Düzce Üniversitesi
Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Forecasting of COVID-19 Cases Under Different Precaution Strategies in Turkey

Serdar ARSLAN [a,*]

[a] *Computer Engineering Department, Çankaya University, Ankara, TÜRKİYE*
**Corresponding author: sarslan@cankaya.edu.tr*
DOI: 10.29130/dubited.1234168

ABSTRACT

The coronavirus disease started at the end of 2019 and affected all the countries in the world. In Turkey, the vaccination process started at the beginning of 2021 but performed in slow progress. Thus, the Turkish Government tried to implement precautions to control this virus's spread. In this study, we evaluated and compared five different forecasting models, ARIMA, Prophet, NARNN, Stacked LSTM, and Bidirectional LSTM, in order to show the effect of these precaution strategies on virus spread using a real-world data set. According to the test results, ARIMA and Prophet were found to be the most accurate models for small data sets that are split regarding precautions. Moreover, test results showed that when data size grows, LSTM model performance increases. However, these models' performance decreased when we fed these models by using the entire data set without splitting.

Keywords*: *COVID-19, Forecasting, ARIMA, Prophet, NARNN, Deep learning, LSTM

Türkiye'de COVID-19 Vakalarının Farklı Önlemler Altında Tahminlemesi

ÖZ

Korona virüs salgını 2019 sonunda başladı ve tüm dünyayı etkisi altına aldı. Türkiyede aşılama süreci 2021 senesini başlarında başlatıldı ama çok yavaş ilerledi. Bu yüzden, bu süreçte Türk Hükümeti virüs yayılımını engellemek için çeşitli önlemler aldı. Bu çalışmada, bu önlemlerin virüs yayılımına olan etkisini anlamak için beş farklı tahminleme modeli (ARIMA, Prophet, NARNN, Yığıt LSTM ve çiftyönlü LSTM) gerçek dünya verileri ile kullanıldı ve karşılaştırıldı. Test sonuçları önlemlere göre parçalanan veri setinde küçük olanlar için ARIMA ve Prophet'in diğer modellere göre iyi sonuçlar verdiğini gösterdi. Veri setinin büyüklüğü arttıkça derin öğrenme yöntemlerinin daha iyi sonuçlar ortaya koyduğu gözlemlendi. Fakat, önlemlere göre ayırmadan tüm veri setini tek bir seferde kullandığımızda bu modellerin performanslarının düştüğü gözlemlendi.

***Anahtar Kelimeler:** COVID-19, Tahminleme, ARIMA, Prophet, NARNN, Derin öğrenme, LSTM*

Geliş: 14/01/2023, Düzeltme: 18/09/2023, Kabul: 02/10/2023

I. INTRODUCTION

The global spread of the coronavirus epidemic accelerated at the end of December 2019[1]. The governments experimented with various strategies to mitigate the spread. On March 11, 2020, Turkey announced its first case. Over five million patients have been exposed to the virus since the outbreak. Each country takes steps to safeguard people's lives by establishing regulations like travel restrictions, quarantines, social seclusion, hard and soft locks, and delaying and canceling events. Despite these precautions, there is still a significant problem with the COVID-19 virus spreading. It is essential to evaluate the impact of such control strategies on epidemic advancement relative to global expectations in order to manage health resources and establish strategies for prevention. All of this control and preparation for preventative concerns can be assisted by model-based forecasting.

In the literature, many studies try to model and forecast the spread of COVID-19 [2]. In [3], the authors used a SIR-based model to predict the spread's impact on China. In [4], a discrete-time SIR model is proposed to predict Wuhan's mortality and recovery rates. Moreover, the effect of the different lockdown strategies in Italy is studied in [5] using the SIR model. However, SIR models must be more efficient in predicting long-term data, and more sophisticated modeling strategies should be used [6].

The Auto-Regressive Integrated Moving Average (ARIMA) technique is another statistical approach widely used to make predictions over time series data. In [7], they used this model for forecasting COVID-19 cases in various European countries. They also compared the ARIMA model with The Nonlinear Autoregression Neural Network (NARNN) and Long-Short Term Memory (LSTM) models. ARIMA is also used in [8] for Asian countries simply for the total number of cases. Similarly, [9] used ARIMA method for short-term prediction. Many studies use ARIMA as a base forecasting method in order to compare other models with this widely used approach [10] [11] [12].

Facebook developed Prophet, an open-source time-series forecasting technique [13]. Prophet is a time-series forecasting methodology that is relatively new but has gained popularity since it is simple to use while still being effective. In [14], Prophet and SIR modes are compared using COVID-19 data. Prophet is also compared with ARIMA and LSTM in [15] and performed well on the data set containing COVID-19 total cases and deaths of Turkey. The Nonlinear Autoregression Neural Network (NARNN) model uses neural network by performing nonlinear regression through it. This method is used for forecasting when the series are non-linear. In [16], NARNN model performance is compared with ARIMA and LSTM models for different countries. In [17], NARNN model is used for forecasting COVID-19 confirmed, recovered and death numbers in India. Another study that uses NARNN for data set of Egypt is proposed in [18] and the model is compared with ARIMA model.

Various deep learning techniques have been proposed for time series forecasting including recurrent neural networks (RNNs), gated recurrent units (GRUs), long short-term memory networks (LSTMs), graph neural networks (GNN), and others [19]. LSTM models improve RNN in order to capture long-term dependencies while preserving the previous network states. Therefore, LSTM is also widely used for time series forecasting [7] [20] [21] [22] [23] [24] [25] [26], [27]. These works use similar unidirectional LSTM models for prediction.

The research community has shown a great deal of interest in machine learning applications to Covid-19. There are numerous research publications that attempt to employ machine learning to prevent the pandemic [2] [28] [29]. The COVID-19 and Spain Market Index (IBEX 35) short-term confirmed cases were projected using the SutteARIMA approach in [30]. Based on the mean absolute percentage error (MAPE) values, the Sut- teARIMA approach was found to be superior to the AutoRegressive Integrated Moving Average (ARIMA) for predicting daily confirmed cases in Spain. In order to predict the number of confirmed COVID-19 cases in China, the study in [31] proposed an enhanced version of the Adaptive Neuro-Fuzzy Inference System (ANFIS) based on the Flower Pollination Algorithm (FPA). The goal is to use a hybrid of the Flower Pollination and Swarm Swarm Algorithms to find the parameters of the Adaptive Neuro-Fuzzy Inference System. The susceptible, infected, recovered, and deceased (SIRD) model and other crucial variables were used by the authors in [4] to forecast the COVID-19 epidemic's growth in China. They suggested a method for forecasting the reproduction number (R0) from these variables. For India, the authors developed two genetic programming-based COVID-19 prediction models [32]. Their findings show that genetic evolutionary programming models are extremely reliable for COVID-19 situations in India. The SIR model was employed by the authors in [33] to predict confirmed COVID-19 cases in the Eastern Mediterranean region, specifically in Iran, Iraq, Saudi Arabia, the United Arab Emirates, Lebanon, Egypt, and Pakistan, with an emphasis on Pakistan.

