Transfer Graph Neural Networks for Pandemic Forecasting

George Panagopoulos[1], **Giannis Nikolentzos** [2], **Michalis Vazirgiannis** [1]
[1] École Polytechnique, Palaiseau, France
[2] Athens University of Economics and Business, Athens, Greece
george.panagopoulos@polytechnique.edu, {nikolentzos, mvazirg}@aueb.gr

Abstract

The recent outbreak of COVID-19 has affected millions of individuals around the world and has posed a significant challenge to global healthcare. From the early days of the pandemic, it became clear that it is highly contagious and that human mobility contributes significantly to its spread. In this paper, we utilize graph representation learning to capitalize on the underlying relationship of population movement with the spread of COVID-19. Specifically, we create a graph where the nodes correspond to a country's regions, the features include the region's history of COVID-19, and the edge weights denote human mobility from one region to another. Subsequently, we employ graph neural networks to predict the number of future cases, encoding the underlying diffusion patterns that govern the spread into our learning model. Furthermore, to account for the limited amount of training data, we capitalize on the pandemic's asynchronous outbreaks across countries and use a model-agnostic meta-learning based method to transfer knowledge from one country's model to another's. We compare the proposed approach against simple baselines and more traditional forecasting techniques in 4 European countries. Experimental results demonstrate the superiority of our method, highlighting the usefulness of GNNs in epidemiological prediction. Transfer learning provides the best model, highlighting its potential to improve the accuracy of the predictions in case of secondary waves, given data from past/parallel outbreaks.

1 Introduction

In late 2019, a highly infectious new virus, SARS-CoV-2, started spreading in Wuhan, China. In early 2020, the virus had spread to most countries around the world causing the pandemic of the COVID-19 disease. As of December 7, 2020, a total of 1,532,418 deaths and 66,422,058 cases of COVID-19 were confirmed worldwide[1]. During a pandemic, accurately predicting the spread of the infection is of paramount importance to governments and policymakers in order to impose measures to combat the spread of the virus or decide on the allocation of healthcare resources.

Given the severity of the pandemic and the need for accurate forecasting of the disease spread, machine learning, and artificial intelligence approaches have recently started to emerge as a promising methodology to combat COVID-19. However, it turns out that the nature of the problem is rather challenging (Zeroual et al. 2020). In many domains, data admits a natural graph representation. For instance, to predict the spread of COVID-19, ideally, we would like to have access to the social network of all individuals and to make predictions based on the interactions between them. However, in the absence of such data, we consider a similar problem: predicting the development of the disease based on mass mobility data, i. e., how many people moved from one place to another. Mobility inside a region can be regarded as a proxy of the interaction i.e. the more people move, the higher the risk of transmission inside the region. Interestingly, mobility between different regions is known to play a crucial role in the growth of the pandemic, especially for long range travels (Colizza et al. 2006; Soriano-Panos et al. 2020). Mobility gives rise to a natural graph representation allowing the application of recent relational learning techniques such as graph neural networks (GNNs).

GNNs have been applied to a wide variety of tasks, including node classification (Kipf and Welling 2017), graph classification (Morris et al. 2019) and text categorization (Nikolentzos, Tixier, and Vazirgiannis 2020). GNNs capitalize on the concept of message passing, that is, at every iteration, the representation of each vertex is updated based on messages received from its neighbors. Since GNNs have been successfully applied to several real-world problems, in this paper, we also investigate their effectiveness in forecasting COVID-19. We focus on the problem of predicting the number of confirmed COVID-19 cases in each node. We propose a model that captures both spatial and temporal information, thus combining mobility data with the history of COVID-19 cases.

To investigate if the model can learn the underlying complex dynamics associated with COVID-19, we evaluate it on recent data where we predict the number of new cases. Our results demonstrate that GNNs on mobility exhibit a substantial potential in predicting the disease spread. Furthermore, since the availability of data is limited at the start of the outbreak in a country, we employ a transfer learning method based on Model-Agnostic Meta-Learning (MAML) to capitalize on knowledge from other countries' models. Our main contributions are summarized as follows:

- We propose a model for learning the spreading of

[1] https://covid19.who.int

COVID-19 in a country's graph of regions. The model relies on the representational power of GNNs and their capability to encode the underpinnings of the epidemic.

- We apply a method based on MAML to transfer a disease spreading model from countries where the outbreak has been stabilized to another country where the disease is at its early stages.
- We evaluate the proposed approach on data obtained from regions of 4 different countries, namely France, Italy, Spain, and England. We observe that it can indeed surpass the benchmarks and produce useful predictions.

The rest of this paper is organized as follows. Section 2 provides an overview of the related work and elaborates our contribution. Section 3 provides a detailed description of the proposed model. Section 4 evaluates the proposed model on data from the first COVID-19 wave in 4 European countries. Finally, section 5 summarizes the work and presents potential future work.

