


## MULTI-STEP FORECASTING OF COVID-19 CASES IN EUROPEAN COUNTRIES USING TEMPORAL CONVOLUTIONAL NETWORKS

Osman Tayfun BİŞKİN\*, Department of Electrical-Electronics Engineering, Burdur Mehmet Akif Ersoy University, Turkey, tbiskin@mehmetakif.edu.tr

( <https://orcid.org/0000-0002-2326-9438>)

Received: 05.02.2021, Accepted: 31.03.2021

Research Article

\*Corresponding author

DOI: 10.22531/muglajsci.875414

### Abstract

The novel Coronavirus (COVID-19) has significantly affected millions of people around the world since the first notification until nowadays. The rapid spread of the virus has dramatically increased the workload of healthcare systems in many countries. Therefore, the need for efficient use of the healthcare system leads researchers to forecast the trend of virus spread. For this purpose, Machine Learning (ML) and Artificial Intelligence (AI) applications have intensively used to struggle against the coronavirus outbreak. In this study, Temporal Convolutional Network (TCN) is applied for modeling the cumulative confirmed COVID-19 cases and forecasting the spread of it in various European countries using time series data. It is also presented that numerical examples for comparing performances of TCN against Long-Short Term Memory (LSTM) and Gates Recurrent Units (GRU) in terms of computation time, root-mean-square error (RMSE), normalized root-mean-square error (NRMSE), root mean squared log error (RMSLE), mean absolute percentage error (MAPE), and symmetric mean absolute percentage error (SMAPE). Simulation results indicate that the Temporal Convolutional Networks used in this manuscript performs better than other models for forecasting the cumulative confirmed COVID-19 cases.

**Keywords:** COVID-19, Modeling, Forecasting, Machine Learning, Deep Learning, GRU, LSTM, TCN

## ZAMANSAL EVRİŞİMLİ AĞLAR KULLANILARAK AVRUPA ÜLKELERİNDEKİ COVID-19 VAKALARININ ÇOK ADIMLI TAHMİNİ

### Özet

Yeni tip Koronavirüs (COVID-19), ilk bildirimden bu günlere kadar dünya çapında milyonlarca insanı önemli ölçüde etkilemiştir. Virüsün hızla yayılması, birçok ülkede sağlık hizmetlerinin iş yükünü önemli ölçüde artırmıştır. Bu nedenle, sağlık sisteminin verimli kullanımına duyulan ihtiyaç, araştırmacıları virüsün yayılma eğilimini tahmin etmeye yönlendirmiştir. Bu amaçla, makine öğrenimi (ML) ve yapay zeka (AI) uygulamaları, koronavirüs salgınına karşı mücadele etmek için yoğun bir şekilde kullanılmıştır. Bu çalışmada, kümülatif doğrulanmış COVID-19 vakalarını modellemek ve zaman serisi verilerini kullanarak çeşitli Avrupa ülkelerinde yayılmasını tahmin etmek için Zamansal Evrişimli Ağları kullanılmıştır. Ayrıca Zamansal Evrişimli Ağların (TCN) performanslarını, hesaplama süresi, kök ortalama kare hatası (RMSE), normalleştirilmiş kök ortalama kare hatası (NRMSE), kök ortalama kare günlük hatası (RMSLE), ortalama mutlak yüzde hatası (MAPE) ve simetrik ortalama mutlak yüzde hatası (SMAPE) cinsinden literatürdeki diğer modeller olan Uzun-Kısa Süreli Bellek (LSTM) ve Kapılı Yinelemeli Birim (GRU) ile karşılaştırdık. Bu çalışmada kullandığımız Benzetim sonuçlarında, bu makalede kullandığımız Zamansal Evrişimli Ağların performansının COVID-19 vakalarının tahmininde diğer modellere göre daha iyi olduğu gösterilmiştir.

**Anahtar Kelimeler:** COVID-19, Modelleme, Tahmin, Makine Öğrenmesi, Derin Öğrenme, GRU, LSTM, TCN

### Cite

Bişkin, O. T., (2021). "Multi-Step Forecasting of Covid-19 Cases in European Countries Using Temporal Convolutional Networks", *Mugla Journal of Science and Technology*, 7(1), 117-126.

### 1. Introduction

On December 31, 2019, 27 cases of pneumonia of unknown etiology were recognized in Wuhan City in China [1, 2]. In the following days, the virus was named Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), and the virus-related disease was named COVID-19 by World Health Organization (WHO). The

virus has rapidly spread around the world since the first notification and WHO characterized COVID-19 as a pandemic on March 11, 2020 [3].

The spreading of the virus among a huge amount of people dramatically increased the workload of healthcare systems for many countries. Therefore, the need for optimizing the available resources leads

researchers to forecast the trend of virus spread. In order to understand the pattern of virus spread, researchers have proposed some models using time series data. For this purpose, mathematical, Machine Learning (ML), and Artificial Intelligence (AI) based applications have been intensively used to struggle against coronavirus outbreak.

There already exist studies in the literature for modeling and forecasting of COVID-19 using time-series data [4-9]. In this manuscript, the same problem is also addressed by employing an alternative strategy.

As a consequence of significant interest in the forecasting of COVID-19 spread, this topic has already been addressed in some recent studies. These studies can be categorized into two groups: Mathematical based approaches and ML and AI-based approaches. In [10], a mathematical method was proposed for modeling and forecasting the COVID-19 pandemic in India. In [11], employing the ensemble Kalman filter was introduced as a short-term predictor for pandemic spread. The forecasting method with the Logistic and Prophet model was proposed to analyze the COVID-19 in [12]. In [13], a genetic algorithm was also applied for time series analyzing and forecasting the virus pandemic in India. On the other hand, a wide variety of ML and DL based methods have been proposed for modeling and forecasting of COVID-19 as a result of growing interest in ML and AI issues in recent years. In [14], linear regression (LR), support vector machines (SVM), and exponential smoothing methods are used for forecasting the number of confirmed and death cases. One of the most popular methods used for time-series forecasting problem is the Auto-Regressive Integrated Moving Average (ARIMA) model. ARIMA models have been employed to predict many diseases such as brucellosis [15] and influenza [16]. Lastly, ARIMA was utilized to predict the confirmed cases of COVID-19 in [4, 17]. This method was also used in many comparative analyses to compare the performance of ARIMA against other ML-based approaches such as the Nonlinear Autoregression Neural Network (NARNN) and Long-Short Term Memory (LSTM) [8]. A deep learning method, Gated Recurrent Units (GRU) was used for COVID-19 epidemic prediction in [5, 7]. One of the other deep learning methods widely used in time series forecasting, LSTM, is a gate memory unit designed to overcome the vanishing gradient problems decreasing the efficiency of Recurrent Neural Networks (RNN) [5]. LSTM was used for the time series forecasting of COVID-19 transmission in Canada [6] and prediction for the spread of the virus in India [18]. Moreover, there exist many studies comparing the prediction performance of LSTM against RNN [5], GRU [5, 7], and ARIMA and NARNN [8]. These studies showed that the LSTM achieved satisfactory forecasting performance by means of its capacity in handling time-series data [5]. Although the LSTM model is extensively utilized in time series forecasting problems, it is time-consuming and non-robust for aperiodic data forecasting [19].

