Exploring the Predictive Power of Correlation and Mutual Information in Attention Temporal Graph Convolutional Network for COVID-19 Forecasting

Subas Rana(✉), Nasid Habib Barna, and John A. Miller

University of Georgia, Athens, GA 30602, USA
{subas.rana,nasidhabib.barna,jamill}@uga.edu

Abstract. Accurately forecasting the COVID-19 spread across all states is crucial for implementing effective measures to control its transmission and minimize its impact. Since the virus' spread in one state can significantly affect other states over time through connections between them, a graph structure with temporal data is needed to capture the interdependence of COVID-19 spread among the states in the United States. In forecasting tasks that involve complex spatial and temporal dependencies, it is crucial to ensure that the model captures these dependencies accurately. In this study, we implemented an Attention Temporal Graph Convolutional Network based model for COVID-19 mortality long-term prediction which can effectively capture these dependencies. This model incorporates attention that enables us to weigh the significance of different time points and focus on the most informative data, including both adjacent and distant time points that capture the temporal dynamics accurately. For capturing spatial dependencies, we assessed the impact of using Pearson's correlation and Mutual Information to establish connections between highly dependent states. Our experiments showed that our model, particularly when utilizing mutual information, outperformed the existing baselines and the models that only consider neighboring states resulting in lower sMAPE and MAE values. This emphasizes the importance of selecting the appropriate technique for accurate COVID-19 forecasting in each state. Furthermore, our model achieved the second-highest performance among the forecasting models submitted to the Centers for Disease Control and Prevention.

Keywords: COVID-19 forecasting · Attention Temporal Graph Convolutional Network · PyTorch Geometric Temporal · Pearson's Correlation · Mutual Information

Supported by University of Georgia.

S. Zhang et al. (Eds.): BigData 2023, LNCS 14203, pp. 18–33, 2023.
https://doi.org/10.1007/978-3-031-44725-9_2

1 Introduction

The first case of COVID-19 in the United States (US) was reported in the state of Washington on January 20, 2020. Since then, the virus has spread rapidly across all 50 states, resulting in over 104 million confirmed cases and 1 million deaths as of May 4, 2023 [1]. The availability of daily and weekly COVID-19 data from various countries and states has provided opportunities to develop improved time-series models. Several statistical and machine learning (ML) models [2–4] have been used for COVID-19 forecasting. These methods either use the epidemic data of a single region to predict the future trend of the epidemic situation or establish a generalized model to predict the trend of all regions. Extensive research [5,6] has confirmed the highly contagious nature of the virus, and these studies have identified factors that contribute to its rapid spread across regions. Consequently, the spread of COVID-19 within a particular state can exert a substantial influence on neighboring states over time. This influence arises from various interconnected factors such as travel, trade, and social connections. To accurately forecast the spread of the virus in a particular state, it is important to consider the graph structure to capture these interconnections between states.

Recent studies [7–9] have developed spatiotemporal Graph Neural Networks (GNNs) that operate on graphs, representing regions as nodes and capturing spatial and temporal dependencies between them. In forecasting tasks that involve complex spatial and temporal dependencies, it is crucial to ensure that the model captures both types of relationships accurately. These GNNs combine Graph Convolutional Networks (GCNs) to capture spatial dependencies and Recurrent Neural Networks (RNNs) or their variants to model temporal dependencies. RNNs process sequential data over time, and their hidden states carry the latest information from the past. However, this sequential processing can restrict the model's access to global information present throughout the input sequence. To address this issue, Attention mechanisms offer a solution by allowing the model to focus on relevant parts of the sequence and assign different weights, providing a means to learn and leverage global correlations. We utilize an attention-based model from PyTorch Geometric Temporal library (PyGT) [10] to capture global information for COVID-19 weekly mortality prediction.

Another crucial aspect of GNN-based models is to accurately capture strong connections between different regions. Many existing GNN-based models [9,11,12] for epidemic prediction rely on data such as mobility or social connections to establish connections between regions and capture spatial dependencies. However, acquiring and utilizing such data can pose challenges related to data availability, privacy, and accuracy. In contrast, we utilize both correlation and mutual information (MI) to capture both linear and nonlinear dependencies between the state's features. For instance, our approach shows a high correlation between Ohio and Illinois, which suggests there is a strong linear relationship between the deaths or confirmed cases in these two states. Specifically, it suggests that as deaths or confirmed cases increase in one state, they tend to increase in the other state as well. This relationship can be seen in Fig. 1.