2 Related Work

As mentioned above, many recent studies have leveraged machine learning and artificial intelligence techniques to make predictions about the spread of COVID-19. For instance, Lorch et al. (2020) propose a compartmental SEIR model which is based on a parameterized counting process. The parameteres of the model are estimated using bayesian optimization. and was evaluated on regions of Germany and Switzerland. Flaxman et al. (2020) study the effect of major non-pharmaceutical interventions across 11 European countries with a bayesian model whose parameters are estimated based on the observed deaths in those countries. Their results indicate that the interventions have had a large effect on reducing the spread of the disease. Time-series based models have also been utilized and will serve as our baselines. For instance, Chimmula and Zhang (2020) employed an LSTM to predict the number of confirmed COVID-19 cases in Canada, while Kufel (2020) investigated the effectiveness of the ARIMA model in predicting the dynamics of COVID-19 in certain European countries.

A GNN for epidemic forecasting, ColaGNN, was recently developed by Deng et al. (2019). ColaGNN learns a hidden state for each location using an RNN, and then an attention matrix is derived from these representations that captures how locations influence each other. This matrix forms the graph that is passed on to a GNN to generate the outputs. This work was evaluated on influenza-like illness (ILI) prediction in US and Japan, without the use of an underlying graph. More recent works on predicting COVID-19 using the graph of US counties include a static (Kapoor et al. 2020) and a temporal GNN (Gao et al. 2020). The former forms a supergraph using the instances of the mobility graph, where the spatial edges capture county-to-county movement at a specific date, and a county is connected to a number of past instances of itself with temporal edges. The node features include demographics, number of deaths and recoveries. In STAN (Gao et al. 2020), on the other hand, the edges are determined based on demographic similarity and geographical proximity between the counties. STAN takes advantage of the nature of the pandemic and predicts the parameters of an epidemic simulation model together with the infected and recovered cases, using multiple outputs in the neural network. These are used to produce long-term predictions based on the simulation and to penalize the original long-term predictions of the model. The main difference between these approaches and our work lies in the size of the constructed graph and the amount of available data for training. To be specific, in both these approaches, the size of the training data ranges from 50 to 60 days. In our case, this is not feasible as the pandemic is already at its peak before the 30th day, and has already cost too many lives. This is why we utilize transfer learning to account for the limited training samples in the initial stages of the pandemic. Moreover, our graphs are relatively small compared to the graph of US counties. Finally, our open data lacks in many cases the number of recovered cases, deaths and population demographics required for training these models. This kind of data may not always be available at the regional level for a real-life pandemic, especially for smaller, less developed countries.

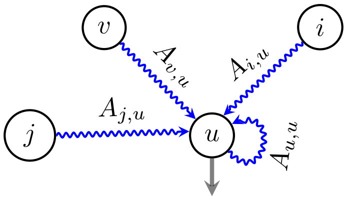
Transfer learning for disease prediction has been proposed in the past by Zou et al. (2019) who have mapped a disease model trained on online google searches obtained from one location, where the virus spreading is available, to another location, where the virus has not spread widely yet. More recently, this approach was utilized in the context of COVID-19 (Lampos et al. 2020). In the context of graph representation learning, transfer learning has only been used to the best of our knowledge for classifying textual documents represented as graphs (Lee et al. 2017), for traffic prediction (Mallick et al. 2020), for semi-supervised classification (Yao et al. 2020) and for designing GNNs that are robust to adversarial attacks (Tang et al. 2020).

3 Methodology

In this section, we present the proposed neural network architecture for predicting the course of the COVID-19 disease. It should be mentioned that our analysis involves a series of assumptions. First, we assume that people that use Facebook on their mobile phones with Location History enabled (see subsection 4.1 below) constitute a uniform random sample of the general population. Second, we assume that the number of cases in a region reported by the authorities is a representative sample of the number of people that have been actually infected by the virus. Finally, we hypothesize that the more people move from one region to another or within a region, the higher the probability that people in the receiving region are infected by the virus. This is a well-known observation in the field of epidemics(Colizza et al. 2006; Soriano-Panos et al. 2020), and motivates the use of a message passing procedure as we delineate below.

3.1 Graph Construction

We chose to represent each country as a graph $G = (V, E)$ where $n = |V|$ denotes the number of nodes. Specifically, given a country, we create a series of graphs, each corresponding to a specific date t, i.e., $G^{(1)}, \dots, G^{(T)}$. A single date's mobility data is transformed into a weighted, di-



$$\mathbf{z}_u = (A_{j,u}\mathbf{x}_j + A_{i,u}\mathbf{x}_i + A_{v,u}\mathbf{x}_v) + A_{u,u}\mathbf{x}_u$$

Figure 1: Example of the message passing procedure.

rected graph whose vertices represent the NUTS3 regions and edges capture the mobility patterns. For instance, the weight $A_{v,u}^{(t)}$ of the edge (v, u) from vertex v to vertex u denotes the total number of people that moved from region v to region u at time t. Note that these graphs can also contain self-loops which correspond to the mobility behavior within the regions. The mobility between administrative regions u and v at time t forms an edge which, multiplied by the number of cases $c_u^{(t)}$ of region u at time t, provides a relative score expressing how many infected individuals might have moved from u to v. To be more specific, let $\mathbf{x}_u^{(t)} = (c_u^{(t-d)}, \dots, c_u^{(t)})^\top \in \mathbb{R}^d$ be a vector of node attributes, which contains the number of cases for each one of the past d days in region u. We use the cases of multiple days instead of just the day before the prediction because case reporting is highly irregular between days, especially in decentralized regions. Intuitively, message passing over this network computes a feature vector for each region with a combined score from all regions, as illustrated below.

