A Deep Learning BiLSTM Encoding-Decoding Model for COVID-19 Pandemic Spread Forecasting

Ahmed I. Shahin * and Sultan Almotairi *

Abstract: The COVID-19 pandemic has widely spread with an increasing infection rate through more than 200 countries. The governments of the world need to record the confirmed infectious, recovered, and death cases for the present state and predict the cases. In favor of future case prediction, governments can impose opening and closing procedures to save human lives by slowing down the pandemic progression spread. There are several forecasting models for pandemic time series based on statistical processing and machine learning algorithms. Deep learning has been proven as an excellent tool for time series forecasting problems. This paper proposes a deep learning time-series prediction model to forecast the confirmed, recovered, and death cases. Our proposed network is based on an encoding–decoding deep learning network. Moreover, we optimize the selection of our proposed network hyper-parameters. Our proposed forecasting model was applied in Saudi Arabia. Then, we applied the proposed model to other countries. Our study covers two categories of countries that have witnessed different spread waves this year. During our experiments, we compared our proposed model and the other time-series forecasting models, which totaled fifteen prediction models: three statistical models, three deep learning models, seven machine learning models, and one prophet model. Our proposed forecasting model accuracy was assessed using several statistical evaluation criteria. It achieved the lowest error values and achieved the highest R-squared value of 0.99. Our proposed model may help policymakers to improve the pandemic spread control, and our method can be generalized for other time series forecasting tasks.

Keywords: machine learning; deep learning; COVID-19 spread; comparison methods; ARIMA; prophet model; LSTM; BiLSTM; Encoder–Decoder

1. Introduction

During the writing of this paper, an infectious disease from a new generation of Coronavirus (Coronavirus disease 2019, COVID-19) appeared in several countries. Transportation restrictions also underwent changes due to the COVID-19 pandemic spread.

As a precautionary procedure, several countries have stopped international air flights more than once. The first version of COVID-19 appeared in Wuhan city in China at the end of December 2019. COVID-19 was announced by the World Health Organization (WHO) to be a global pandemic on 11 March 2020 [1]. COVID-19 exponentially spread over all the world and highly affected the healthcare systems in several countries [2]. The total number of positive confirmed cases reached about 80 million people with death cases of about 1.7 million people [3]. This high death rate has led researchers and scientists in different fields to look for ways to address the challenges of this virus and work on overcoming the epidemic.

Artificial intelligence (AI) tool plays an important role to help the human survival during the COVID-19 pandemic [4,5]. Many researchers have employed AI tool in different applications [6–12].

In [13], an early detection and diagnosis system for COVID-19 was proposed. The patient provided his early disease symptoms, and the patient was classified as a positive
or negative case. In [14], people contact tracing systems were applied in nine counties
with several tracing technologies, such as Bluetooth, WiFi, and GPS. This tracing aimed to
reduce the pandemic spread.

In [15], several AI algorithms were applied to CT and X-Ray imaging modalities
to classify and detect COVID-19. These algorithms have succeeded in differentiating
between COVID-19, Non-COVID-19, and pneumonia. These algorithms have achieved
high accuracy classification reached to 99 %. In [16], an AI algorithm was introduced to
recognize the potential old drugs with anti-coronavirus activities.

In [17], AI was integrated with surveillance systems that enable Non-mask violation
detection. In [18], an AI algorithm was able to monitor and track social distancing.
The system monitored the minimum safe distancing 2-m, which the governments and
health authorities have recommended in shopping centers, streets, and schools. In [19],
the AI algorithm was introduced to forecast the future spread of COVID-19 in several
countries and globally. Therefore, forecasting pandemic spread represents a real important
policymaker parameter that needs accurate prediction models.

Deep learning networks were introduced by LeCun [20], which represent the new
era of AI algorithms. The traditional AI algorithms are based on pre-processing, feature
extraction and selection, and classification. These sequences consume much time and
lacking to generalization [21]. Deep learning introduced the convolutional neural network
(CNN), which extracts the features without pre-processing, hand-crafted features extraction.
CNN can automatically extract valuable features through consequences convolutional
layers. These layers are defined as separable learnable parameters, which acquire their
parameters during their training process.

The recurrent neural network (RNN) is more efficient with time series analysis prob-
lems. A special type of recurrent neural network (RNN) is long-short term memory (LSTM),
which works in one-directional. Bidirectional long-short term memory (BiLSTM) is pro-
posed to work in both past and future directions. Both LSTM and BiLSTM can extract
temporal features and predict time-series [22]. Gated recurrent unit (GRU) network is also a
special type of deep network with a simpler structure than LSTM, with higher performance
and less training processing time.