In this paper, four different models, ARIMA, Prophet, NARNN, and LSTM, are used to forecast the rate of COVID-19 spread from the beginning of the spread in Turkey by using whole data (as a long-term data set). Moreover, we employ two different LSTM models: Stacked Unidirectional LSTM (SLSTM) and Bidirectional LSTM (BDLSTM). We selected these models because LSTM and its variations are the most widely used models and also have the best performance values [29]; however, the performance of these models depends on the data (country and time frame). Convolutional LSTM and bidirectional LSTM have demonstrated the highest accuracy among LSTM extensions, according to these studies [2] [29]. The performance of recurrent neural networks is based on the data set, and in the literature, most studies on COVID-19 predictions focus on complete data, starting with the first case. Employing the whole data set for prediction can result in erroneous results because governments have experimented with various control mechanisms throughout various periods. Therefore, the proposed models are also applied to four different short-term data sets, which contain daily cases and mortality rates of periods in which the government applies different precaution strategies. The results are compared with each other in order to find the best forecasting model for each precaution strategy.

Table 1. *Overview of COVID-19 Data sets.*

No	Date	Total Case	Total Death	Daily Case	Daily Death
1	18/05/21	5139485	45186	11937	231
2	19/05/21	5151038	45419	11553	233
3	20/05/21	5160423	45626	9385	207
4	21/05/21	5169951	45840	9528	214
5	22/05/21	5178648	46071	8697	231

Table 2. *Data set properties.*

Data Set	Start date (2021)	End date (2021)	Days	Period
DS_1	13/01	03/03	50	Before normalization
DS_2	04/03	19/04	47	Normalization
DS_3	20/04	05/05	16	Partial shutdown
DS_4	06/05	22/05	17	Full shutdown
DS_5	13/01	22/05	130	Whole period

II. METHODOLOGY

A. DATA PREPARATION

The daily prevalence data of COVID-19 has been taken from The Ministry of Health of Turkey [34]. Long-term data consists of daily results for total confirmed cases, daily confirmed cases, total deaths and daily deaths starting from January 13, 2021 (from first vaccination) to May 22, 2021 and sample data is shown in Table 1.

In order to show the effect of precaution strategies, this long term data is split into different data sets. Each data set consists of the COVID-19 results starting from the one precaution period to the next one. Since average incubation period of corona virus is 5.1 days [35], the start and end of each data set is shifted by using this value. After splitting whole data set, there are five different data sets shown in Table 2.

The data set for starting from first vaccination date of Turkey to start of the gradual lifting of COVID-19 restrictions and called DS_1. The second data set contains data starting from the gradual lifting of COVID-19 restrictions to start of next restrictions period and called DS_2. The third data set contains data for restriction period and called DS_3. The fourth data set is for total lockdown period and called DS_4. And finally, whole data set is used in order to compare the forecasting models and this data set is called DS_5.

B. FORECASTING METHODS

B. 1. ARIMA

Auto Regressive Integrated Moving Average (ARIMA) is a statistical analysis technique and uses time series data to explain or predict this data based on its own past values. A non-seasonal ARIMA model is represented by using three parameters (p, d, q), where p is the number of autoregressive terms, d is the number of differencing degree, and q is the number of lagged forecast errors in the prediction equation.

ARIMA consists of three different components; AR, I, and MA models. Auto regression (AR) part of ARIMA refers to a model that shows uses the dependent relationship between current data and its past values. AR(p) means p lagged error terms are going to be used in the ARIMA model and the general formula for AR(p) can be expressed in Eq.1.

$$Y_t = \delta + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{p-1} + \varepsilon_t \qquad (1)$$

where δ is constant value, p is past value, φ is auto regression value, t is time, Y_t is observed value at a time t, ε_t is error term.

MA component stands for moving average and shows dependency between outcome of the model and previous observations. *qth* degree of moving average process MA(q) can be found as:

$$Y_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{q-1} + \varepsilon_t \qquad (2)$$

where, μ is constant value, q is past value, θ indicates moving average value.
Without I component, we can define ARMA (p, q) by combining these two equations:

$$Y_t = \delta + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \cdots + \varphi_p Y_{p-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{q-1} \qquad (3)$$

The I component stands for integrated and represents the differencing of raw observations to allow for the time series to become stationary. If the processed time series is not stationary, it can be made stationary by taking the difference process *d* times. First order differencing (d=1) is represented as:

$$Y'_t = Y_t - Y_{t-1} \qquad (4)$$

Similarly, second order differencing (d=2) is represented as:

$$Y''_t = Y'_t - Y'_{t-1} \qquad (5)$$

Here, Y'_t and Y''_t are first and second order differences for Y_t. Thus, ARIMA (p, d, q) can be calculated by differencing on the time series as per d. Hence, non-stationary feature is removed by this differencing process. After that process, our model can fit the generated time series with the equation that combines Eq.1 and Eq.2:

$$Y_t = \delta + \varepsilon_t + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j Y_{t-j} \tag{6}$$

B. 2. Prophet

Facebook created the open-source time-series forecasting library known as Prophet. For time series forecasting, it employs a variety of unique techniques. Seasonality is also supported by Prophet. It consists of three essential parts; trend, seasonality and holidays. The time series data's trend is described by the first component, which is referred to as trend in Prophet also. Seasonality and holidays make up the second and third components, respectively. The following equation can be used to describe these tree components;

$$Y_t = g_t + s_t + h_t + \varepsilon_t \tag{7}$$

where, g_t is trend, s_t is seasonality and h_t is holidays. Once more, ε_t is an error term that accounts for any irregular changes that the model might not be able to tolerate.

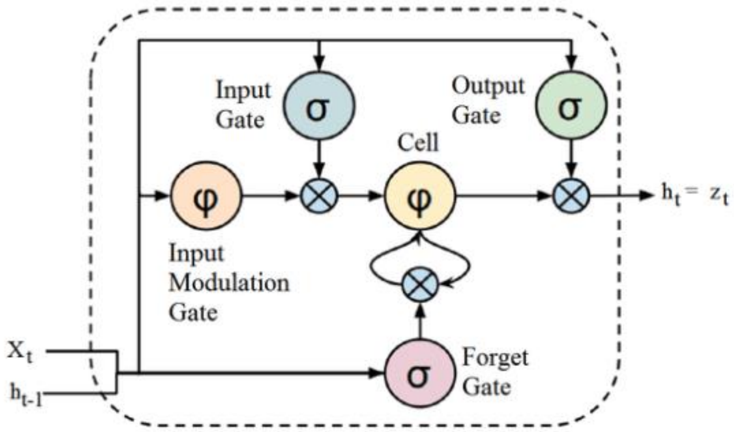

***Figure 1**. LSTM cell structure*[36].

B. 3. LSTM

Because they have their own memory structure, LSTMs, a particular type of recurrent neural network (RNN), are frequently employed in time series forecasting, emotional analysis, text analysis, and speech recognition. A gating mechanism is used in the LSTM model to remember or store lengthy data sequences. This gating mechanism makes use of data from earlier steps to evaluate a function and generate an output. The current LSTM cell state is changed via this output. Input gates, output gates, and forget gates are the three gate configurations that make up an LSTM cell. Figure 1 depicts the structure of an LSTM cell.