Recently, Temporal Convolution Network (TCN) is proposed for modeling time series data. TCN has been applied for multivariate time series forecasting [19], wind speed forecasting [20], forecasting electricity demand [21], classification of satellite image time series [22], short-term traffic speed forecasting [23], action segmentation [24], and prediction of sepsis [25]. In the latest researches, [21, 26], LSTM and TCN models are compared and as indicated in those researches that the TCN outperforms the LSTM model. In addition, TCN shows better performance especially in processing very long sequences of inputs [26]. Because of this reason, there exists a significant interest in using TCN for time series forecasting. In this study, employing the TCN is the proposed approach for multi-step forecasting of COVID-19 cases in various European countries using time series data. Although there is a growing interest in TCN and it has been increasingly employed for time-series data last years, to the best of our knowledge, this is the first time TCN is used for forecasting of COVID-19 cases or any epidemiological modeling.

The rest of the manuscript is organized as follows. In Section 2, after the problem statement is given, the LSTM, GRU, and TCN models are briefly introduced. Besides, performance metrics are also formulated in this section. In Section 3, some numerical experiments are provided and simulation results are given. Finally, in Section 4, after a brief summary, our conclusions is presented.

## 2. Methods and Models

### 2.1. Problem Statement

In this manuscript, the main motivation is the forecasting of COVID-19 cases employing deep learning-based methods. Forecasting is a process of making a prediction of future values using historical values of a dataset. Let a sequence  $\mathbf{x} = [x_1 \dots x_T]$  denote time series data with  $x_t \in \mathbb{R}^d$  where  $d$  represents the number of dimensions of time-series data,  $x_t$ , at time  $t$ . Then, our purpose is to predict the corresponding future values  $\mathbf{x} = [x_T \dots x_{T+k}]$ . Here,  $k$  represents the number of future values to be predicted by utilizing past values. Thus, the aim of the forecasting problem can be formulated as finding a nonlinear mapping function,  $f$ , such that  $[x_T \dots x_{T+k}] = f([x_1 \dots x_T])$ . Various deep learning models have been proposed for general time-series forecasting problems such as LSTM, GRU, and TCN. In this manuscript, employing TCN models is proposed for forecasting of COVID-19 cases and compare the performance of TCN against LSTM and GRU in terms of performance metrics.

### 2.2. LSTM Models

RNN has been used for time series applications having temporal dependencies [7]. However, vanishing gradient problem limits the capability of RNN networks

for long term dependencies. Therefore, LSTM, a gate memory unit, was proposed in order to tackle the vanishing gradient problem [27]. Information flow throughout the memory is controlled by three gates, input, output, and forget gates [28]. Input gates control the flow of input activation information and output gates control the flow of output activation information, respectively. On the other hand, since cell state is controlled via input and forget gates [5], when the cell memory is not needed anymore, forget gates reset the memory of the cell [28]. Fig. 1 shows the internal architecture of the LSTM unit. Let  $\mathbf{x}_t$  denote the input time-series data at  $t$ , then input ( $i_t$ ), output ( $o_t$ ), and forget gates ( $f_t$ ) can be given as

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \end{aligned} \quad (1)$$

where  $W_i$ ,  $W_o$ , and  $W_f$  are input, output, and forget weight parameters, respectively. In addition,  $b_i$ ,  $b_o$ , and  $b_f$  represent bias parameters in (1).  $h_t$  denote the hidden state vector (also known as output state) at the time step  $t$ . In (1),  $\sigma(\cdot)$  is given as the sigmoid function. The hidden state ( $h_t$ ) and cell state ( $c_t$ ) at the time step  $t$  is written as follows

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \quad (2)$$

$$h_t = o_t \odot \tanh(c_t), \quad (3)$$

where  $\odot$  represents Hadamard product and  $\tanh(\cdot)$  is the hyperbolic tangent function.  $\tilde{c}_t$  in (2) is given as

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c). \quad (4)$$

Here,  $W_c$  and  $b_c$  denote weight and bias parameters, respectively.

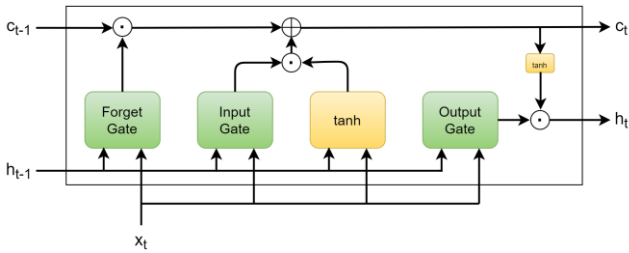


Figure 1. The architecture of the LSTM unit

### 2.3. GRU Models

GRU was proposed as a variant of LSTM in order to reduce the complexity of LSTM architecture and to decrease the parameters of it [5, 7, 29, 30]. GRU has two gates: update and reset gates. The update gate in GRU is a combination of input and forget gates in LSTM. The update gate provides to control how much of the memory is transferred to the new state. The function of the reset gate is similar to the forget gate in LSTM and it is responsible to forget the past information being not used anymore. The architecture of the GRU unit is given

in Fig. 2. The update ( $u_t$ ) and reset gates ( $r_t$ ) in GRU can be written as

$$u_t = \sigma(W_u[h_{t-1}, x_t] + b_u) \quad (5)$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (6)$$

Here  $W_u$  and  $W_r$  represent update and reset weight parameters, respectively.  $b_u$  and  $b_r$  given in (5) and (6) are bias parameters.  $h_t$  denote the hidden state vector (also known as output state vector)