The contributions of our proposed method are as follows:

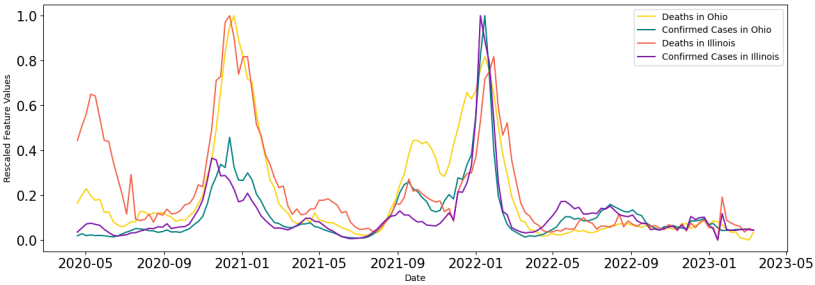


Fig. 1. Rescaled Deaths and Confirmed Cases in Ohio and Illinois indicating a strong linear relationship

1. We use the Attention Temporal Graph Convolutional Network (A3T-GCN) model from PyGT on the state graph for COVID-19 forecasting. The complete model allowed us to capture both local and global information about all the regions in the model, allowing us to capture temporal dependencies effectively.
2. We use correlation and MI to capture both linear and nonlinear dependencies and establish connections between highly dependent states in a graph, allowing us to effectively capture spatial dependencies. We use these graphs in the A3T-GCN model and compare these techniques with A3T-GCN based on adjacent states (states that share a border). We also compare our best models with the baselines.
3. We compared and achieved the second-highest performance among the forecasting models submitted to the Centers for Disease Control and Prevention (CDC).

2 Related Work

Several studies have applied statistical, ML, and deep learning methods to predict COVID-19 forecasting based on various clinical, laboratory, and epidemiological data. Infectious disease prediction is often modeled as a time-series prediction problem in many studies, so many time-series methods have been studied [3]. Chimmula et al. [2] analyzed the important factors and applied a Long Short Term Memory (LSTM) model to forecast the patterns and probable end date of the COVID-19 pandemic in Canada and also around the world. Cramer et al. [13] conducted a study that evaluates the effectiveness of both individual and ensemble probabilistic forecasts for predicting COVID-19 mortality in the US. The study focuses on evaluating the accuracy and reliability of these forecasts to provide insights into the effectiveness of different forecasting methods. Their work has been used by CDC for COVID-19 cases and death forecasts.

Kapoor et al. [7] presented a forecasting method for COVID-19 cases using a spatiotemporal GNN that aims to capture the intricate dynamics involved in disease modeling by incorporating mobility data. In their proposed model, nodes in the graph represent county-level human mobility, spatial edges depict

inter-regional interactions, and temporal edges account for the evolution of node features over time. Panagopoulos et al. [9] introduced MPNN-TL, a GNN for COVID-19 dynamics across four European countries. By integrating mobility data and reported cases, the model aims to comprehend complex transmission patterns. It recognizes the vital role of mobility patterns in the disease's spread, emphasizing the significance of a graph-based representation for studying transmission dynamics and its impacts. Fritz et al. [14] proposed a fusion approach that combines GNN with epidemiological models to forecast the infection rate of COVID-19. The study utilized data from Facebook, including mobility and association information, as well as structural and spatial details of cities and districts in Germany. Cao et al. [15] developed StemGNN, a model for multivariate time series that captures inter and intra-temporal correlations using the Graph Fourier Transform (GFT) and Discrete Fourier Transform (DFT). With a graph structure representing different countries, the study evaluated the model's performance in forecasting confirmed cases across multiple horizons.

3 Methodology

In this work, we utilized the A3T-GCN model from PyGT on the state graph to predict COVID-19 outcomes. This model allowed us to effectively consider both local and global information from all regions, capturing temporal dependencies. To capture both linear and nonlinear dependencies and establish connections between highly dependent states, we incorporated correlation and MI. By integrating these graphs into the A3T-GCN model, we compared its performance with the version that only considered adjacent states. Our evaluation showed a significant improvement when using different association techniques beyond adjacency. In this section, we will discuss the PyGT library, the model architecture with a customized dataset, and the association techniques used for the model.

3.1 PyGT

PyGT is an extension library for PyTorch Geometric. It is specifically designed for handling spatiotemporal data, such as dynamic graphs where edges and nodes change over time. PyGT includes various tools for creating, manipulating, and visualizing temporal graphs, as well as implementing GNN architectures for spatiotemporal data. This library provides several data iterators. We use **StaticGraphTemporalSignal** which is used when the underlying graph is fixed, but the features on each node or edge change over time. This data iterator provides an efficient way to iterate over temporal snapshots of a graph in batches. The library also comes with a train-test splitter that creates temporal splits of the data using a fixed split ratio and some benchmark datasets. In addition to that, it includes several types of existing Neural network models that operate on graphs. We use the model A3T-GCN which is based on the paper "A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting" [16].

3.2 A3T-GCN

We implement an A3T-GCN (the second version of A3T-GCN) based model which is capable of handling batches. Our objective with this model is to use past values of COVID-19 deaths and confirmed cases to predict the weekly death counts for each state. This model, shown in Fig. 2, is a combination of GCN and Gated Recurrent Unit (GRU) and features an attention mechanism.