The following formula is used to use the forget gate to decide which data will be retained or not;

$$f_t = \sigma\,(W_f[\,h_{t-1}, x_t] + b_f) \tag{8}$$

In this equation, x_t represents input at time t, h_{t-1} represents previous cell output, and σ represents sigmoid function. Information is preserved in the cell state if the forget gate's output is 1 (one). Following this, the sigmoid function generates a vector of potential new values. Which values will be updated is decided by input gates and the vector $C^{\sim}{}_t$ is computed by following equation;

$$i_t = \sigma(W_i[\, h_{t-1}, x_t] + b_i) \tag{9}$$

$$C^{\sim}{}_t = \tanh(\, W_c[\, h_{t-1}, x_t] + b_c \tag{10}$$

Now cell's old state C_{t-1} is updated to new cell state C_t.

$$C_t = f_t * C_{t-1} + i_t * C^{\sim}{}_t \tag{11}$$

At the end, we decide the output of the network, and this output depends on the integrity of our cells. The *tanh* function is used for cell state and multiplied by this *sigmoid* layer after the *sigmoid* layer determines which components of the cell state will be utilized;

$$o_t = \sigma(W_o[\, h_{t-1}, x_t] + b_o) \tag{12}$$

$$h_t = o_t * \tanh(C_t) \tag{13}$$

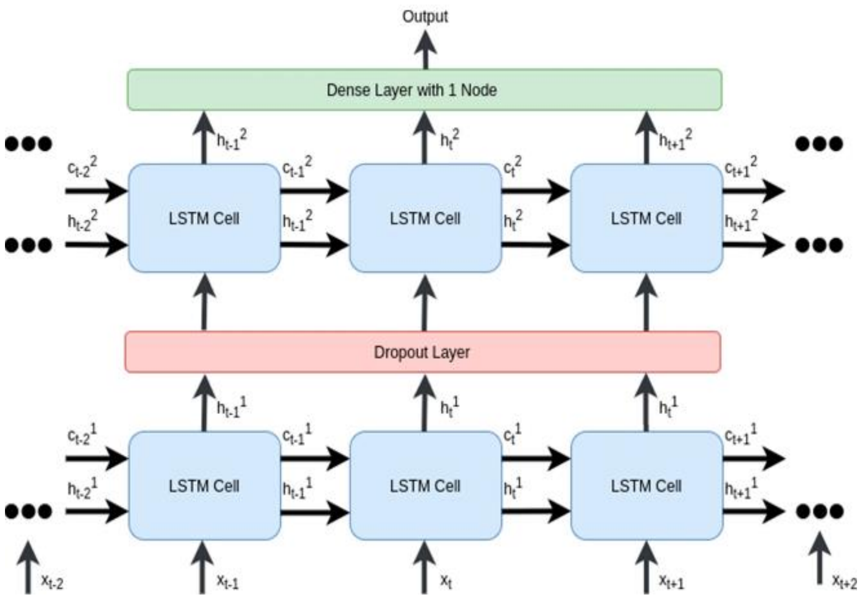

Figure 2. *Stacked LSTM network.*[36]

Multiple LSTM layers are utilized in the stacked LSTMs (SLSTMs) that are presented in [37]. Time series information is fed into the first LSTM layer, which then generates the output. The following LSTM layer is fed using this output. The internal architecture of every LSTM layer is the same, but the number of units differs. Figure 2 displays a Stacked LSTM structure.

Conventional LSTMs utilize only prior information in order to resolve the following states. the In order to handle input in both directions, bidirectional LSTMs (BDLSTMs) are developed [38]. Two distinct hidden layers are combined to create BDLSTMs, which enable bidirectional information transmission at every time step by combining two independent LSTMs. The BDLSTM cell has two different inputs; one from prior steps and another one from the following step. The BDLSTM network may store information from the past and the future by combining the inputs and outputs of two independent BDLSTM cells. The general architecture of BDLSTM is depicted in Figure 3 [39].

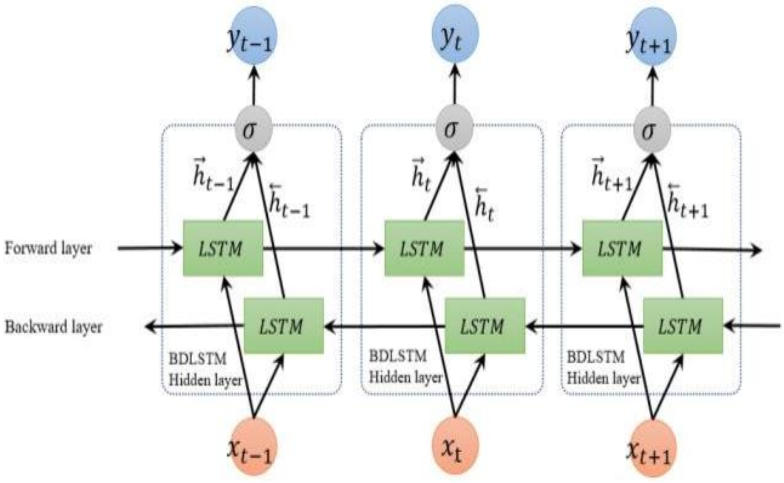

Figure 3. *Bidirectional LSTM network* [36].

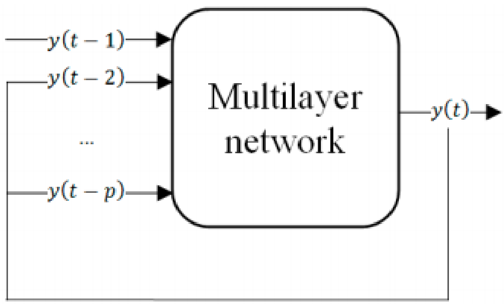

Figure 4. *Nonlinear Autoregression Neural Network (NARNN)* [40].

B. 4. NARNN

A popular method, particularly for time series predictions, is the nonlinear autoregression neural network (NARNN). Given previous values for the same time series, the NARNN model can accurately forecast a simple time series as described by the following equation;

$$Y_t = f(Y_{t-1}, \dots, Y_{t-p}) + \varepsilon_t \tag{14}$$

This equation indicates that the value of Y in time t, Y_t is a function of the past p number. The topology of a NARNN is shown in Figure 4.

C. MODEL SELECTION

In order to evaluate the performance of the proposed models, the actual values and the predicted values are compared. Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) are employed to analyze each model's performance. The following equations are used to compute each metric;

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y'_i - Y_i)^2 \tag{15}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (Y'_i - Y_i)^2}{n}} \tag{16}$$

$$MAPE = \frac{100}{n}\ x\ \sum_{i=1}^{n} \left| \frac{(Y_i - Y'_i)}{Y_i} \right| \quad (17)$$

III. RESULTS and DISCUSSION

In this study, unlike other studies on forecasting COVID-19 cases, the data set is split into four different categories according to precaution strategies.