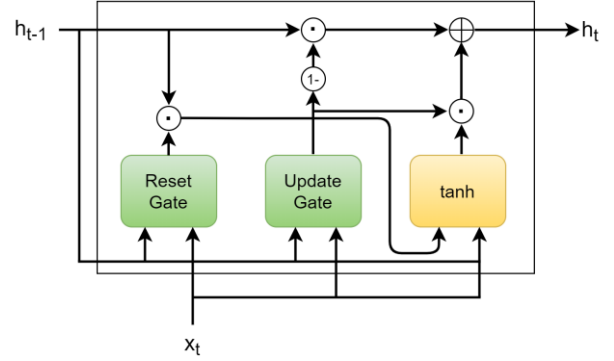


Figure 2. The architecture of the GRU unit

at the time step  $t$ . The hidden state ( $h_t$ ) at the time step  $t$  is expressed as

$$h_t = u_t \odot \tilde{h}_t + (1 - u_t) \odot c_{t-1}, \quad (7)$$

where  $\tilde{h}_t$  in (7) is given as follows

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h), \quad (8)$$

where  $W_h$  and  $b_h$  denote weight and bias parameters, respectively.

### 2.4. TCN Models

In [31], TCNs are proposed for speech processing tasks. These models are able to handle long-range patterns using a hierarchy of temporal convolutional filters [32]. TCN uses causal and dilated convolution layers. Causal convolution does not violate the ordering because only present and past values are used for making predictions. That is, let  $\mathbf{x} = [x_1 \dots x_T]$  denote time series data, prediction making by the model at time  $t$  does not depend on future time steps  $x_{t+1}, x_{t+2}, \dots, x_T$ . In addition, since causal convolutions have not a recurrent connection, training TCN is faster than training RNNs especially for very long sequences [31]. On the other hand, dilated convolution in TCN models allows us to increase the receptive field. Although this might be considered to be similar to pooling or strided convolutions, in dilated convolutions input and output size are the same [31].

Let  $w: \{0, \dots, k-1\} \rightarrow \mathbb{R}$  denote a kernel, then the dilated causal convolution of  $\mathbf{x}$  with the kernel is defined as [33]

$$F(\mathbf{x}) = (\mathbf{x} \otimes_d w)(t) = \sum_{i=0}^{k-1} w(i) \mathbf{x}(t - id) \quad (9)$$

where  $\otimes_d$  represents dilated convolution operator,  $k$  is the kernel size, and  $d$  denotes dilation factor.

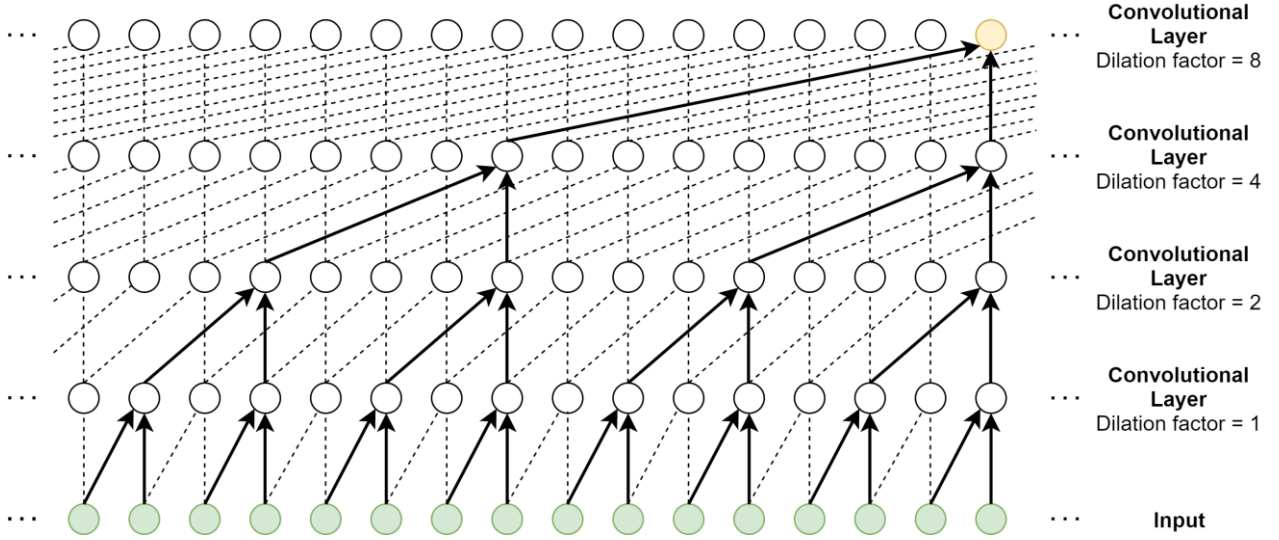


Figure 3. The architecture of the TCN network.

If the dilation factor,  $d$ , is chosen as  $d = 1$ , then the dilated causal convolution given in (9) becomes regular causal convolution as

$$F(\mathbf{x}) = (\mathbf{x} \otimes w)(t) = \sum_{i=0}^{k-1} w(i)\mathbf{x}(t-i) \quad (10)$$

The dilation factor,  $d$ , in dilated convolution given in (9) can be increased in order to accomplish a large receptive field. Fig 3 shows the TCN network with the dilation factor growing exponentially such that the  $d = 2^{k-1}$  at the  $k^{\text{th}}$  convolutional layer. Let  $f$  denote the filter size in TCN and  $L$  represents the number of convolutional layers. Then, the receptive field size can be written as  $R = (f-1)(2^L-1) + 1$ . Notice that the receptive field size increase with a growing dilation factor in convolutional layers.

### 2.5. Performance Metrics

Forecasting performance of TCN model against LSTM and GRU are evaluated by utilizing performance metrics such as Root-Mean-Square Error (RMSE), Normalized Root-Mean-Square Error (NRMSE), Root Mean Squared Log Error (RMSLE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE). Let a vector  $\mathbf{y}$  be the future values of the sequence  $\mathbf{x}$  such that  $\mathbf{y} = \tilde{\mathbf{x}} = [x_T \dots x_{T+k}]$ . Then,

$y_i$  and  $\hat{y}_i$  denote actual and predicted values, respectively. Thus, the aforementioned performance metrics can be formulated as

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2} \quad (11)$$

$$\text{NRMSE} = \frac{\text{RMSE}}{y_{\max} - y_{\min}} \quad (12)$$

$$\text{RMSLE} = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (\log(y_i) - \log(\hat{y}_i))^2} \quad (13)$$

$$\text{MAPE} = \frac{100}{n} \sum_{i=0}^{n-1} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (14)$$

$$\text{SMAPE} = \frac{200}{n} \sum_{i=0}^{n-1} \left| \frac{y_i - \hat{y}_i}{y_i + \hat{y}_i} \right| \quad (15)$$

In (12),  $y_{\max}$  and  $y_{\min}$  indicate the maximum and minimum element in the vector  $\mathbf{y}$ , respectively. On the other hand,  $\log(\cdot)$  represents the natural logarithm operator in (13). RMSLE metric given in (13) is extensively utilized for evaluating model performance in regression problems because of its robustness to outliers [5].