$$\mathbf{A}^{(t)}\,\mathbf{X}^{(t)} = \begin{bmatrix} A_{1,1}^{(t)} & A_{2,1}^{(t)} & \dots & A_{n,1}^{(t)} \\ A_{1,2}^{(t)} & A_{2,2}^{(t)} & \dots & A_{n,2}^{(t)} \\ \vdots & \vdots & \vdots & \vdots \\ A_{1,n}^{(t)} & A_{2,n}^{(t)} & \dots & A_{n,n}^{(t)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^{(t)} \\ \mathbf{x}_2^{(t)} \\ \vdots \\ \mathbf{x}_3^{(t)} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_3 \end{bmatrix}$$

where $\mathbf{A}^{(t)}$ is the adjacency matrix of $G^{(t)}$ and $\mathbf{X}^{(t)}$ is a matrix whose rows contain the attributes of the different regions. In this case, $\mathbf{z}_u \in \mathbb{R}^d$ is a vector that combines the mobility within and towards region u with the number of reported cases both in u and in all the other regions. Here, we would like to stress the importance of the mobility patterns $A_{u,u}$ within a region u which correspond to good indicators of the evolution of the disease, especially during lockdown periods. To visualize concretely how the representations are extracted from the message passing, Figure 1 contains a toy example with a region u receiving individuals from other regions. Intuitively, the average value $\bar{\mathbf{z}}_u$ represents an estimate of the number of new latent cases in u, and is broken down to the cases received from other regions (inside the parenthesis) and the new cases due to the mobility within region u (i. e., $A_{u,u}\mathbf{x}_u$).

3.2 Models

To model the dynamics of the spreading process, we use two instances of a well-known family of GNNs, known as message passing neural networks (MPNNs) (Gilmer et al. 2017). These networks consist of a series of neighborhood aggregation layers. Each layer uses the graph structure and the node feature vectors from the previous layer to generate new representations for the nodes.

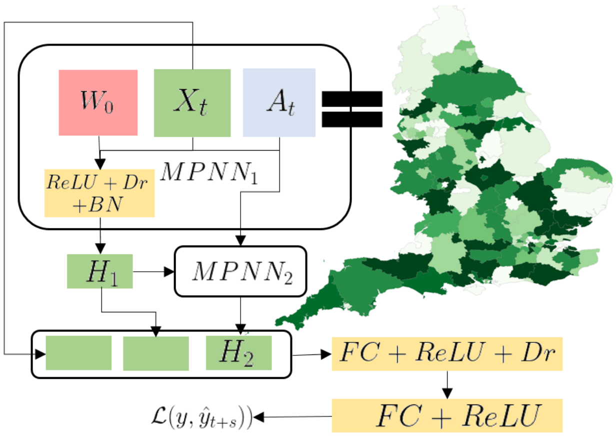


Figure 2: Overview of the MPNN architecture.

MPNN. To update the representations of the vertices of each of the input graphs, we use the following neighborhood aggregation scheme:

$$\mathbf{H}^{i+1} = f(\tilde{\mathbf{A}}\,\mathbf{H}^i\,\mathbf{W}^{i+1})$$

where $\mathbf{H}^i$ is a matrix that contains the node representations of the previous layer, with $\mathbf{H}^0 = \mathbf{X}$, $\mathbf{W}^i$ is the matrix of trainable parameters of layer i, and f is a non-linear activation function such as ReLU. Following Kipf and Welling (2017), we normalize the adjacency matrix $\mathbf{A}$ such that the sum of the weights of the incoming edges of each node is equal to 1. Note that for simplicity of notation, we have omitted the time index. The above model is in fact applied to all the input graphs $G^{(1)}, \dots, G^{(T)}$ separately. Given a model with K neighborhood aggregation layers, the matrices $\tilde{\mathbf{A}}$ and $\mathbf{H}^0, \dots, \mathbf{H}^K$ are specific to a single graph, while the weight matrices $\mathbf{W}^1, \dots, \mathbf{W}^K$ are shared across all graphs. Typically, an MPNN contains K neighborhood aggregation layers. As the number of neighborhood aggregation layers increases, the final node features capture more and more global information. However, retaining local, intermediary information might be useful as well. Thus, we concatenate the matrices $\mathbf{H}^0, \mathbf{H}^1, \mathbf{H}^2, \dots, \mathbf{H}^K$ horizontally, i. e., $\mathbf{H} = \text{CONCAT}(\mathbf{H}^0, \mathbf{H}^1, \mathbf{H}^2, \dots, \mathbf{H}^K)$, and the rows of the emerging matrix $\mathbf{H}$ can be regarded as vertex representations that encode multi-scale structural information, including the initial features of the node. In other words, we utilize skip connections from each layer to the output layer which consists of a sequence of fully-connected layers. Note that we apply the ReLU function to the output of the network since the number of new cases is a nonnegative integer. We choose the mean squared error as our loss function as shown below:

$$\mathcal{L} = \frac{1}{nT}\sum_{t=1}^{T}\sum_{v\in V}\left(y_v^{(t+1)} - \hat{y}_v^{(t+1)}\right)^2 \qquad (1)$$

where $y_v^{(t+1)}$ denotes the reported number of cases for region v at day $t+1$ and $\hat{y}_v^{(t+1)}$ denotes the predicted number of cases. An overview of the **MPNN** is given in Figure 2.