Since time series forecasting frequently uses ARIMA, we began with this approach. Checking whether or not the time series is stationary is the first step in implementing the ARIMA model. Because the variance and mean of the data must remain constant throughout time, ARIMA performs best when our data exhibits a stable or consistent pattern. In other words, the data is not stationary when there is an upward or downward trend and a certain pattern (seasonality).

Table 3. *ADF Tests of COVID-19 Datasets.*

	DS_1	DS_2	DS_3	DS_4	DS_5
ADF	-2.038913096	-3.772966554	-77.26962262	-8.958119357	-3.569350232
p-value	0.99872143	0.003196334	0	0.008366016	0.98683529
lags used	8	7	6	6	8
Critical values (%1)	-3.600983367	-3.610399601	-4.473135048	-3.924019385	-3.485585146
Critical values (%5)	-2.935134816	-2.939108946	-3.289880604	-3.068498203	-2.885738566
Critical values (%10)	-2.60596298	-2.608062965	-2.772382346	-2.673892656	-2.579675908

While a statistical test like the Augmented Dickey Fuller (ADF) Test makes significant assumptions about our data, we employed it to verify the stationarity of the data. ADF reveals the degree to which a null hypothesis can be rejected or not rejected in order to evaluate if our data is stationary or not. This is expressed using a cutoff (0.05) indicating whether we accept or reject the null hypothesis. Table 3 displays the data's ADF results. ADF results show that, for DS_1 and DS_5, our data is not stationary while for DS_2, DS_3 and DS_4, our data is stationary.

Table 4. *ARIMA parameters.*

	p	d	q
DS_1	10	2	1
DS_2	10	2	0
DS_3	8	0	0
DS_4	9	0	0
DS_5	10	2	0

Hence, we have to transform DS_1 and DS_5 into stationary data. There are many methods for this purpose and we have used log-scale transformation. After converting our data into stationary data, we have to find best p, d and q parameters of ARIMA model. The Akaike Information Criteria (AIC) expression is typically used to measure model performance in order to determine the most acceptable parameter in the ARIMA technique. It is computed using the following equation:

$$AIC = -2\log(L) + 2(p + q + k) \quad (18)$$

Thus, L stands for the probability of the data, p for the autoregressive part's order, q for the moving average part's order, and k for the ARIMA model's intercept. The model with the lowest AIC criterion is distinguished from the rest by using this parameter to determine its success. In this work, for each dataset, the indicators with lowest AIC value are chosen by evaluating performance test and these ARIMA parameters for each data set are shown in Table 4.

Table 5. *Respective LSTM Model topologies for each dataset.*

Dataset	Method	Input	Hidden layer 1	Hidden layer 2	Hidden layer 3	Output	Sample Length
DS_1	SLSTM	(50,1)	32	32	32	(1)	20
	BDLSTM	(50,1)	32	16	-	(1)	20
DS_2	SLSTM	(47,1)	32	32	32	(1)	14
	BDLSTM	(47,1)	32	16	-	(1)	20
DS_3	SLSTM	(16,1)	8	8	8	(1)	7
	BDLSTM	(16,1)	8	8	-	(1)	7
DS_4	SLSTM	(17,1)	8	8	8	(1)	7
	BDLSTM	(17,1)	8	8	-	(1)	7
DS_5	SLSTM	(130,1)	64	64	64	(1)	14
	BDLSTM	(130,1)	64	32	-	(1)	14

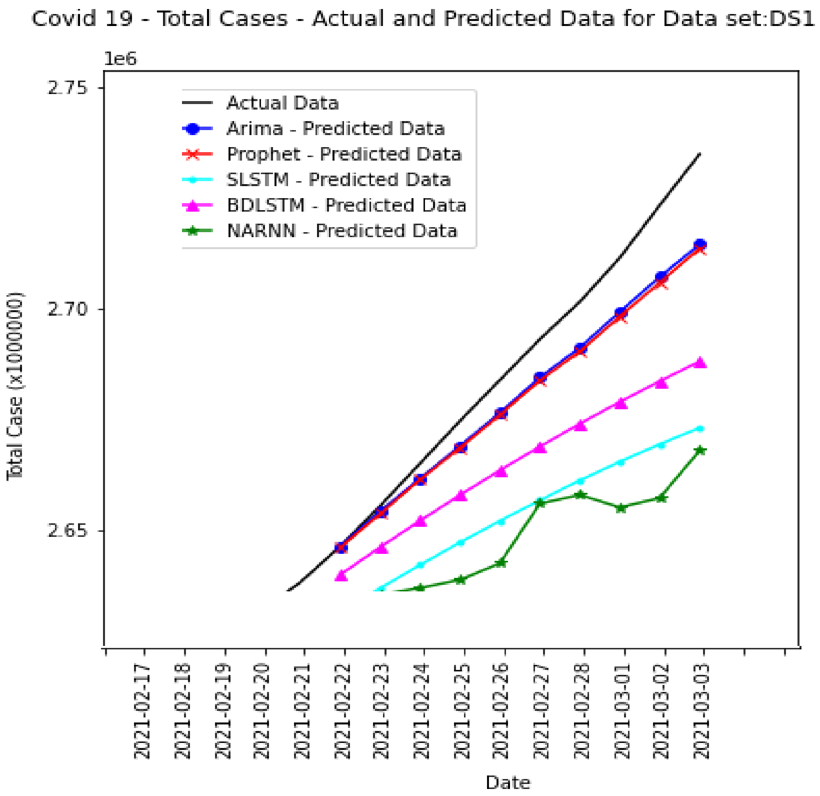

Figure 5*. Actual and predicted cumulative confirmed cases for DS_1.*

For each dataset, one SLSTM and one BDLSTM is developed and trained. In the training phase of both LSTM models, *adam optimizer* is utilized and as the loss function, Mean squared error (MSE) is computed. Table **5** shows the topology for both SLSTM and BDLSTM models.

