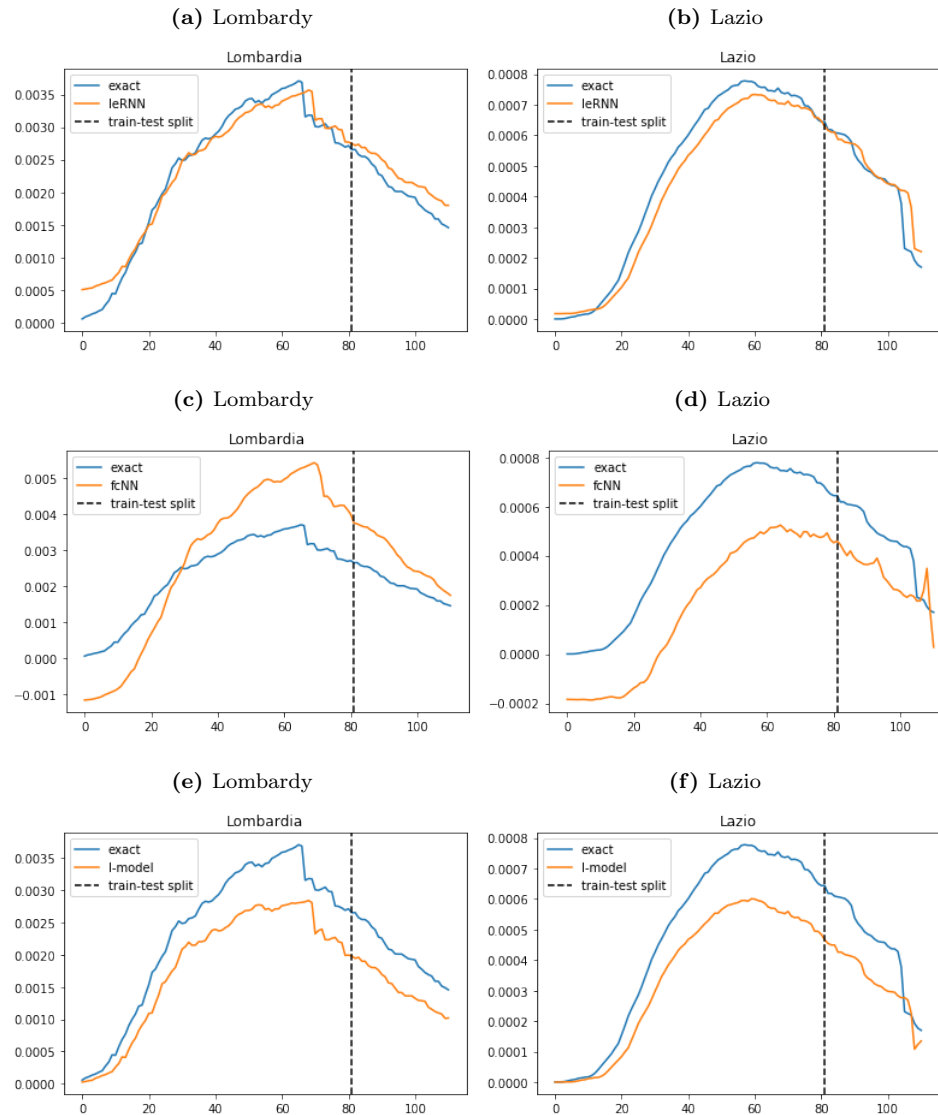


Figure 13: Training and 3-day ahead forecast of 3 models (IeRNN, fcNN, and I-model) with reduced (40 %) training data in 3 rows respectively.