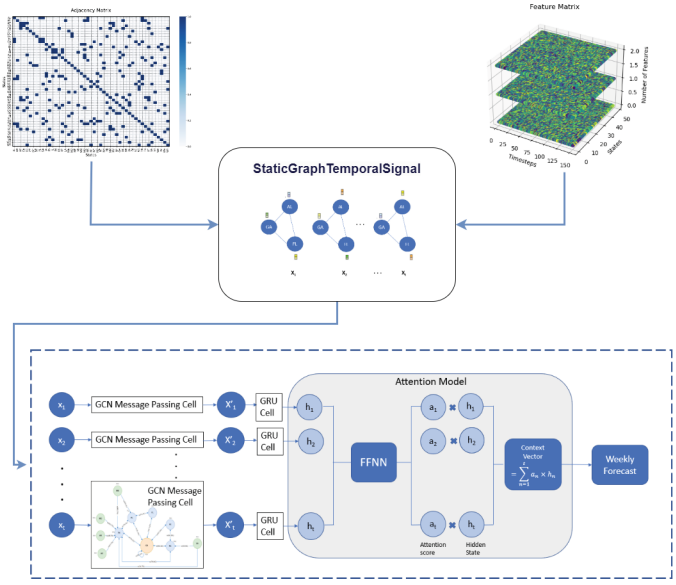


Fig. 2. Customized PyGT dataset based A3T-GCN architecture

The library requires a customized PyTorch Geometric-based dataset that is tailored to the problem of predicting COVID-19 deaths for the states. To achieve this, we define an adjacency matrix (based on adjacent states as an example in this figure) and a feature matrix with information on timesteps, number of states, and features. These matrices are passed to the model expanded from the A3T-GCN [16] paper as shown in Fig. 2.

1. **Adjacency Matrix:** A graph $G = (V, E)$ defines the graph structure of the US states where each node $v \in V$ represents a state in the US and each edge $e \in E$ represents the edge between two states. This whole information is defined by an adjacency matrix $A \in R^{N \times N}$ with N vertices (states in our case) and all the rows i and columns j are indexed by the states. The entry in i and j of the matrix, denoted by $A[i, j]$, represents the weight or degree of interdependence between state i and state j.
   We use correlation and MI described in the next section to construct the matrices based on the degree of dependency between pairs of states. The

procedure involves calculating the correlation and MI scores between the variables. We consider all pairs of states with a score greater than a certain threshold to be highly dependent and connect them in the adjacency matrix. Conversely, we set the weight of the edge to zero for all pairs of states with the scores below the threshold, indicating that they are not strongly interconnected.

2. **Feature Matrix:** COVID-19 deaths and confirmed cases are shown as the features of the nodes represented by a feature matrix $X \in R^{N \times F}$, where N is the number of states and F is the number of features. These features change over time, X_t is the features matrix at time t. We passed the historical values of all the features $X_{t-n}, \cdot, X_{t-1}, X_t$ along with the adjacency matrix A through the **StaticGraphTemporalSignal** iterator which returns PyTorch Geometric Data object for a single time period i.e., a week, which represents a snapshot of the graph for that time period. These temporal snapshots represent different feature values for the same underlying graph structure and are passed as historical inputs to the GCN layers of the A3T-GCN model, which perform computations on the graph structure and associated features to produce output representations of the hidden state.

GCN. In a GCN, each node in the graph is associated with a feature vector, and the goal is to learn a new set of feature vectors that capture the relationships in the graph through a message-passing process. Specifically, at each layer of the GCN, the feature vector of each node is updated by taking a weighted sum of the feature vectors of its neighbors, where the weights are learned by the network. This weighted sum is then combined with the feature vector of the node itself to produce a new feature vector. This enables the model to capture spatial dependencies. A two-layer GCN model [17] can be defined in (1) below:

$$f(X, A) = \sigma(\overline{A}\ Sigmoid\ (\overline{A}\ X\ W_0)\ W_1) \tag{1}$$

$$\overline{A} = D'^{-.5} A' D'^{-.5} \tag{2}$$

$$A' = A + I_N \tag{3}$$

Here X is a feature matrix and A is an adjacency matrix defined above, $\overline{A}$ defined in (2) is a preprocessing step, where A' is an adjacency matrix with self connectivity defined in (3), I_N is an identity matrix, D' is a degree matrix. $W_0 \in R^{K \times H}$ is a weight matrix from the input layer to the hidden unit layer where K is the length of time and H is hidden unit numbers, and $W_1 \in R^{H \times T}$ is a weight matrix from the hidden layer to the output layer, where T is the length of forecasted points. $f(X, A) \in R^{N \times T}$ is the output of the GCN model of the length T. The updated vectors are then passed to GRU.

GRU. GRUs have two gates, namely an update gate and a reset gate. The update gate determines how much of the previous hidden state should be retained for the current time step, and the reset gate controls how much of the new input should be incorporated into the new hidden state.

In our case, the gated mechanism of GRU allows them to capture the mortality information at the current moment while retaining the variation trends of historical COVID information. As a result, this model can effectively capture the dynamic temporal variation features of COVID data.

Attention. The Attention model, which is a modified version of the Encoder-Decoder model, was initially developed for use in neural machine translation tasks [18]. In this particular study, a soft Attention model was employed to determine the importance of COVID information at each moment. This information was then used to calculate a context vector that captured the overall trends in the COVID mortality state, which could be utilized for predicting future mortality conditions.