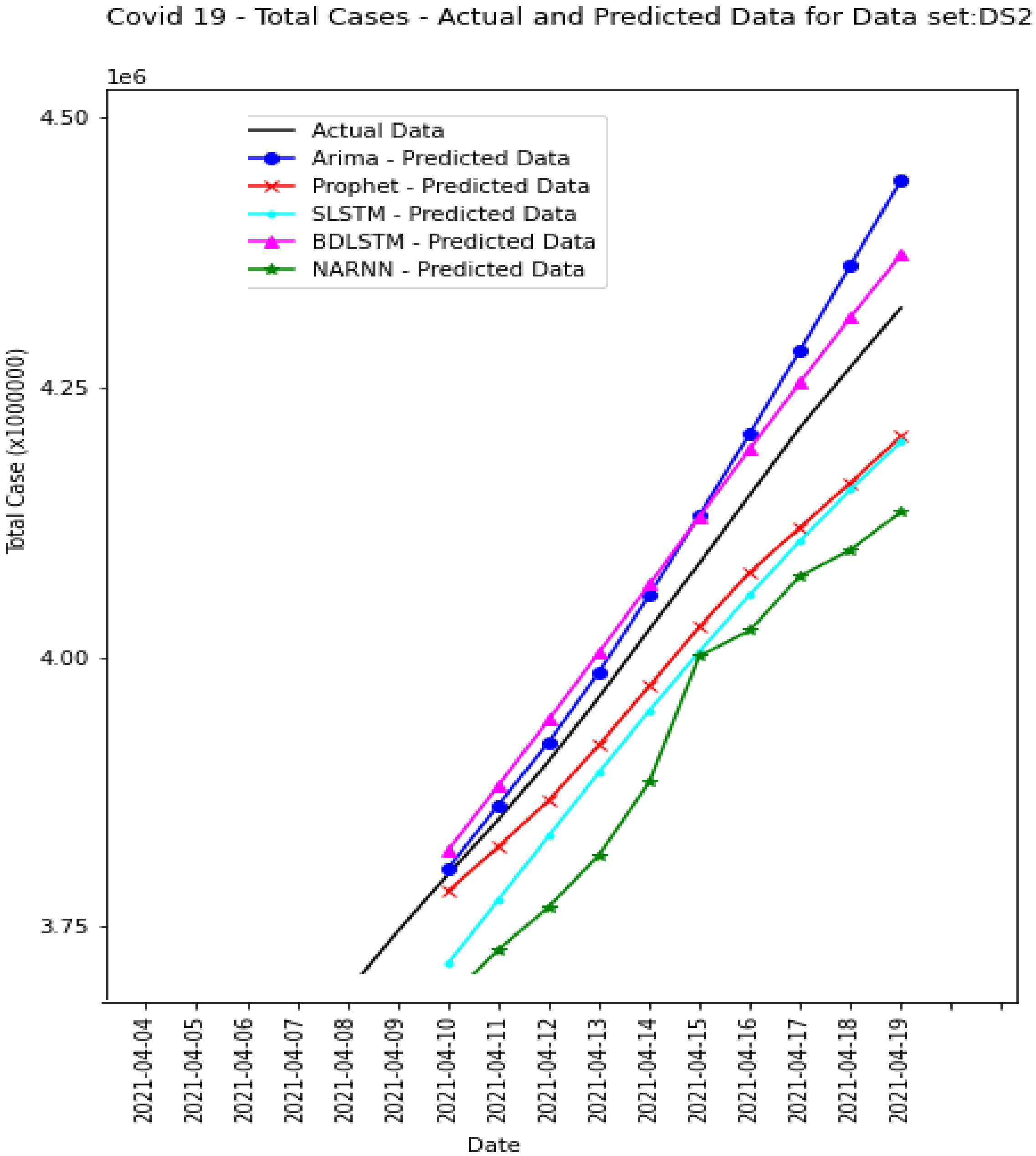

Figure 6*- Actual and predicted cumulative confirmed cases for DS_2.*

For each dataset, SLSTMs contain three hidden layers and one input layer. Between these layers, a model with a 0.2 dropout rate creates the dropout layers. Dropout is utilized to reduce overfitting and enhance the model's performance [41]. To generate a one-dimensional output, one dense layer is placed as the model's last layer. Similarly, BDLSTMs are constructed using two hidden layers with same dropout factor and one dense layer. Thus, number of hidden layers of SLSTM and BDLSTM differ for each data set. To use time series data with LSTM models, they must be transformed into a structure of samples with input and output components.

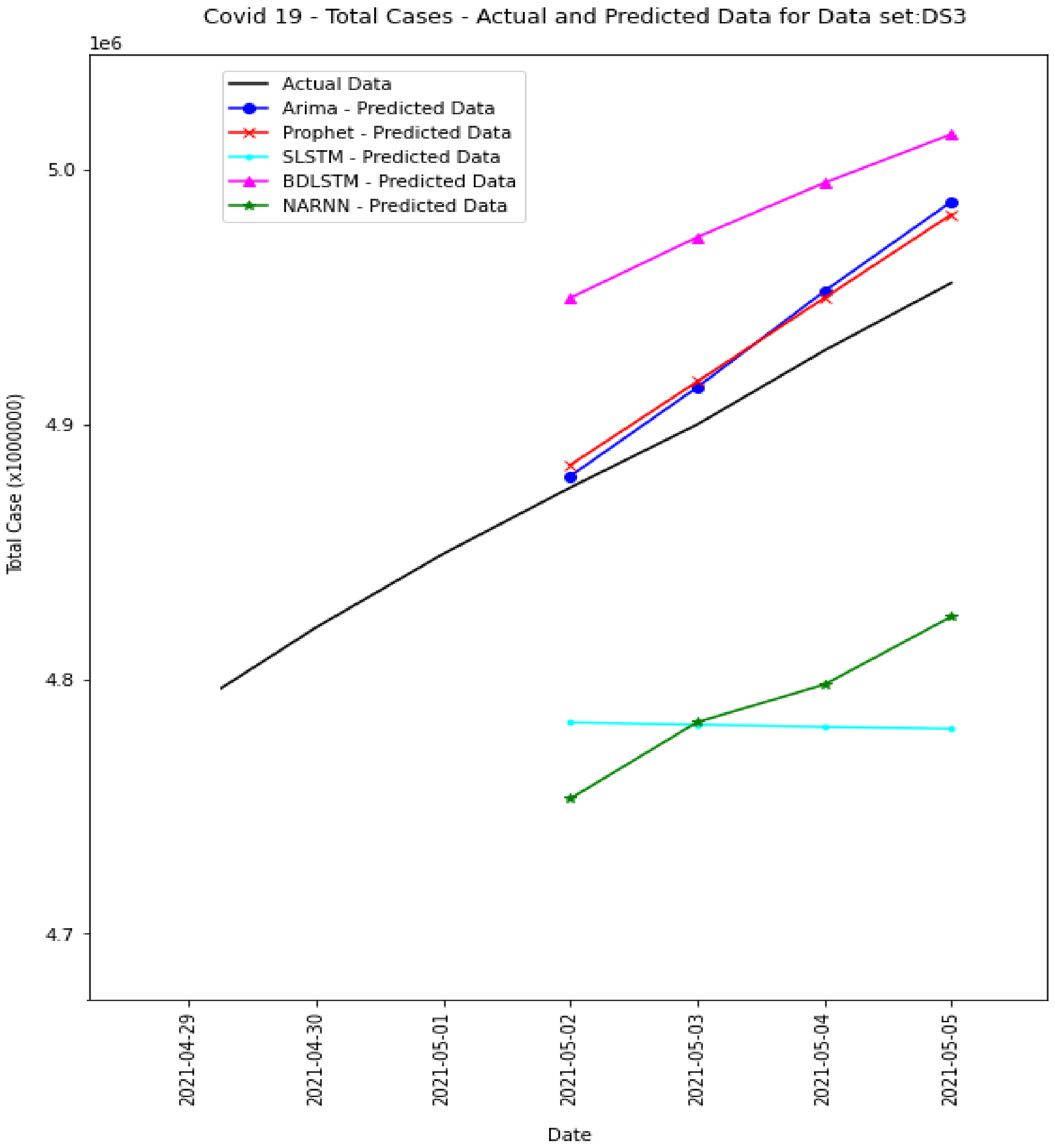

Figure 7- *Actual and predicted cumulative confirmed cases for* DS_3.