MPNN+LSTM. In order to take advantage of the temporal correlation between the target and the confirmed cases in the past, we build a time-series version of our model using the different snapshots of the mobility graph. Given a sequence of graphs $G^{(1)}, G^{(2)}, \dots, G^{(T)}$ that correspond to a sequence of dates, we utilize an MPNN at each time step, to obtain a sequence of representations $\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \dots, \mathbf{H}^{(T)}$. These representations are then fed into a Long-Short Term Memory network (LSTM) (Graves and Jaitly 2014) which can capture the long-range temporal dependencies in time series. We expect the hidden states of an LSTM to capture the spreading dynamics based on the mobility information encoded into the node representations. We use a stack of two LSTM layers. The new representations of the regions correspond to the hidden state of the last time step of the second LSTM layer. These representations are then passed on to an output layer similar to the **MPNN**, along with the initial features for each time step. Note that this model resembles other attempts to spatio-temporal prediction, but instead of convolutional (Yao et al. 2018), it employs message passing layers.

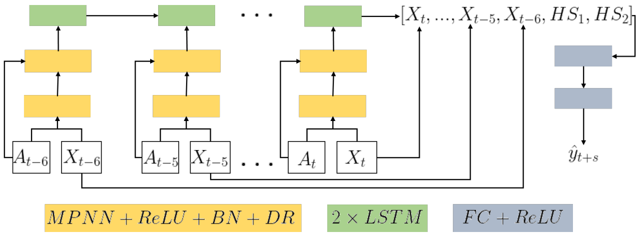


Figure 3: Overview of the MPNN+LSTM architecture.

MPNN+TL. Note that the different countries were hit by the pandemic at different times. Indeed, there are cases where once the epidemic starts developing in one country, it has already stabilized in another. Furthermore, a new wave of COVID-19 is very likely to share fundamental characteristics with the previous ones, as it is the same virus. This additional information may prove rather important in case of insufficient training data. In our setting, as discussed in subsection 4.2, the model starts predicting as early as the 15^{th} day of the dataset. In such a scenario, the model has access to a few samples to learn from (split into validation and training sets), while it is used to make predictions for as far as 14 days ahead. Given the inherent need for data in neural networks, this setting is rather challenging. Moreover, our intuition is that a model trained in the whole cycle or an advanced stage of the epidemic can capture patterns of its different phases, which is missing from a new model working in a country at the start of its infection.

To incorporate past knowledge from models running in other countries, we separate our data into tasks and propose an adaptation of MAML (Finn, Abbeel, and Levine 2017). Lets initially assume that the Meta Train set $M_{\text{tr}} = \{D^{(1)}, \dots, D^{(p)}\}$, corresponds to the datasets of p countries that we can use to obtain a set of parameters θ. The learnt parameters can then be employed to initialize the model for the country left out in the Meta Test M_{te}. In reality, each dataset $D^{(k)}, k \in \{1, \dots, p\}$ is divided into subtasks itself. More specifically, each country has different training sets of increasing size (as the train days increase) as well as shorter- and longer-term targets (next day, two days ahead and so on). For each combination of these two, we train a different model. Hence, the set of tasks for a country k is $D^{(k)} = \left\{\left(Tr_{i,j}^{(k)}, Te_{i,j}^{(k)}\right) : 14 < i \leq T, 1 \leq j \leq dt\right\}$ where $\left(Tr_{i,j}^{(k)}, Te_{i,j}^{(k)}\right)$ is a dataset (train and test set) associated with country k, where the train set consists of the first i days of the data and the task is to predict the number of cases in the j-th day ahead.

The set of parameters θ corresponds to the weight matrices and biases of all layers in the **MPNN** model. As mentioned above, in MAML, θ is randomly initialized and undergoes gradient descent steps during the metatrain phase. The algorithm is shown in Algorithm 1. In each task, we minimize the loss on the task's train set towards a task-specific θ_t, as shown in Equation (2) and on lines 3-5 of the algorithm. Then, we use the emerging θ_t to compute the gradient with respect to θ in the task's test set as illustrated in Equation (3) below and on line 6. This gradient is normalized by the total number of tasks in the set to refrain from taking too big steps.

$$\theta_t = \theta - \alpha\nabla_\theta\mathcal{L}\big(f_\theta(Tr_{i,j}^{(k)})\big) \qquad (2)$$

$$\theta = \theta - \alpha_m\nabla_\theta\mathcal{L}\big(f_{\theta_t}(Te_{i,j}^{(k)})\big) \qquad (3)$$

The standard update includes the gradient of θ_t and that of θ, which is in fact the hessian matrix, as shown in Equation (4).

$$\frac{\partial\mathcal{L}_T(f_{\theta_t}(Te_{i,j}^{(k)}))}{\partial\theta} = \nabla_{\theta_t}\mathcal{L}(f_{\theta_t}(Te_{i,j}^{(k)}))(I - a\nabla_\theta^2\mathcal{L}f_\theta(Tr_{i,j}^{(k)})) \qquad (4)$$

We are dropping the term that contains the hessian, as it was shown to have insignificant contribution in practice (Finn, Abbeel, and Levine 2017), possibly due to the vanishing gradient. Finally, we train θ on the train set of M_{te} and test on its test set (lines 7-10 and 11 respectively).