In order to calculate the weight of each hidden state, a scoring function is designed, followed by an attention function that computes the context vector to capture the global COVID data variation information.

$$e_i = W_2(W_1 H + d_1) + d_2 \tag{4}$$

$$a_i = \frac{exp(e_i)}{\sum_{k=1}^{n} exp(e_k)} \tag{5}$$

$$C_t = \sum_{i=1}^{n} a_i \times h_i \tag{6}$$

Here H is the set of hidden states $\{h_1, h_2, \ldots, h_n\}$, W_1 and d_1 are the weight and deviation of the first layer and W_2 and d_2 are the weight and deviation of the second layer. The context vector $C_t \in R^{N \times T}$ captures global COVID data information and lastly, a linear layer is used to produce the final output results. In this study, a Feed Forward Neural Network (FFNN) was used instead of a scoring function to determine the weight of each hidden state resulting from the GRU.

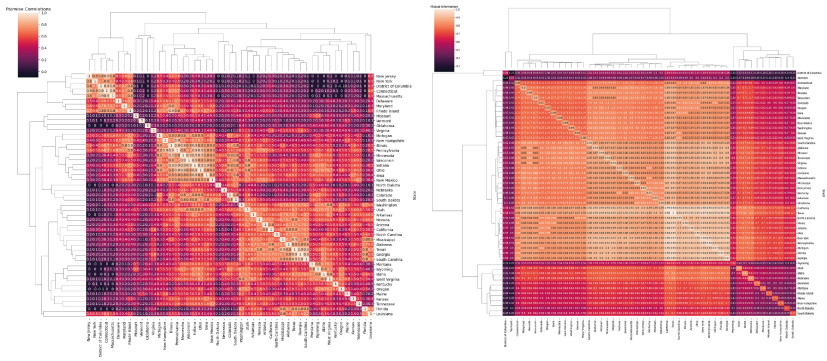
3.3 Correlation

Correlation is used to measure the linear relationships between the variables. However, it does not measure the strength of a nonlinear relationship between variables. Figure 3a shows the adjacency matrix based on correlation. To connect the states/nodes, we only selected those with high correlation coefficients.

3.4 MI

MI [19] measures the mutual dependence between two random variables. It measures the amount of information that one random variable provides about the other random variable. It can measure both linear and nonlinear relationships between two random variables. MI is derived from the definition of entropy and is defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{7}$$



(a) Correlation Matrix (b) MI Matrix

Fig. 3. Association matrices

Let X and Y be two random variables over the space $X \times Y$. $I(X;Y)$ is the MI calculated between the two variables, $P(X,Y)$ is their joint distribution and $P(X)$ and $P(Y)$ is the marginal distribution of X and Y, respectively. The MI score is greater than or equal to zero. If it is zero, it means that the two variables are independent, i.e., knowing the value of one variable does not provide any information about the other variable. If the MI score is greater than zero, the two variables are dependent, and the larger the MI score, the stronger the dependence.

Figure 3b shows the MI-based matrix. To create edges between the states, we only selected those with high MI scores. Additionally, we rescaled the MI values by dividing each value by its maximum.

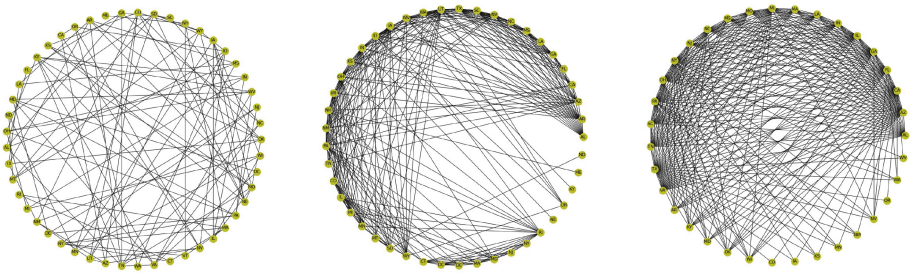
For both techniques, we experimented with different thresholds and selected the ones that yielded the best results. We used 0.6 for correlation and 0.85 for MI which preserved the strong connections between states while discarding weaker ones that fell below the threshold. For creating a weighted graph, we assigned the scores as edge weights for connected edges and assigned 0 for disconnected ones. Figure 4 shows how the graph changes based on the association techniques.

4 Experiments and Results

4.1 Data and Code

The weekly dataset used for this study is available on GitHub https://github.com/scalation/data/blob/master/COVID-State/2023-05-02-17-53-19-State_Weekly.csv. It contains 19 columns and 8819 rows corresponding to weekly reporting for each region in the US from April 18^{th}, 2020, to March 11^{th}, 2023. We only consider the data for states and drop other regions. For features, we use $Confirmed$ and $Deaths$ because other features have missing values.

The data was extracted from the COVID-19 Dataset by COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at



(a) Adjacency Graph (b) Correlation Graph (c) MI Graph

Fig. 4. Connection between states with different association techniques

Johns Hopkins University [20] (JHU) https://github.com/CSSEGISandData/COVID-19. Daily data were cumulative and produced some negative values when converted to non-cumulative, for instance, Maryland's dataset showed 1140 deaths on 04/30/2020 and 1080 on 05/01/2020 which resulted in -60 deaths on May 1^{st}, 2020. To resolve this issue, we opted to aggregate the data on a weekly basis instead. Once converted, we removed the initial missing values and divided the dataset into an 80/20 split using the PyGT Temporal Signal Train-Test Split for training and testing, which returns train and test data iterators. We employed the sliding window approach for model training, where each window consisted of 114 weeks in the training set and forecasted 4 weeks in the testing set. We then shifted the window by 1 week and repeated the process until the end of the dataset. Moreover, for CDC comparison, the data was extracted from https://covid.cdc.gov/COVID-DATA-TRACKER/?submenu-select=national-lab#datatracker-home. Our code is available on: https://www.github.com/Subasranaa/COVID-19-A3T-GCN2.