### 3. Results and Discussion

In this manuscript, forecasting cumulative confirmed COVID-19 cases in 8 European countries, Belgium, France, Germany, Italy, Portugal, Spain, Switzerland, and the United Kingdom, is accomplished by employing the TCN model for time series data. European countries analyzed in this study are listed in Table 1. Table 1 also indicates the total confirmed cases in those countries. In addition, minimum (min), maximum (max), mean, median, standard deviation (Std), skew, and kurtosis values of data are given in Table 1. In our simulations, 7-day ahead forecasting of new cases is provided using historical data of confirmed cases. The data used in this manuscript were obtained from the European Center for Disease Prevention and Control presented in [34]. In order to predict the future number of cumulative confirmed cases in the countries, the last 6 months before December 8, 2020 were taken into consideration.



Table 1. Summary of COVID-19 dataset for 7 countries

Country	Cumulative Confirmed Cases (December 8, 2020)	Min	Max	Mean	Median	Std	Skew	Kurtosis
Belgium	532,226	1	22,210	2,956.81	748.50	4704.89	2.22	4.28
France	2,140,772	0	86,852	11,893.18	6,351	14,871.11	1.88	4.00
Germany	1,012,293	159	23,648	5,623.85	1,466	7,356.84	1.26	0.09
Italy	1,506,794	113	40,902	8,371.08	1,419.50	12,138.98	1.28	0.10
Portugal	289,471	0	8,371	1,608.17	447	2,014.30	1.43	0.87
Spain	1,460,048	0	55,019	8,111.38	3860	10,473.78	1.88	4.46
Switzerland	321,947	0	21,842	1,788.59	201	3,530.72	2.77	8.89
United Kingdom	1,470,943	70	33,470	8,171.91	2789	8,966.02	0.77	-0.93

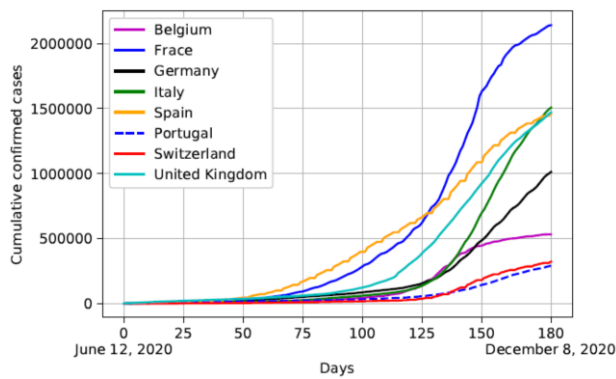


Figure 4. Cumulative confirmed COVID-19 cases in European countries

In Fig. 4, the number of cumulative confirmed cases is demonstrated from June 12, 2020 to December 8, 2020.

All the simulations are performed on a PC employing i7-9750H CPU with a memory of 16 GB and a processor speed of 2.6 GHz. NVIDIA GTX 1660 Ti GPU with a memory of 6 GB is also utilized in order to speed up the model training. To train networks on GPU, NVIDIA CUDA 10.1 Toolkit is employed with cuDNN v7.6.5 library (NVIDIA CUDA Deep Neural Network library) for 64 bit Windows 10 operating system. Tensorflow v2.3.0 and Keras v2.4.3 libraries are also used to be compatible with the mentioned version of cuDNN library. In the experiments performed, a three-layer network is used for all models, LSTM, GRU, and TCN models. These networks include two layers, each of which has 64 hidden units, and a dense layer. On the other hand, dilation factors for TCN layers are selected as  $d = 1, 2$  at the first and the second layers, respectively. For all networks, mean squared error is selected as the loss function which is used to estimate the loss of the model during the training process. The loss function is minimized during the training using Adam [35] optimizer and the learning rate of the optimizer is taken as  $lr = 0.00001$ .

The time step is assigned as 3 for all employed models. The maximum epoch number is set to 5000 but the early stopping method is utilized during training in order to avoid overfitting problems. Fig. 4 shows the cumulative confirmed COVID-19 cases in various European countries from June 12, 2020 to December 8, 2020. It can be seen that the dramatic increase in COVID-19 cases starts from the beginning of autumn.

In the first experiments, a 7-day ahead forecast of confirmed cases is made by utilizing the GRU, LSTM, and TCN models and evaluate the performances of them for European countries. Fig. 5 indicates the actual and predicted cumulative confirmed cases for Belgium, France, Germany, Italy, Portugal, Spain, Switzerland, and the United Kingdom. The prediction results are plotted for both train and test data. The results of the 7-day ahead forecast for test data are also demonstrated in Fig 5. Looking at Fig. 5, it can be seen that the TCN model can fit the actual data better than the LSTM and GRU models for all countries. On the other hand, LSTM and GRU networks demonstrate similar performance for these test data. However, the GRU model performs better than the LSTM for the prediction of confirmed cases.

In Fig. 6, in order to compare the performance of the TCN model against GRU and LSTM, the forecasting performance of modes in terms of performance metrics RMSE, NRMSE, RMSLE, MAPE, and SMAPE, are plotted together with all simulated countries. As can be seen from Fig. 6, 7-day ahead forecasting made by utilizing TCN models outperforms GRU and LSTM in terms of the aforementioned metrics. Besides, for 8 European countries, the variation in the performance of the TCN model is lower compared to other models. Thus, TCN provides more stable forecast results than GRU and LSTM models. Figure 7 shows the prediction error of confirmed cases for 7 days in different countries. Not only prediction error of the TCN model is lower than other models, but also the variation in the error of TCN is smaller than others.