Python Keras library is used to convert time series data into samples by using *TimeseriesGenerator*. TimeseriesGenerator requires the model's training data sample size as length paramter. The purpose of this parameter is to forecast the value that will follow next to the sample input. This sample input should have a length of the number of elements. Therefore, a various number of sample lengths are utilized for both models to achieve the highest levels of accuracy, and these lengths are displayed in Table 5's "Sample Length" column.

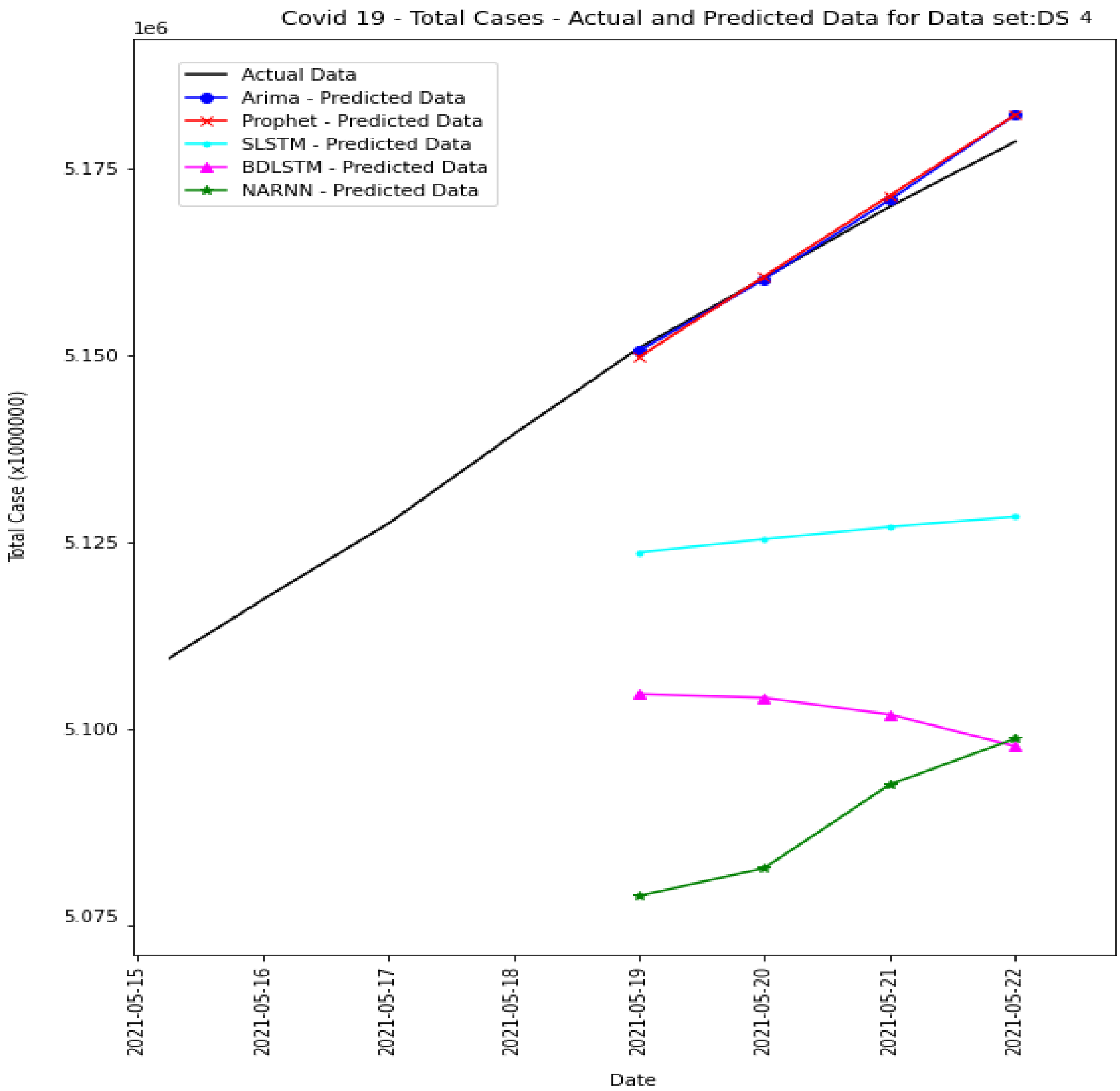

Figure 8. *Actual and predicted cumulative confirmed cases for DS_4.*

For NARNN, Matlab's *narnet* model is used. In order to create the NARNN, the delay of the feedback is set to 1:2; with 10 hidden Layer size. For the training algorithm, the default training algorithm (*trainlm*) is also used for each dataset. Figure 5, Figure 6, Figure 7, Figure 8, and Figure 9 show the prediction results for each data set, including overall data set, for cumulative confirmed cases. Table 6 shows the performance results of models for each data set.

The results given here demonstrate significant variation in accuracy between and among standalone models. From the figures, we can conclude that using Prophet or ARIMA as forecasting model lead to better results when we have small portion of data. Stacked LSTM outperforms all other models for the entire data set, since the model uses much more input sequences than the other data sets. Moreover, the test results show that if we split our data according to precaution strategy and use enough input sequences, all models have their best accuracy results. Since DS_3 and DS_4 contain very small data, machine learning methods suffer to forecast meaningful values.