Note that in the Algorithm 1 , E is our error function (i. e., Equation (5)), $\mathcal{L}$ is the loss function defined in Equation (1), and f is an **MPNN**.

4 Experiments

In this section, we first describe the datasets that we used for our experiments. We next give details about the experimental settings. We last report on the performance of the proposed models and the baselines.

Algorithm 1 MPNN+TL

```
Input: $M_{\text{tr}}$, $M_{\text{te}}$, $\alpha$ , $\alpha_m$, $n\_epochs$
Output: $\theta$
Initialize $\theta$ randomly
for $D \in M_{\text{tr}}$ do
    for $(Tr, Te) \in D$ do
        for Batch $b \in Tr$ do
            $\theta_t = \theta - \alpha \nabla_\theta \mathcal{L}(f_\theta(b))$
        $\theta = \theta - \alpha_m \nabla_\theta \mathcal{L}(f_{\theta_t}(Te))/|M_{\text{tr}}|$
for $(Tr, Te) \in M_{\text{te}}$ do
    for Epoch $e \in n\_epochs$ do
        for Batch $b \in Tr$ do
            $\theta = \theta - \alpha \nabla_\theta \mathcal{L}(f_\theta(b))$
    $error+ = E(f_\theta(Te))$
return $error/|M_{\text{te}}|$
```

COUNTRY	TIME	REGIONS	AVG NEW CASE
ITALY	24/2-12/5	105	25.65
ENGLAND	13/3-12/5	129	16.7
SPAIN	12/3-12/5	35	61
FRANCE	10/3-12/5	81	7.5

Table 1: Summary of the available data for the 3 considered countries.

4.1 Datasets

Facebook has released several datasets in the scope of Data For Good program[2] to help researchers better understand the dynamics of the COVID-19 and forecast the spread of the disease (Maas et al. 2019). We use a dataset that consists of measures of human mobility between administrative NUTS3[3] regions. The data is collected directly from mobile phones that have the Facebook application installed and the Location History setting enabled. The raw data contains three recordings per day (i. e., midnight, morning, and afternoon), indicating the number of people moving from one region to another at that point of the day. We compute a single value for each day and each pair of regions by aggregating these three values. We focus on 4 European countries: Italy, Spain, France, and England. The number of cases in the different regions of the 4 considered countries was collected from open data listed in the github repository[4] along with the code and the aggregated mobility data. An overview of the preprocessed data can be found in Table 1. The start date is the earliest date for which we have both mobility data and data related to the number of cases available. Some regions were removed either due to having no COVID-19 cases or due to being unable to map with the Facebook mobility data.

4.2 Experimental Setup

In our experiments, we train the models using data from day 1 to day T, and then use the model to predict the number of cases for each one of the next dt days (i. e., from day $T+1$ to day $T+dt$). We are interested in evaluating the effectiveness of the model in short-, mid- and long-term predictions. Therefore, we set dt equal to 14. We expect the short-term predictions (i. e., small values of dt) to be more accurate than the long-term predictions (i. e., large values of dt). Note that we train a different model to predict the number of cases for days $T+i$ and $T+j$ where $i, j > 0$ and $i \neq j$. Therefore, each model focuses on predicting the number of cases after a fixed time horizon, ranging from 1 day to 14 days. With regards to the value of T, it is initially set equal to 14 and is gradually increased (one day at a time). Therefore, the size of the training set increases as time progresses. Note that a different model is trained for each value of T. Furthermore, for each value of T, to identify the best model, we build a validation set which contains the samples corresponding to days $T-1$, $T-3$, $T-5$, $T-7$ and $T-9$, such that the training and validation sets have no overlap with the test set.

With regards to the hyperparameters of the **MPNN**, we train the models for a maximum of 500 epochs with early stopping after 50 epochs of patience. Early stopping starts to occur from the 100th epoch and onward. We set the batch size to 8. We use the Adam optimizer with a learning rate of 10^{-3}. We set the number of hidden units of the neighborhood aggregation layers to 64. Batch normalization and dropout are applied to the output of every neighborhood aggregation layer, with a dropout ratio of 0.5. We store the model that achieved the highest validation accuracy in the disk and then retrieve it to make predictions about the test samples. For the **MPNN+LSTM** model, the dimensionality of the hidden states of the LSTMs is set equal to 64. All the models are implemented with PyTorch (Paszke et al. 2019). We evaluate the performance of a model by comparing the predicted total number of cases in each region versus the corresponding ground truth, throughout the test set:

$$\text{error} = \frac{1}{n\,dt} \sum_{t=T+1}^{T+dt} \sum_{v \in V} |\hat{y}_v^{(t)} - y_v^{(t)}| \qquad (5)$$

4.3 Baselines

We compare the proposed models against the following baselines and benchmark methods, which have been applied to the problem of COVID-19 forecasting:

- AVG: The average number of cases for the specific region up to the time of the test day.
- AVG_WINDOW: The average number of cases in the past d days for the specific region where d is the size of the window.
- LAST_DAY: The number of cases of the previous day is the prediction for the next days.
- LSTM (Chimmula and Zhang 2020): A two-layer LSTM that takes as input the sequence of new cases in a region for the previous week.