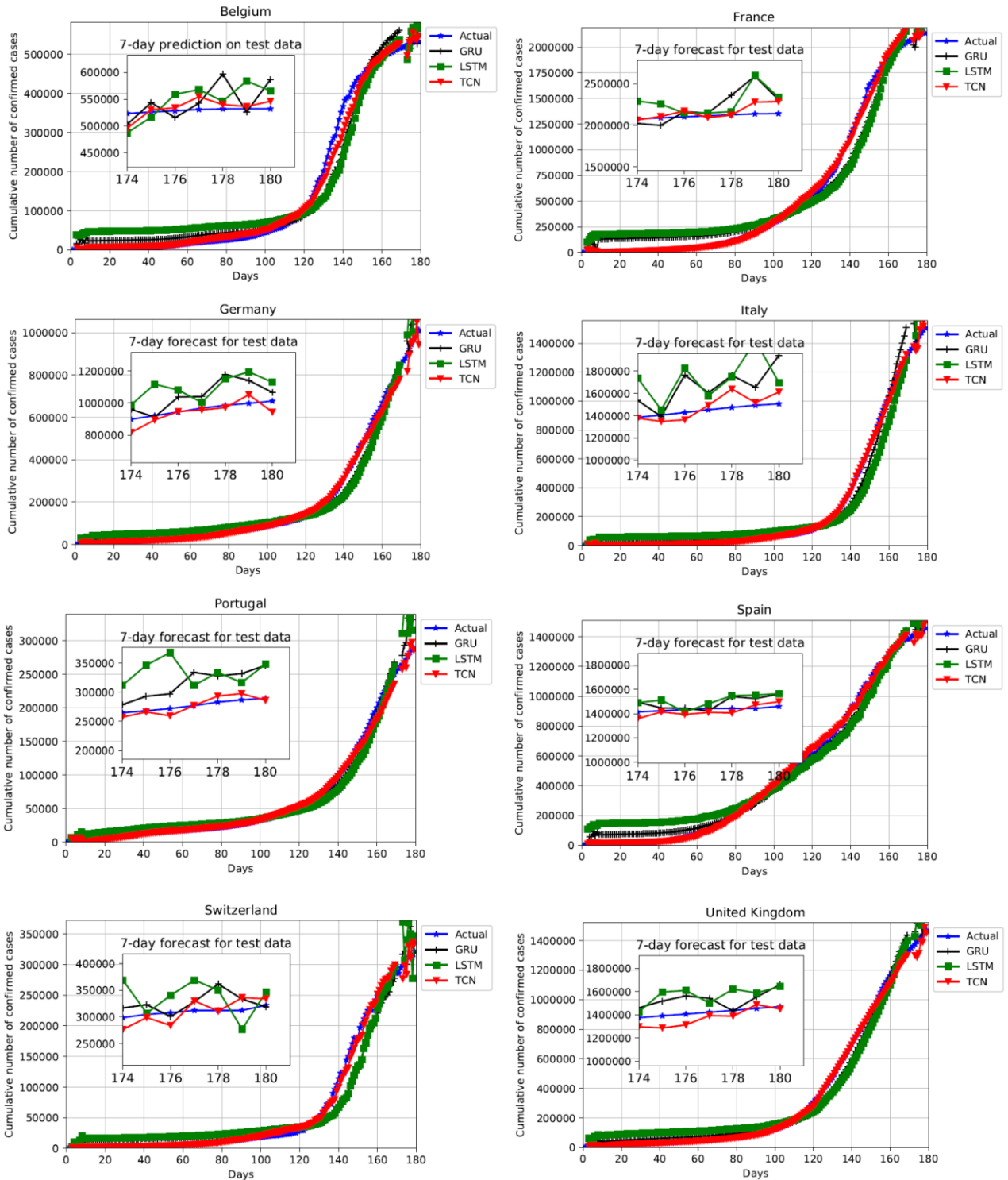


Figure 5. Actual and predicted cumulative confirmed COVID-19 cases for (a) Belgium, (b) France, (c) Germany, (d) Italy, (e) Portugal, (f) Spain, (g) Switzerland, and (h) United Kingdom by employing GRU, LSTM, and TCN.

The numerical results of the experiment for LSTM, GRU, and TCN, are presented in Table 2. It can be seen from Table 2 that the TCN model achieves lower RMSE, NRMSE, RMSLE, MAPE, and SMAPE values than GRU and LSTM models. Although the LSTM model performs slightly better than the GRU in Belgium, the GRU model achieves lower values in terms of performance metrics in the other countries. GRU model showed its worst

forecasting performance in terms of RMSE, RMSLE, MAPE, and SMAPE in Italy and in terms of NRMSE in Belgium. On the contrary, the best performance of GRU model in terms of RMSLE, MAPE, and SMAPE is seen in Spain. On the other hand, the worst performance of LSTM in terms of MAPE and SMAPE occurs in Portugal. It can be seen that the LSTM model reaches its highest RMSE and RMSLE values in Italy.