4.2 Implementation Details

We implemented our model in Python using PyTorch [21] and PyGT. The hyperparameters shown in Table 1 are fine-tuned using grid search. The activation function is ReLU, 100 epochs, and a 1e-4 learning rate using the ADAM optimizer. We used the mean squared error (MSE) as the loss function. The data were standardized (subtracting the mean and dividing by the standard deviation) and transformed back to the original data scale for evaluation. The model was evaluated on the symmetric mean absolute percentage error (sMAPE) and the mean absolute error (MAE). sMAPE measures the absolute difference between the actual and forecast values normalized by absolute values of both [22]. It is a bounded metric where 0% indicates a perfect model with no errors, while 200% indicates an erroneous model of opposite actual and forecast values [23]. The

sMAPE is calculated using (8)

$$\text{sMAPE} = \frac{200}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|} \tag{8}$$

where y_t is the true value, and $\hat{y}_t$ is the forecast value. MAE measures the mean absolute error between the actual and forecast values.

Statistical and ML Baselines. To compare our A3T-GCN model, we employed several baselines. Auto-Regressive(1) (AR) model is sufficient when the future is mainly dependent only on the most recent value. SARIMA is a time series forecasting model that combines AR, integrated (I), and moving average (MA) components to capture the patterns and seasonality in data. LSTM is designed to process and retain information over long sequences, enabling it to capture and learn from long-term dependencies in data. It uses a gating mechanism to selectively remember or forget information based on relevance. GRU is similar to LSTM but has fewer gates, making it faster to compute and easier to train. FFNN is where information flows in one direction, from the input layer through hidden layer(s) to the output layer, without any loops. We experimented with these models using various hyperparameters as shown in Table 1.

Table 1. Optimal hyperparameters for A3T-GCN and all the baselines. A3T-GCN, LSTM, GRU, and FFNN hyperparameters are listed in the sequence of [past values, batch size, hidden dimensions, learning rate]. FFNN utilized ELU and Tanh activation functions and four linear layers.

	Virginia	Georgia	Illinois	Pennsylvania	Kentucky
A3T-GCN	[6,16,4,0.0001]	[6,16,2,0.0001]	[6,16,32,0.0001]	[6,16,2,0.0001]	[6,16,16,0.0001]
SARIMA	$(6,0,1) \times (0,1,1)_{10}$	$(3,1,2) \times (1,0,1)_{10}$	$(0,1,0) \times (1,1,1)_{10}$	$(4,0,0) \times (5,1,1)_{10}$	$(3,1,1) \times (1,1,1)_{10}$
LSTM	[6,16,256,0.0006]	[6,16,32,0.0006]	[6,16,128,0.0006]	[6,16,256,0.0006]	[6,16,512,0.0006]
GRU	[6,16,128,0.0007]	[6,16,512,0.0007]	[6,16,64,0.0007]	[6,16,256,0.0007]	[6,16,128,0.0007]
FFNN	[4,16,256,0.0004]	[4,16,32,0.0004]	[4,16,32,0.0004]	[4,16,16,0.0004]	[4,16,512,0.0004]
	Maryland	Massachusetts	Minnesota	Ohio	Washington
A3T-GCN	[6,16,8,0.0001]	[6,16,32,0.0001]	[6,4,32,0.0001]	[6,16,16,0.0001]	[6,16,32,0.0001]
SARIMA	$(0,1,2) \times (1,0,1)_{10}$	$(1,1,0) \times (0,0,2)_{10}$	$(1,1,1) \times (0,0,1)_{10}$	$(2,0,0) \times (0,1,2)_{10}$	$(4,0,1) \times (0,0,1)_{10}$
LSTM	[6,16,8,0.0006]	[6,16,128,0.0006]	[6,16,64,0.0006]	[6,16,256,0.0006]	[6,16,512,0.0006]
GRU	[6,16,512,0.0007]	[6,16,32,0.0007]	[6,16,4,0.0007]	[6,16,512,0.0007]	[6,16,256,0.0007]
FFNN	[4,16,16,0.0004]	[4,16,32,0.0004]	[4,16,32,0.0004]	[4,16,16,0.0004]	[4,16,16,0.0004]

4.3 Results Analysis

Comparison of A3T-GCN Models Using Adjacent, Correlation, and MI Matrices. We implemented five A3T-GCN models using Adjacent, Correlation, MI, Correlation with adjacent (Corr_Adj), and MI with adjacent (MI_Adj) matrices. The reason behind this is each state was showing different trends when

they were connected to different states based on their relation. We are using the adjacent-based A3T-GCN model as the baseline here to compare it with other association technique-based models. Tables 2 and 3 show sMAPE and MAE results respectively, of 1 week, 2 weeks, 3 weeks, and 4 weeks ahead forecast for our best models and the baseline in this case.