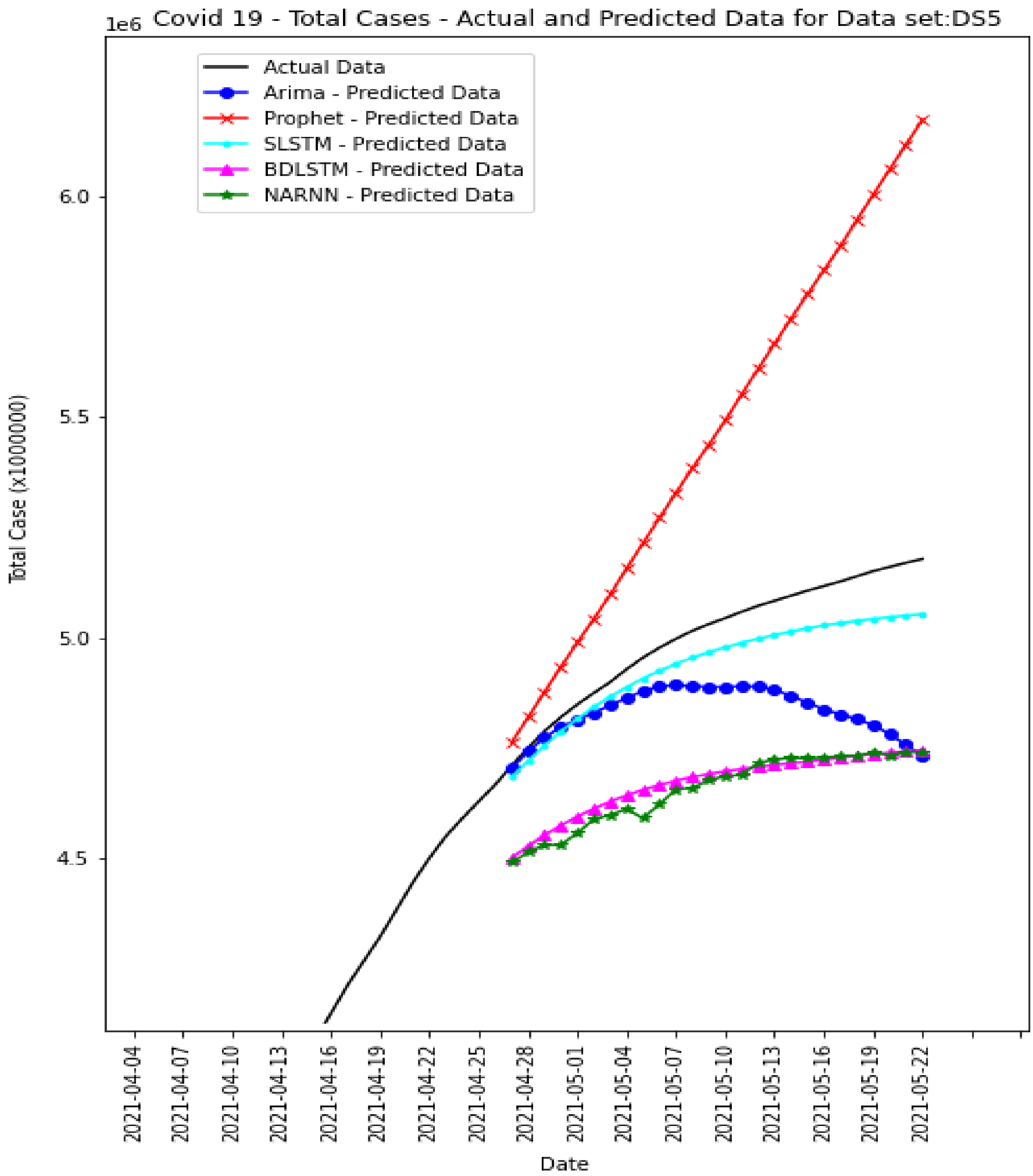

Figure 9. *Actual and predicted cumulative confirmed cases for DS.*

On the other hand, when all data is included in our input space, models cumulative performance decreases because of government pandemic strategies. Thus, for meaningful and accurate prediction, the data set should include much number of data with appropriate splitting method. Moreover, in order to have better forecasting and to reduce uncertainty, more features such as age, seasonality etc. can be used as input incorporating with split strategy.

Table 6. *Performance results of models for each data set.*

Data Set	Method	MAPE	MSE	RMSE
DS_1	ARIMA	852335	110240920	10499.57
	Prophet	920115	125570644	11205.83
	NARNN	3285475	979201448	31292.19
	SLSTM	2670070	834524179	28888.13
	BDLSTM	1215421	241116240	15527.92
DS_2	ARIMA	4775929	3524265883	59365.53
	Prophet	7247629	7025117047	83815.97
	NARNN	12787839	27737527283	166545.87
	SLSTM	9714512	15784545130	125636.56
	BDLSTM	6225460	4968966033	70490.896
DS_3	ARIMA	1852802	446212980	21123.756
	Prophet	1826471	375859006	19387.08
	NARNN	3745542	2001322581	44736.14
	SLSTM	4596954	2404672686	49037.46
	BDLSTM	1347133	1296391385	36005.43
DS_4	ARIMA	126840	3492913	1868.93
	Prophet	162587	4162234	2040.15
	NARNN	6085991	3854004009	62080.62
	SLSTM	3787150	2781408706	52739.06
	BDLSTM	5802520	3465741641	58870.55
DS_5	ARIMA	10633720	14524776223	120518.78
	Prophet	46301198	299748127805	547492.13
	NARNN	19058715	58901238467	242695.77
	SLSTM	6713450	4585028698	67712.84
	BDLSTM	17148845	46870775713	216496.59

IV. CONCLUSION

We have presented the use of statistical methods, machine learning methods, and the use of two different LSTM models for COVID-19 forecasting for Turkey. Moreover, in order to show the effect of the government's precaution strategies, we have split the whole data set according to these strategies and compared the performance of these models over these split data.

Our findings highlight the difficulties of forecasting given sparse, highly biased data because there are fluctuations when considering the whole data set. Therefore, we split the entire data set into different sets by considering the precautionary strategies. We experimented with different ways of creating training and test data and models, which all showed strengths and limitations that made it difficult to choose a single model. The test results show that for small-sized periods, the statistical approaches outperform deep learning models in terms of accuracy. On the other hand, when input size increases, the performance of LSTM models and NARNN also increases. LSTM model performance is directly dependent on data. Its performance increases when the data with a long-term relationship is used because it enables the learning of even more parameters, and the LSTM cell increases long-term memory even more efficiently. Moreover, LSTM models can extract features that other traditional approaches cannot process.

The proposed model has several shortcomings. Despite this performance gain, the respective LSTM models can be improved using different parameters and architectures. Moreover, it is essential to use more features, such as seasonality, age, etc., in the models to have better performance results.

V. REFERENCES

[1] A. Rismanbaf, "Potential Treatments for COVID-19; a Narrative Literature Review," *Archives of academic emergency medicine*, vol.8(1), 2020.

[2] I. Rahimi, F. Chen, and A. H. Gandomi, "A review on COVID-19 forecasting models," *Neural Comput. Appl.*, vol. 35, pp.23671–23681, 2023.

[3] I. Nesteruk, "Statistics based predictions of coronavirus 2019-nCoV spreading in mainland China," *medRxiv*, 2020.

[4] C. Anastassopoulou, L. Russo, A. Tsakris, and C. Siettos, "Data-based analysis, modelling and forecasting of the COVID-19 outbreak," *PLoS One*, vol. 15, no. 3, pp. 1–21, 2020.

[5] G. Giordano *et al.*, "Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy," *Nat. Med.*, vol. 26, no. 6, pp. 855–860, 2020.

[6] S. Moein *et al.*, "Inefficiency of SIR models in forecasting COVID-19 epidemic: a case study of Isfahan," *Sci. Rep.*, vol. 11, no. 1, p. 4725, 2021.