[2] https://dataforgood.fb.com/tools/disease-prevention-maps/

[3] https://en.wikipedia.org/wiki/Category:NUTS_3_statistical_regions_of_the_European_Union

[4] https://github.com/geopanag/pandemic_tgnn

Model	Up to next 3 Days				Up to next 7 Days				Up to next 14 Days			
	England	France	Italy	Spain	England	France	Italy	Spain	England	France	Italy	Spain
AVG	9.75	8.50	21.38	45.10	9.99	8.55	22.23	45.87	10.09	8.55	23.09	47.63
LAST_DAY	7.11	7.47	17.40	33.58	7.62	7.37	18.49	37.06	8.66	8.03	20.69	43.63
AVG_WINDOW	6.52	6.04	15.17	32.19	7.34	6.40	16.81	36.06	8.54	7.24	19.45	42.79
LSTM	9.11	8.08	22.94	51.44	8.97	8.13	23.17	49.89	9.10	7.91	23.12	47.26
ARIMA	13.77	10.72	35.28	40.49	14.55	10.53	37.23	41.64	15.65	10.91	39.65	46.22
PROPHET	10.58	10.34	24.86	54.76	12.25	11.56	27.39	62.16	16.24	14.61	33.07	79.42
TL_BASE	9.65	7.67	19.12	42.25	12.30	9.21	23.44	52.29	13.48	12.27	24.89	59.68
MPNN	6.36	6.16	14.39	35.83	6.86	5.99	15.47	38.51	8.13	6.93	17.88	44.25
MPNN+LSTM	6.41	6.39	15.56	33.35	6.67	7.21	16.41	34.47	7.02	7.36	17.25	35.31
MPNN+TL	**6.05**	**5.83**	**14.08**	**29.61**	**6.33**	**5.90**	**14.61**	**31.55**	**6.84**	**6.13**	**16.69**	**34.65**

Table 2: Average error for $dt = 3, 7$ and 14, in number of cases per region.

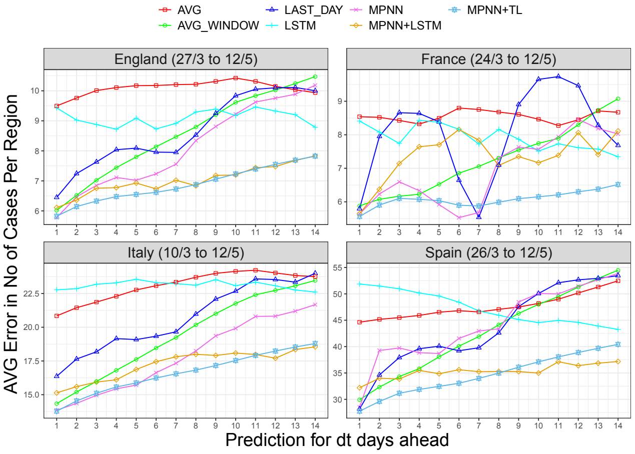


Figure 4: Average error in number of cases per region for each target shift. PROPHET and ARIMA are omitted for illustration purposes.

- ARIMA (Kufel 2020): A simple autoregressive moving average model where the input is the whole time-series of the region up to before the testing day.
- PROPHET (Mahmud 2020): A forecasting model for various types of time series[5].The input is similar to ARIMA.
- **TL_BASE**: An MPNN that is trained on all data from the three countries and the train set of the fourth (concatenated), and tested on the test set of the fourth. This serves as a baseline to quantify the usefulness of **MPNN+TL**.

We should note here that we cannot utilize models that work with recoveries and deaths such as SEIR because we only have the confirmed cases. That said, a simple approach is to run SI at every T with a β taken from the COVID-19 literature, along with the number of infected people at T and the population. In the preliminary experiments, however, this provided errors in a different scale than the ones mentioned here, similar to Gao et al. (2020), which is why we have not experimented further.

4.4 Results and Discussion

The average error per region for each one of the next 14 days is illustrated in Figure 4. We observe that the proposed models yield lower average errors compared to the baselines, **MPNN+TL** having the best performance. In terms of short term predictions its performance is close to some baselines, but the difference increases for longer scopes. Even simple baselines can be competitive at predicting the next day's number of cases since proximal samples for the same region from the same phases of the pandemic tend to have a similar number of cases. However, a prediction that goes deeper in time requires the identification of more persistent patterns. In the case of our model, as mentioned above, we aim to capture unregistered cases moving from one region to the other or spreading the disease in their new region. These cases would inevitably take a few days to appear, due to the delay of symptoms associated with COVID-19. This is why **MPNN** performs well throughout the 14-days window. The results also demonstrate the benefit of transfer learning techniques since **MPNN+TL** outperforms **MPNN** and its baseline **TL_BASE** in all cases. We expect **MPNN** to perform similar towards the end of the dataset, when the training of both models has become similar due to the number of epochs and. The main difference occurs when T is small, where the training samples are scarce and **MPNN** is unable to capture the underlying dynamics. One way to see this is again the accuracy of **MPNN+TL** in the long term predictions. Due to the size of the prediction window, long term tasks have diminished train set, meaning if the task is to predict t+14 and the set ends at day 60, t+14 training will stop at day 46 while the t+1 will stop at 59. Thus **MPNN** performs similarly at the short-term predictions but fails compared to **MPNN+TL** in the long term.