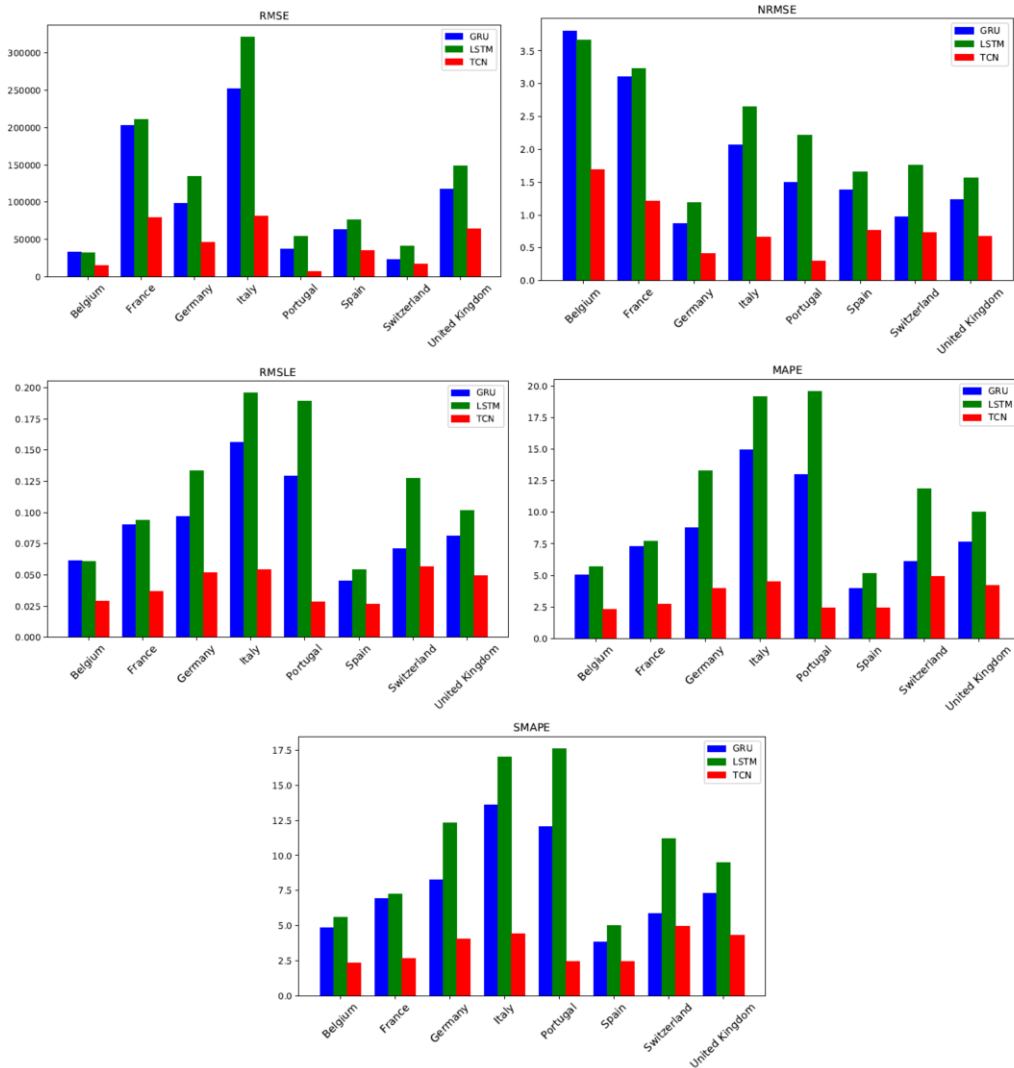


Figure 6. Performances of TCN, GRU, and LSTM for the countries in terms of (a) RMSE (b) NRMSE, (c) RMSLE, (d) MAPE, and (f) SMAPE.

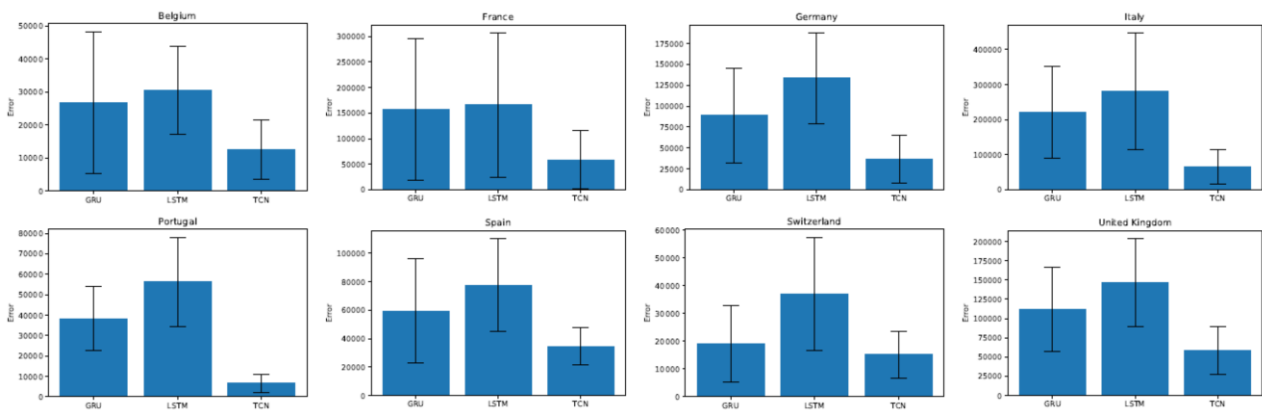


Figure 7. 7-day ahead forecast error for (a) Belgium, (b) France, (c) Germany, (d) Italy, (e) Portugal, (f) Spain, (g) Switzerland, and (h) the United Kingdom.

TCN model indicates its worst performance in terms of MAPE and SMAPE in Switzerland and in terms of RMSE in Italy. TCN model achieves the lowest RMSE, RMSLE, MAPE, and SMAPE values in Belgium and the lowest NRMSE value in Portugal. In Fig. 8, the evolution of training and validation loss of models are given for the

best GRU, LSTM, and TCN models. These figures demonstrate the learning process of models and indicate the speed of learning for different architectures. As can be seen from Fig 8, the TCN model outperforms the other models in terms of the number of epoch.

Table 2. Numerical results of the experiment

Country	Model	RMSE	NRMSE	RMSLE	MAPE	SMAPE
Belgium	GRU	33,138.0682	3.8011	0.0613	5.0370	4.8737
	LSTM	31,956.1134	3.6655	0.0607	5.6925	5.5805
	TCN	14,754.0509	1.6924	0.0289	2.3179	2.3167
France	GRU	203,164.9152	3.1095	0.0900	7.3146	6.9168
	LSTM	211,422.7953	3.2359	0.0940	7.7271	7.2511
	TCN	79,077.1019	1.2103	0.0367	2.7268	2.6594
Germany	GRU	98,030.8401	0.8678	0.0966	8.7716	8.2750
	LSTM	134,263.4196	1.1885	0.1333	13.3047	12.3450
	TCN	46,102.5151	0.4081	0.0517	3.9634	4.0595
Italy	GRU	251,736.1847	2.0692	0.1561	14.9377	13.5951
	LSTM	321,707.6871	2.6444	0.1956	19.1924	17.0254
	TCN	81,213.0915	0.6676	0.0540	4.4884	4.3947
Portugal	GRU	36,929.1794	1.5006	0.1293	12.9685	12.0689
	LSTM	54,619.0960	2.2195	0.1891	19.5804	17.5824
	TCN	7,301.0286	0.2967	0.0284	2.4383	2.4516
Spain	GRU	63,491.1193	1.3837	0.0451	3.9464	3.8455
	LSTM	76,199.3194	1.6607	0.0540	5.1568	5.0099
	TCN	35,309.8042	0.7695	0.0262	2.4225	2.4433
Switzerland	GRU	22,766.5155	0.9752	0.0707	6.1029	5.8503
	LSTM	41,156.4894	1.7630	0.1276	11.8753	11.2055
	TCN	17,029.8943	0.7295	0.0568	4.9301	4.9511
United Kingdom	GRU	117,601.3650	1.2396	0.0809	7.6716	7.3289
	LSTM	148,993.2782	1.5704	0.1017	10.0197	9.4717
	TCN	64,540.4339	0.0491	4.1900	4.3009	4.3009

Moreover, the computation times of GRU, LSTM, and TCN models are presented in Table 3. To obtain the computation time values in Table 3, Monte Carlo trials were performed. In measuring the computation time for all models, averages of 10 independent Monte Carlo runs were calculated and given for each country. Training time is given in terms of second and it represents the time which the training of a model took. On the other hand, prediction time in Table 3 is given in terms of seconds and corresponds to the time which a prediction made by a model took. One can see in Table 3 that the TCN model performs better than GRU and LSTM models by terminating in less computation time.