Virginia, Ohio, Pennsylvania, and Washington demonstrate improved outcomes when using the MI-based A3T-GCN model, suggesting that quantifying the interdependence among these states through MI yields favorable results. The MI_Adj-based model gives us the best results for Georgia, Kentucky, Massachusetts, and Minnesota indicating that the involvement of neighboring states is significant for these particular states. For instance, in the case of Georgia, 21 more states got added along with adjacent states using MI which improved the predictions, showing a strong impact of linear and nonlinear dependencies with other states.

In the case of Maryland and Illinois, the Corr_Adj-based model shows superior performance than all the other models indicating that correlation plays an important role in them. Here adding correlated states led to superior results instead of just using adjacent states, our models give a slightly higher MAE score for Maryland even though it has a lower sMAPE score when we compare it with the adjacent-based model. Moreover, we also get competitive scores for Arizona, Iowa, Michigan, New York, Texas, and Indiana with sMAPEs as 25.52, 28.74, 26.87, 26.90, 25.25, and 27.53, respectively by using MI. Overall, using these association techniques over using only adjacent states shows significantly better performance in terms of forecasting.

Table 2. sMAPE results for 1-week, 2-weeks, 3-weeks, and 4-weeks ahead forecast for the states using Adjacent states-based A3T-GCN (Adjacent) and Correlation/MI-based A3T-GCN models (A3T-GCN). The lower the sMAPEs, the better the forecasting performance.

	Virginia		Georgia		Illinois		Pennsylvania		Kentucky	
Weeks	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN
1	39.96	13.02	22.82	15.07	23.12	11.38	88.93	17.14	20.74	14.28
2	45.36	12.58	27.50	13.77	21.57	9.37	90.10	28.46	19.14	12.37
3	48.41	11.14	33.24	17.77	16.98	10.81	91.32	15.89	16.53	11.23
4	53.60	13.55	22.27	26.38	19.93	14.66	95.40	23.07	14.75	11.83
Average	18.36	**12.57**	26.46	**18.25**	20.40	**11.56**	47.24	**21.14**	17.79	**12.43**
	Maryland		Massachusetts		Minnesota		Ohio		Washington	
Weeks	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN
1	23.47	14.11	15.44	13.65	18.38	12.31	16.16	14.81	28.27	12.90
2	18.87	14.34	17.16	13.91	14.10	12.04	18.16	6.52	27.31	13.95
3	14.45	18.04	18.09	13.33	21.43	14.35	16.26	7.28	30.05	13.90
4	15.83	23.34	20.84	13.45	19.45	13.94	16.59	14.63	31.79	12.77
Average	18.16	**17.46**	17.88	**13.58**	18.34	**13.16**	16.79	**10.81**	29.35	**13.38**

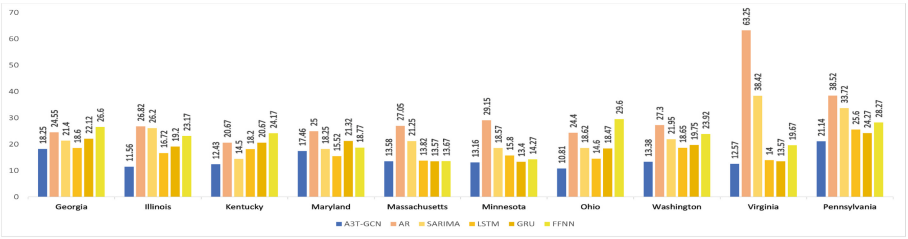
Table 3. MAE results for 1-week, 2-weeks, 3-weeks, and 4-weeks ahead forecast for the states using Adjacent states-based A3T-GCN (Adjacent) and Correlation/MI-based A3T-GCN models (A3T-GCN). The lower the MAEs, the better the forecasting performance.

	Virginia		Georgia		Illinois		Pennsylvania		Kentucky	
Weeks	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN
1	32.76	12.36	27.10	17.38	18.55	9.11	46.32	19.92	12.94	10.35
2	36.37	11.84	34.41	15.76	16.05	7.27	47.59	30.70	11.78	8.65
3	38.81	10.61	43.93	20.69	12.78	8.52	48.63	18.90	10.91	7.86
4	42.57	12.81	26.31	32.69	15.49	11.64	52.66	29.60	10.68	8.27
Average	37.63	**11.90**	32.94	**21.63**	15.72	**9.13**	48.80	**24.78**	11.58	**8.78**
	Maryland		Massachusetts		Minnesota		Ohio		Washington	
Weeks	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN	Adjacent	A3T-GCN
1	9.53	5.75	14.46	8.70	7.70	5.99	13.99	11.34	19.11	8.62
2	7.75	5.90	15.65	8.87	6.14	5.83	15.63	5.14	18.33	9.32
3	5.81	7.90	15.90	8.47	9.09	7.00	14.20	5.85	20.69	9.22
4	6.62	10.71	19.25	8.56	9.20	6.83	14.48	12.14	22.20	8.52
Average	**7.43**	7.57	16.32	**8.65**	8.03	**6.41**	14.57	**8.62**	20.08	**8.92**