Note that for clarity of illustration, we chose not to visualize the performance of PROPHET and ARIMA in Figure 4 as their error was distorting the plot. However, we present in Table 2 the average error for the predictions where dt takes three values: 3, 7 and 14. Overall, it is clear that the time series methods (i. e., LSTM, PROPHET, and ARIMA) and the temporal variant of our method **MPNN+LSTM** yield quite inaccurate predictions. Apart from the inherent difficulty of learning with time-series data, which we analyze

[5] https://github.com/facebook/prophet

further below, this might also happen because of the nature of the epidemic curve. Specifically, sequential models, that are trained with values that tend to increase, are impossible to predict decreasing or stable values. For instance, in our dataset, once the models have enough samples to learn from, there is a transition in the phase of the epidemic due to lockdown measures. The same applies when the epidemic starts to recede at the start of May.

Our error function treats all regions equivalently, independent of the region-specific population and the number of cases, i. e., a region with 10 cases per day should not be treated the same as a region with 1000. We need a measure to take into account the region-specific characteristics as well as the time. Towards this end, we computed the deviation between the average predicted and the actual average number of cases over the next 5 days (not to be confused with the average error for the next x days in Table 2). The relative error of each region is defined below and is illustrated in Figure 5.

$$r = \frac{1}{n\,(T-5)} \sum_{t=1}^{T-5} \sum_{v \in V} \frac{\left| \sum_{i=0}^{5} \hat{y}_v^{(t+i)} - \sum_{i=0}^{5} y_v^{(t+i)} \right|}{\sum_{i=0}^{5} y_v^{(t+i)}} \tag{6}$$

One can see that the regions with high relative error are the ones with the fewest cases. On the other hand, the regions with the highest number of cases tend to have much smaller relative error, less than 20% to be exact, with the exception of one region in Spain. This indicates that our model indeed produces accurate predictions that could be useful in resource allocation and policy-making during the pandemic.

Figure 5 also allows us to evaluate the method more objectively. From a machine learning perspective, one may argue that even though the **MPNN+TL** outperforms the baselines, their predictions are not very accurate in terms of average error. This is partially explained due to the inherent problems of the dataset mentioned at subsection 4.1 as well as the assumption that case reporting is standard throughout the regions. We expect a large improvement in performance in case a standard methodology for case reporting is adopted and the number of tests per region remains constant and proportional to the population. Having said that, utilizing such a model in practice is more than feasible as the difference in scale is more useful at the regional level. In other words, a region predicted to have 10 new cases in the next 5 days will generally have similar needs (e. g., hospital capacity needs) with a region with 5 or 15 real cases (50% error). On the other hand, predicting 500 instead of 1000 cases, although identical in terms of relative error, can prove disastrous. This is not the case for our model, since the regions with large relative error tend to have a small number of cases, e. g., the model may predict 20 cases instead of 10 (100% error), which is arguably not significant.

5 Conclusion

In this paper, we presented a model for COVID-19 forecasting that could provide useful insights to policymakers.We use mobility data as a graph where nodes correspond to regions and edge weights to mass mobility between their end-points. Then, we derive variants of the family of MPNNs to generate representations for the regions based on their interactions and history. Furthermore, since different countries might be in different phases of the epidemic, we propose to transfer the disease spreading model from one country to another where limited data is available. Experiments conducted on data from 4 European countries show that our architectures outperform traditional and more recent approaches in predicting the number of daily new COVID-19 cases.

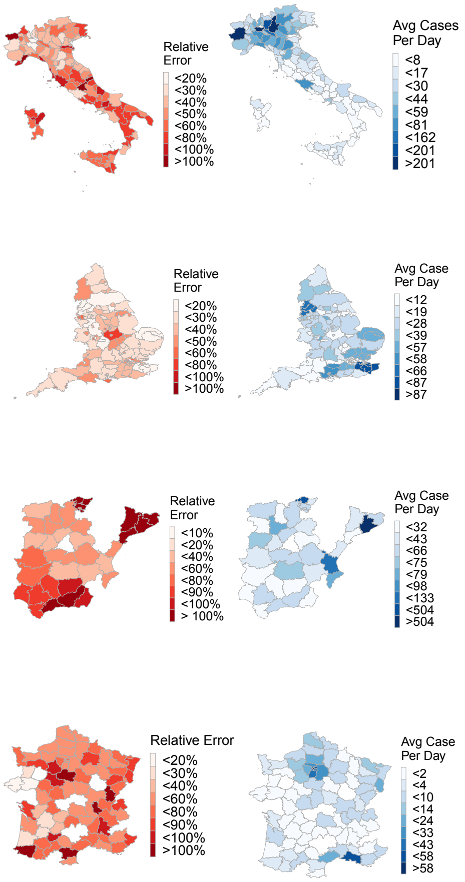


Figure 5: Plot of the relative test error and average number of cases per day for each available region.