#### 4. Conclusion

In this manuscript, TCN models have been employed for Belgium, France, Germany, Italy, Portugal, Spain, Switzerland, and the United Kingdom. Multi-step time series forecasting is accomplished by employing the

TCN model and the performance of models is compared against LSTM and GRU methods in terms of RMSE, NRMSE, RMSLE, MAPE, and SMAPE metrics. Multi-step time series forecasting is accomplished by employing the TCN model and the performance of models is compared against LSTM and GRU methods in terms of RMSE, NRMSE, RMSLE, MAPE, and SMAPE metrics. Besides, models are compared in terms of computation time for both training and prediction processes. Numerical experiments indicate that the TCN model outperforms the other methods in terms of all performance metrics used in the manuscript and computation time. Furthermore, variation in the performance of the proposed model is lower compared to other models. In conclusion, we believe that the proposed TCN model can reliably be employed for forecasting of confirmed COVID-19 cases in European countries and also it can be applied to other countries around the world.



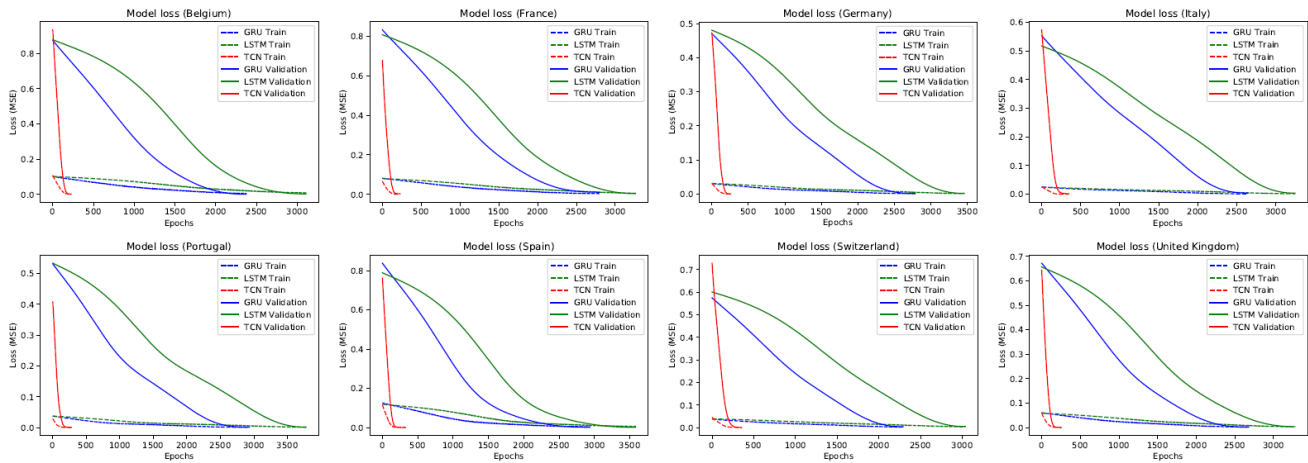


Figure 8. Training and validation losses of models for (a) Belgium,(b) France, (c) Germany, (d) Italy, (e) Portugal, (f) Spain, (g) Switzerland, and (h) the United Kingdom.

Table 3. The computation time of GRU, LSTM, and TCN models.

Country	Training Time (sec.)			Prediction Time (sec.)		
	GRU	LSTM	TCN	GRU	LSTM	TCN
Belgium	50.5844	68.4563	5.5844	0.0348	0.0322	0.0312
France	54.0156	70.4219	5.7250	0.0352	0.0315	0.0301
Germany	58.4344	75.2656	6.3938	0.0321	0.0315	0.0305
Italy	58.6313	76.5625	7.8875	0.0328	0.0318	0.0305
Portugal	70.1688	85.7906	7.5156	0.0316	0.0306	0.0301
Spain	75.5656	83.3062	7.0781	0.0314	0.0311	0.0304
Switzerland	55.4531	80.2938	7.4406	0.0318	0.0310	0.0299
United Kingdom	58.7156	78.1219	6.2250	0.0308	0.0317	0.0308

### 5. References

- [1] Lu, H., Stratton, C. W. and Tang, Y.-W., "Outbreak of pneumonia of un-known etiology in wuhan china: the mystery and the miracle", *Jour.Med. Virol.* 92 (4), 401-402, 2020.
- [2] Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A, Iosifidis C. and Agha, R., "World Health Organization Declares Global Emergency: A Review of The 2019 Novel Coronavirus (Covid-19)", *Int. Jour. Surgery*, 76, 71-76, 2020.
- [3] World Health Organization, Archived: Who timeline -covid-19, Retrieved from: [https://www.who.int/news/item/27-04-2020-who-timeline---covid-19\(21/11/2020\)](https://www.who.int/news/item/27-04-2020-who-timeline---covid-19(21/11/2020)).
- [4] Ribeiro, M. H. D. M., Silva, R. G., Mariani, V. C., Coelho, L. S., "Short-Term Forecasting Covid-19 Cumulative Confirmed Cases: Perspectives for Brazil", *Chaos, Solit. and Frac.*, 135, 109853, 2020.
- [5] Zeroual, A., Harrou, F., Dairi, A. and Sun, Y., "Deep Learning Methods Forforecasting Covid-19 Time-Series Data: A Comparative Study", *Chaos,Solit. and Frac.*, 140, 110121, 2020.
- [6] Chimmula, V. K. R. and Zhang, L., "Time Series Forecasting of Covid-19 Transmission in Canada Using LSTM Networks", *Chaos, Solit. and Frac.*, 135, 109864, 2020.
- [7] F. Shahid, F., Zameer, A. and Muneeb, M., "Predictions For Covid-19 With Deep Learning Models of LSTM, GRU and Bi-LSTM", *Chaos, Solit. and Frac.*, 140, 110212, 2020.
- [8] Kırbaş, İ., Sözen, A, Tuncer, A. D. and Kazancıoğlu, F. S. "Comparative Analysis and Forecasting Of Covid-19 Cases in Various European Countries with ARIMA, NARNN and LSTM Approaches", *Chaos, Solit. and Frac.*, 138, 110015, 2020.
- [9] Arora, P., Kumar, H. and Panigrahi, B., "Prediction and Analysis of Covid-19 Positive Cases Using Deep