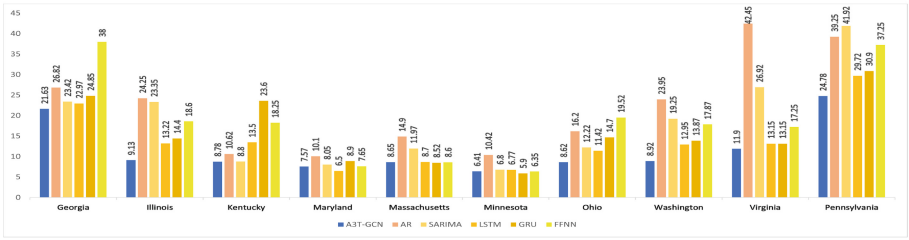
Baselines Comparison. In this work, we compare our best-performing A3T-GCN models against the baselines. Figure 5 displays the average results of sMAPE and MAE for our best A3T-GCN models and the baselines across the states. A3T-GCN achieves a 52.96% reduction in sMAPE and a 46.84% reduction in MAE compared to AR and a 37.99% reduction in sMAPE and a 36.32% reduction in MAE compared to SARIMA. Although LSTM performs slightly better than A3T-GCN for Maryland, when considering the average results across all states, A3T-GCN demonstrates a 15.86% lower sMAPE and a 16.26% lower MAE. Compared to GRU, A3T-GCN achieves a 22.55% lower sMAPE and a 26.72% lower MAE, despite GRU outperforming in the case of Massachusetts and Minnesota (for MAE only). Finally, when compared to FFNN, A3T-GCN shows a 35.03% lower sMAPE and a 38.51% lower MAE.

This is likely due to the model's use of Attention, GCN, and GRU, which can capture non-linear dependencies and temporal relationships between distant data points. In contrast, AR and SARIMA may not be able to capture these non-linear dependencies and relationships, leading to lower performance. The findings also highlight that relying solely on historical data from a single state may not be sufficient for accurately predicting deaths, as the interactions and relationships between different states can impact the results. Therefore, it is crucial to consider these relationships in the model, which A3T-GCN can capture. This becomes particularly significant when dealing with time series data. While LSTM, FFNN, and GRU performed well, their limitations in capturing these relationships might have hindered their performance for some states.

CDC Analysis. We compared our best A3T-GCN models with the models submitted to CDC for the states. The evaluation of these models was based on the sMAPE definition, and Table 4 shows the average sMAPE score for all states, excluding Virginia and Pennsylvania because the data for those two states



(a) sMAPE results



(b) MAE results

Fig. 5. Average of prediction results of the best A3T-GCN and the baselines

was unavailable. We limited our selection to only those models with matching forecasting dates. A3T-GCN ranked second overall, outperforming most other models. In addition, our model demonstrated superior performance for Ohio, as depicted in Fig. 6.

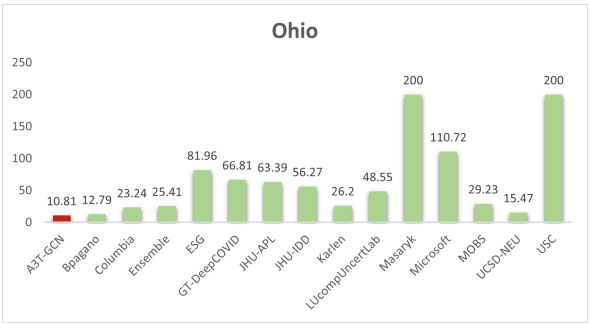


Fig. 6. A3T-GCN comparison with CDC models for Ohio using sMAPE

5 Conclusion and Future Work

Our work highlights the importance of accurately capturing both spatial and temporal dependencies in COVID-19 forecasting. The MI and correlation-based

Table 4. Comparison of A3T-GCN with the models forecast submitted to CDC using sMAPE

Model	sMAPE	Model Method
JHU-IDD [24]	82.67	Metapopulation Susceptible-Exposed-Infected-Recovered (SEIR) model
LUcompUncertLab [25]	66.33	A Bayesian Vector Auto Regression model
Microsoft [26]	53.1	SEIR model on a spatiotemporal network
USC [27]	48.82	Discrete heterogeneous rate model
ESG [28]	36.04	Fitting reported data to multiple skewed gaussian distributions
Masaryk	35.59	ARIMA models with outlier detection applied to transformed series
UCSD-NEU [29]	26.54	Age-structured metapopulation model with deep learning
JHU-APL [30]	25.73	Metapopulation SEIR model
Karlen [31]	23.33	Discrete time difference equations
MOBS [32]	21.3	Metapopulation, age-structured Susceptible-Latent-Infected-Removed (SLIR) model
GT-DeepCOVID [33]	20.22	Deep Learning
BPagano [34]	15.97	Susceptible-Infected-Recovered (SIR) model
Ensemble [35]	14.28	Combination of several forecasts
A3T-GCN	**13.82**	Our Model
Columbia [36]	10.51	Metapopulation SEIR model

A3T-GCN model proposed in this work addresses this challenge by leveraging the attention mechanism to focus on the most informative data that includes both recent and distant data points to capture temporal dependencies and correlation and MI to capture the spatial dependencies effectively. We found utilizing MI to be the most effective technique to create the graph that the model uses for the US state's weekly mortality predictions. Our model outperforms the existing baselines and fairly competes with the models adopted by CDC.