[7] İ. Kırbaş, A. Sözen, A. D. Tuncer, and F. Ş. Kazancıoğlu, "Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches," *Chaos, Solitons and Fractals*, vol. 138, Sep. 2020.

[8] T. Dehesh, H. A. Mardani-Fard, and P. Dehesh, "Forecasting of COVID-19 Confirmed Cases in Different Countries with ARIMA Models," *medRxiv* ,2020.

[9] D. Benvenuto, M. Giovanetti, L. Vassallo, S. Angeletti, and M. Ciccozzi, "Application of the ARIMA model on the COVID-2019 epidemic dataset," *Data Br.*, vol. 29, p. 105340, 2020.

[10] M. Yousaf, S. Zahir, M. Riaz, S. M. Hussain, and K. Shah, "Statistical analysis of forecasting COVID-19 for upcoming month in Pakistan," *Chaos, Solitons & Fractals*, vol. 138, p. 109926, 2020.

[11] O. D. Ilie, R. O. Cojocariu, A. Ciobica, S. I. Timofte, I. Mavroudis, and B. Doroftei, "Forecasting the spreading of COVID-19 across nine countries from Europe, Asia, and the American continents using the arima models," *Microorganisms*, vol. 8, no. 8, pp. 1–19, 2020.

[12] Z. Ceylan, "Estimation of COVID-19 prevalence in Italy, Spain, and France," *Sci. Total Environ.*, vol. 729, Aug. 2020.

[13] Facebook, "Prophet Web Page." https://facebook.github.io/prophet/ (accessed Feb. 15, 2023).

[14] B. M. Ndiaye, L. Tendeng, and D. Seck, "Analysis of the COVID-19 pandemic by SIR model and machine learning technics for forecasting,", *arXiv preprint arXiv:2004.01574*, 2020.

[15] S. S. Helli, Ç. Demirci, O. Çoban, and A. Hamamci, “Short-Term Forecasting COVID-19 Cases in Turkey Using Long Short-Term Memory Network,” *TIPTEKNO 2020 - Tip Teknol. Kongresi - 2020 Med. Technol. Congr. TIPTEKNO 2020*, pp. 2–5, 2020.

[16] İ. Kırbaş, A. Sözen, A. D. Tuncer, and F. Ş. Kazancıoğlu, “Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches,” *Chaos, Solitons and Fractals*, vol. 138, 2020.

[17] S. Namasudra, S. Dhamodharavadhani, and R. Rathipriya, “Nonlinear Neural Network Based Forecasting Model for Predicting COVID-19 Cases,” *Neural Process. Lett.*, vol.55, pp-171-191, 2023.

[18] A. I. Saba and A. H. Elsheikh, “Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks,” *Process Saf. Environ. Prot.*, vol. 141, pp. 1–8, 2020.

[19] W. Jiang and H. D. Schotten, “Deep Learning for Fading Channel Prediction,” *IEEE Open J. Commun. Soc.*, vol. 1, pp. 320–332, 2020.

[20] A. Tomar and N. Gupta, “Prediction for the spread of COVID-19 in India and effectiveness of preventive measures,” *Sci. Total Environ.*, vol. 728, p. 138762, 2020.

[21] V. K. R. Chimmula and L. Zhang, “Time series forecasting of COVID-19 transmission in Canada using LSTM networks,” *Chaos, Solitons and Fractals*, vol. 135, Jun. 2020.

[22] R. Pal, A. A. Sekh, S. Kar, and D. K. Prasad, “Neural Network Based Country Wise Risk Prediction of COVID-19", *Applied Sciences*, vol.10(18), 2020.

[23] A. J. Aljaaf, T. M. Mohsin, D. Al-Jumeily, and M. Alloghani, “A fusion of data science and feed-forward neural network-based modelling of COVID-19 outbreak forecasting in IRAQ,” *J. Biomed. Inform.*, vol. 118, no.04, p. 103766, 2021.

[24] A. Dairi, F. Harrou, A. Zeroual, M. M. Hittawe, and Y. Sun, “Comparative study of machine learning methods for COVID-19 transmission forecasting,” *J. Biomed. Inform.*, vol. 118, no. 04, p. 103791, 2021.

[25] S. Dutta and S. Kumar Bandyopadhyay, “Machine learning approach for confirmation of COVID-19 cases: positive, negative, death and release,” *Iberoam. J. Med.*, vol. 03, pp. 172–177, 2020.

[26] S. Arslan, “A hybrid forecasting model using LSTM and Prophet for energy consumption with decomposition of time series data,”, *PeerJ Computer Science*, vol.8, 2022.

[27] S. Arslan, “Gated recurrent unit network-based fuzzy time series forecasting model,” *Afyon Kocatepe Univ. J. Sci. Eng.*, vol. 23, no. 3, pp. 677–692, 2023.

[28] S. Tiwari, P. Chanak, and S. K. Singh, “A Review of the Machine Learning Algorithms for Covid-19 Case Analysis,” *IEEE Trans. Artif. Intell.*, vol. 4, no. 1, pp. 44–59, 2023.

[29] F. Kamalov, K. Rajab, A. K. Cherukuri, A. Elnagar, and M. Safaraliev, “Deep learning for Covid-19 forecasting: State-of-the-art review.,” *Neurocomputing*, vol. 511, pp. 142–154, 2022.

[30] A. S. Ahmar and E. B. del Val, “SutteARIMA: Short-term forecasting method, a case: Covid-19 and stock market in Spain,” *Sci. Total Environ.*, vol. 729, p. 138883, 2020.

[31] M. A. A. Al-qaness, A. A. Ewees, H. Fan, and M. Abd El Aziz, “Optimization Method for Forecasting Confirmed Cases of COVID-19 in China,” *J. Clin. Med.*, vol. 9, no. 3, 2020.

[32] R. Salgotra, M. Gandomi, and A. H. Gandomi, “Time Series Analysis and Forecast of the COVID-19 Pandemic in India using Genetic Programming,” *Chaos, Solitons & Fractals*, vol. 138, p. 109945, 2020.

[33] S. Dil, N. Dil, and Z. H. Maken, “COVID-19 Trends and Forecast in the Eastern Mediterranean Region With a Particular Focus on Pakistan.,” *Cureus*, vol. 12, no. 6, p. e8582, Jun. 2020.

[34] Ministry of Health Web Page, https://covid19.saglik.gov.tr/ (accessed May 23, 2021).

[35] W. H. Organization, https://www.who.int/emergencies/diseases/novel-coronavirus-2019 (accessed Feb. 15, 2021).

[36] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[37] A. Graves, A. Mohamed, and G. E. Hinton, “Speech Recognition with Deep Recurrent Neural Networks,” *2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649)*, 2013.

[38] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[39] Z. Cui and Y. Wang, “Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction,” *arXiv preprint arXiv:1801.02143*, 2018.

[40] J. G. De Gooijer and R. J. Hyndman, “25 years of time series forecasting,” *Int. J. Forecast.*, vol. 22, no. 3, pp. 443–473, 2006.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.