- Learning Models: A Descriptive Case Study of India", *Chaos, Solit. and Frac.*, 139, 110017, 2020.
- [10] Sarkar, K., Khajanchi, S. and Nieto, J., "Modeling and Forecasting the Covid-19 Pandemic in India", *Chaos, Solit. and Frac.*, 139, 110049, 2020.
- [11] Yang, Q., Yi, C., Vajdi, A., Cohnstaedt, L. W., Wu, H., Guo, X. and Scoglio, C. M., "Short-Term Forecasts and Long-Term Mitigation Evaluations for the Covid-19 Epidemic in Hubei Province, China", *Infectious Disease Modelling*, 5, 563–574, 2020.
- [12] Wang, P., Zheng, X., Li, J. and Zhu, B., "Prediction of Epidemic Trends in Covid-19 with Logistic Model and Machine Learning Technics", *Chaos, Solit. and Frac.*, 139, 110058, 2020.
- [13] Salgotra, R., Gandomi, M. and Gandomi, A., "Time Series Analysis and Forecast of the Covid-19 Pandemic in India Using Genetic Programming", *Chaos, Solit. and Frac.* 138, 109945, 2020.
- [14] Rustam, F., Reshi, A. A., Mehmood, A., Ullah, S., On, B.-W., Aslam, W. and Choi, A. G. S., "Covid-19 Future Forecasting Using Supervised Machine Learning Models", *IEEE Access*, 8, 101489 – 101499, 2020.
- [15] Cao, L., Liu, H., Li, J., Yin, X., Duan, Y. and Wang, J., "Relationship of Meteorological Factors and Human Brucellosis in Hebei Province, China", *Sci. Total Envir.*, 703, 135491, (2020).
- [16] He, Z. and Tao, H., "Epidemiology and Arima Model Of Positive-Rate of Influenza Viruses Among Children in Wuhan, China: A Nine-Year Retrospective Study", *Int. Jour. Infect. Dis.*, 74, 61–70, 2018.
- [17] Ceylan, Z., "Estimation of Covid-19 Prevalence in Italy, Spain, and France", *Sci. Total Environ.* 729(4), 1–7, 2020.
- [18] Tomar, A. and Gupta, A., "Prediction for the Spread of Covid-19 in India and Effectiveness of Preventive Measures", *Sci. Total Envir.*, 728, 138762, (2020).
- [19] Wan, R., Mei, S., Wang, J., Liu, M. and Yang, F., "Multivariate Temporal Convolutional Network: A Deep Neural Networks Approach for Multivariate Time Series Forecasting", *electronics*, 8(8):876, 1–18, 2019.
- [20] Gan, Z., Li, C., Zhou, J. and Tang, G., "Temporal Convolutional Networks Interval Prediction Model for Wind Speed Forecasting", *Electric Power Sys. Res.*, 191, 106865, 2021.
- [21] Lara-Benítez, P., Carranza-García, M., Luna-Romera, J. M. and Riquelme, J. C., "Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting", *applied sciences*, 10, 1–17, 2020.
- [22] Pelletier, C., Webb, G. I. and Petitjean, F., "Temporal Convolutional Neuralnetwork for the Classification of Satellite Image Time Series", *remotesensing*, 11(5):523, 1–22, 2019.
- [23] Guo, G. and Yuan, W., "Short-Term Traffic Speed Forecasting Based on Graph Attention Temporal Convolutional Networks", *Neurocomputing*, 410, 387–393, 2020.
- [24] Li, S., Farha, Y. A., Liu, Y., Cheng, M.-M. and Gall, J., "MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation", *IEEE Trans. Patt. Anal. Mach. Intel.*, Early Access, 1–13, 2020.
- [25] Kok, C., Jahmunah, V., Oh, S. L., Zhou, X., Gururajan, R., Tao, X., Cheong, K. H., Gururajan, R., Molinari, F. and Acharya, U. R., "Automated Prediction of Sepsis Using Temporal Convolutional Network", *Comp. Bio. Med.*, 1–10, 2020.
- [26] Yan, J., Mu, L., Wang, L., Ranjan, R. and Zomaya, A. Y., "Temporal Convolutional Networks For The Advance Prediction Of ENSO", *Scientific Reports (Nature Publisher Group)*, 10, 1–15, 2020.
- [27] Hochreiter, S. and Schmidhuber, J., "Long Short-Term Memory", *Neural Computation*, 9(8), 1735–1780, 1997.
- [28] Gers, F. A., Schmidhuber, J., Cummins, F., "Learning to Forget: Continual Prediction with LSTM", *Neural Computation*, 12(10), 2451–2471, 2000.
- [29] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", *arXiv preprint*, arXiv:1412.3555, 1–9, 2014.
- [30] Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y., "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches", *arXiv preprint*, arXiv:1409.1259, 2014.
- [31] Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K., "Wavenet: A Generative Model for Raw Audio", *arXiv preprint*, arXiv:1609.03499, 1–15, 2016.
- [32] Hewage, P., Behera, A., Trovati, M., Pereira, E., Ghahremani, M., Palmieri, F. and Liu, Y., "Temporal Convolutional Neural (TCN) Network for an Effective Weather Forecasting Using Time-Series Data from the Local Weather Station", *Methodologies and Application*, 24, 16453–16482, 2020.
- [33] Zhen, X., Chakraborty, R., Vogt, N., Bendlin, B. B. and Singh, V., "Dilated Convolutional Neural Networks For Sequential Manifold-valued Data", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10621–10631, 2019.
- [34] European Centre for Disease Prevention and Control, Geographic Distribution of Covid-19 Cases, Retrieved from: <https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide> 08/12/2020.
- [35] Kingma, D. P. and Ba, J. L., "Adam: A Method For Stochastic Optimization", *arXiv preprint*, arXiv:1412.6980, 2014.