Future work can explore various association techniques for COVID-19 forecasting and extend the study to other diseases with complex spatiotemporal dynamics. Further research can focus on separate state-specific models, considering inherent differences and inter-state impacts. Overall, this work has the potential to provide accurate COVID-19 forecasting for spatiotemporal data, aiding in implementing effective measures to control the virus's spread.

References

1. CDC Covid Tracker: https://covid.cdc.gov/covid-data-tracker/#forecasting
2. Chimmula, V.K., Reddy, L.Z.: Time series forecasting of COVID-19 transmission in Canada using LSTM networks. Chaos Solitons Fractals **135**, 109864 (2020)
3. Kumar, S., et al.: Forecasting the spread of COVID-19 using LSTM network. BMC Bioinformatics **22**(6), 1–9 (2021)

4. Pavan, K., et al.: Forecasting the dynamics of COVID-19 pandemic in top 15 countries in April 2020: ARIMA model with machine learning approach. MedRxiv (2020)
5. Chinazzi, M., et al.: The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. Science **368**(6489), 395–400 (2020)
6. Ferguson, N.M., et al.: Strategies for mitigating an influenza pandemic. Nature **442**(7101), 448–452 (2006)
7. Kapoor, A., et al.: Examining covid-19 forecasting using spatio-temporal graph neural networks. arXiv preprint arXiv:2007.03113 (2020)
8. Mahmud, S., et al.: A human mobility data driven hybrid GNN+ RNN based model for epidemic prediction. In: 2021 IEEE International Conference on Big Data (Big Data). IEEE (2021)
9. Panagopoulos, G., Nikolentzos, G., Vazirgiannis, M.: Transfer graph neural networks for pandemic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 6. (2021)
10. Rozemberczki, B., et al.: Pytorch geometric temporal: spatiotemporal signal processing with neural machine learning models. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management (2021)
11. Wang, Lijing, et al.: Using mobility data to understand and forecast COVID19 dynamics. medRxiv (2020)
12. Xue, J., et al.: Multiwave covid-19 prediction from social awareness using web search and mobility data. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2022)
13. Cramer, E.Y., et al.: Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the United States. Proc. Natl. Acad. Sci. **119**(15), e2113561119 (2022)
14. Fritz, C., Dorigatti, E., Rögamer, D.: Combining graph neural networks and spatio-temporal disease models to predict covid-19 cases in Germany. arXiv preprint arXiv:2101.00661 (2021)
15. Cao, D., et al.: Spectral temporal graph neural network for multivariate time-series forecasting. Adv. Neural. Inf. Process. Syst. **33**, 17766–17778 (2020)
16. Bai, J., et al.: A3t-GCN: attention temporal graph convolutional network for traffic forecasting. ISPRS Int. J. Geo Inf. **10**(7), 485 (2021)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in neural Information Processing Systems, vol. 30 (2017)
19. Learned-Miller, E.G.: Entropy and mutual information, p. 4. University of Massachusetts, Amherst, Department of Computer Science (2013)
20. Dong, E., Hongru, D., Gardner, L.: An interactive web-based dashboard to track COVID-19 in real time. Lancet. Infect. Dis **20**(5), 533–534 (2020)
21. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
22. Smyl, S., Ranganathan, J., Pasqua, A.: M4 forecasting competition: introducing a new hybrid ES-RNN model (2018). https://eng.uber.com/m4-forecasting-competition
23. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput. Sci. **7**, e623 (2021)
24. JHU-IDD by Johns Hopkins University: Infectious Disease Dynamic Lab. https://github.com/HopkinsIDD/COVIDScenarioPipeline/

25. Computational Uncertainty Lab by Prof. Thomas McAndrew. https://zoltardata.com/model/737
26. Microsoft by Microsoft AI. https://www.microsoft.com/en-us/ai/ai-for-health/
27. Srivastava, A.: The Variations of SIkJalpha Model for COVID-19 Forecasting and Scenario Projections. arXiv preprint arXiv:2207.02919 (2022)
28. ESG by Robert Walraven. https://rwalraven.com/COVID19/
29. UCSD-NEU University of California, San Diego and Northeastern University. https://sites.google.com/view/yianma/epidemiology/
30. JHU-APL by Johns Hopkins University, Applied Physics Lab. https://buckymodel.com/
31. Karlen by Karlen Working Group. https://pypm.github.io/home/
32. MOBS by Northeastern University, Laboratory for the Modeling of Biological and Socio-technical Systems. https://covid19.gleamproject.org/
33. GT-DeepCOVID by Georgia Institute of Technology, College of Computing. https://deepcovid.github.io/
34. BPagano by Bob Pagano. https://bobpagano.com/
35. Ray, E.L., et al.: Ensemble forecasts of coronavirus disease 2019 (COVID-19) in the US. MedRXiv (2020)
36. Columbia by Columbia University. https://columbia.maps.arcgis.com/apps/webappviewer/index.html?id=ade6ba85450c4325a12a5b9c09ba796c