Acknowledgements

This research is i) supported by the French National research agency via the ANR XCOVIF (AAP RA-COVID-19 V6) project, and ii) co-financed by Greece and the European Union (European Social Fund- ESF) through the Operational Programme ≪ Human Resources Development, Education and Lifelong Learning≫ in the context of the project "Reinforcement of Postdoctoral Researchers - 2$^{\text{nd}}$ Cycle" (MIS-5033021), implemented by the State Scholarships Foundation (IKY).

References

Chimmula, V. K. R.; and Zhang, L. 2020. Time Series Forecasting of COVID-19 transmission in Canada Using LSTM Networks. *Chaos, Solitons & Fractals* 109864.

Colizza, V.; Barrat, A.; Barthelemy, M.; and Vespignani, A. 2006. Prediction and predictability of global epidemics: the role of the airline transportation network. *Bulletin of the American Physical Society* .

Deng, S.; Wang, S.; Rangwala, H.; Wang, L.; and Ning, Y. 2019. Graph message passing with cross-location attentions for long-term ILI prediction. *arXiv preprint arXiv:1912.10202* .

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. ICML*, 1126–1135.

Flaxman, S.; Mishra, S.; Gandy, A.; Unwin, H. J. T.; Mellan, T. A.; Coupland, H.; Whittaker, C.; Zhu, H.; Berah, T.; Eaton, J. W.; et al. 2020. Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature* 584(7820): 257–261.

Gao, J.; Sharma, R.; Qian, C.; Glass, L. M.; Spaeder, J.; Romberg, J.; Sun, J.; and Xiao, C. 2020. STAN: Spatio-Temporal Attention Network for Pandemic Prediction Using Real World Evidence. *arXiv preprint arXiv:2008.04215* .

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for Quantum chemistry. In *Proc. ICML*, 1263–1272.

Graves, A.; and Jaitly, N. 2014. Towards End-to-End Speech Recognition with Recurrent Neural Networks. In *Proc. ICML*, 1764–1772.

Kapoor, A.; Ben, X.; Liu, L.; Perozzi, B.; Barnes, M.; Blais, M.; and O'Banion, S. 2020. Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks. In *MLG*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kufel, T. 2020. ARIMA-based forecasting of the dynamics of confirmed Covid-19 cases for selected European countries. *Equilibrium. Quarterly Journal of Economics and Economic Policy* 15(2): 181–204.

Lampos, V.; Moura, S.; Yom-Tov, E.; Cox, I. J.; McKendry, R.; and Edelstein, M. 2020. Tracking COVID-19 using online search. *arXiv preprint arXiv:2003.08086* .

Lee, J.; Kim, H.; Lee, J.; and Yoon, S. 2017. Transfer learning for deep learning on graph-structured data. In *Proc. AAAI*.

Lorch, L.; Trouleau, W.; Tsirtsis, S.; Szanto, A.; Schölkopf, B.; and Gomez-Rodriguez, M. 2020. A spatiotemporal epidemic model to quantify the effects of contact tracing, testing, and containment. *arXiv preprint arXiv:2004.07641* .

Maas, P.; Iyer, S.; Gros, A.; Park, W.; McGorman, L.; Nayak, C.; and Dow, P. A. 2019. Facebook Disaster Maps: Aggregate Insights for Crisis Response & Recovery. In *ISCRAM*.

Mahmud, S. 2020. Bangladesh COVID-19 Daily Cases Time Series Analysis using Facebook Prophet Model. *Available at SSRN 3660368* .

Mallick, T.; Balaprakash, P.; Rask, E.; and Macfarlane, J. 2020. Transfer Learning with Graph Neural Networks for Short-Term Highway Traffic Forecasting. *arXiv preprint arXiv:2004.08038* .

Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; and Grohe, M. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proc. AAAI*, 4602–4609.

Nikolentzos, G.; Tixier, A. J.-P.; and Vazirgiannis, M. 2020. Message Passing Attention Networks for Document Understanding. In *Proc. AAAI*, 8544–8551.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proc. NeurIPS*, 8026–8037.

Soriano-Panos, D.; Ghoshal, G.; Arenas, A.; and Gómez-Gardenes, J. 2020. Impact of temporal scales and recurrent mobility patterns on the unfolding of epidemics. *Journal of Statistical Mechanics: Theory and Experiment* 2020(2): 024006.

Tang, X.; Li, Y.; Sun, Y.; Yao, H.; Mitra, P.; and Wang, S. 2020. Transferring Robustness for Graph Neural Network Against Poisoning Attacks. In *Proc. WSDM*, 600–608.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Li, Z.; Ye, J.; and Chuxing, D. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proc. AAAI*, 2588–2595.

Yao, H.; Zhang, C.; Wei, Y.; Jiang, M.; Wang, S.; Huang, J.; Chawla, N.; and Li, Z. 2020. Graph few-shot learning via knowledge transfer. In *Proc. AAAI*, 6656–6663.

Zeroual, A.; Harrou, F.; Dairi, A.; and Sun, Y. 2020. Deep Learning Methods for Forecasting COVID-19 Time-Series Data: A Comparative Study. *Chaos, Solitons & Fractals* 110121.

Zou, B.; Lampos, V.; and Cox, I. 2019. Transfer Learning for Unsupervised Influenza-like Illness Models from Online Search Data. In *Proc. WebConf*, 2505–2516.