Deep learning has been demonstrated in several applications for the COVID-19 pan-
demic, and it achieved better results [23]. There is a real need to employ deep network
capabilities to build a high-performance pandemic spread prediction model. Policymakers
are always in real need of accurately forecasting the pandemic spread. The correct fore-
casting guides the governments to make the correct decisions, such as (1) opening and
closing procedures that stop the pandemic outbreak and save human life, (2) preparing
the healthcare systems for patient flow demand, especially the intensive care units (ICU)
demand and providing the medical needs, (3) applying the correct economic decision for
rationalizing the government spending.

One of the most crucial COVID-19 research dimensions is forecasting the pandemic
spread through confirmed, deaths, and recovered cases. Therefore, the COVID-19 pandemic
spread prediction was investigated based on countries, period, and model. Each country
has its own characteristics, such as population, contact rate, healthcare sector efficiency,
strict opening and closing policies. Therefore, each country prediction spread model can
lead to individual findings. At country level, the previous researches focused on some
countries, such as Spain [24], Italy [24], China [25], India [26], Saudi Arabia [27], Brazil [28],
South Africa [29], and United States of America (USA) [30].

With the start of the COVID-19 spread, scientists attempted to build models to predict
the future scenario of the pandemic to estimate the peak of the confirmed cases and death
cases. Most of these studies gathered the data with starting of February 2020 to the end of
March 2020. Then, the scientists tried to find the answer to the pandemic spread stopping.
Therefore, they included the period starting from early February to the middle of June.

There are five main frameworks for COVID-19 that were utilized to achieve the
COVID-19 spread prediction task. The first framework is based on statistical models,
which are auto-regressive (AR), moving average (MA), and autoregressive integrated moving average (ARIMA). The second framework is based on machine learning models. Some of these models are regression models, such as linear regression, Lasso regression, Elastic Net, Theil-Sen regression, RANSAC regression, Huber regression, support vector machine (SVM), and decision trees. The third framework is based on deep learning models, such as CNN, LSTM, BiLSTM, and GRU. The fourth framework is based on dynamic system models, which are susceptible infectious recovered (SIR), susceptible exposed infectious recovered (SEIR), immunity susceptible exposed infectious recovered (MSEIR). Finally, the fifth framework is based on recently developed prophet models, such as Prophet Model and Google trend prophet model.

The remainder of our paper is organized as follows. Section 2 discusses the literature for the time series prediction task of the COVID-19 pandemic. Section 3 discusses the material utilized in this study with our proposed method. Section 4 presents the experimental section with a detailed discussion. Finally, Section 5 concludes our work and suggests future works.

2. Review of Predictions Models

This section covers most of the time series prediction models utilized to forecast the COVID-19 pandemic spread. First, we cover the most well-known time series forecasting algorithms based on statistical, machine learning, and deep learning approaches. Second, we cover the pandemic spread in different countries during a different times. Finally, we cover the encoder-decoder deep learning models that have been applied for several time-series forecasting tasks.

Several studies in the literature covered different countries and presented models that can predict the COVID-19 pandemic spread. In [31], the authors proposed a comparative study to predict the confirmed positive cases in several countries. The study included the period from 22 January 2020 till 17 June 2020. They have compared between the CNN LSTM, stacked LSTM, and BiLSTM. They assessed their results based on several metrics. However, they have applied their model to forecast cases for 16 days. Their deep learning models hyper-parameters have not been tuned, and they did not compare their results to other time series forecasting statistical models.

In [24], the authors utilized the ARIMA model to predict the confirmed positive cases in the highest infected countries in Europe, which are Italy, Spain, and France. The study included the period from February 21, 2020 to April 15, 2020. Their study achieved RMSE value 1654 for Italy, RMSE value 2031 for Spain, RSME value 971 for France. However, the study did not cover other statistical models. In addition, the studying duration was limited. The achieved RMSE Value for the three countries’ pandemic spread prediction was very high.

In [32], the authors proposed an excellent comparative study between statistical models (ARIMA and SVR) and three deep models (LSTM, BiLSTM, and GRU). Their study included the period from 22 January 2020 to 27 June 2020. Their study covered 10 different countries. The BiLSTM network achieved the best performance with RMSE value 0.007 for China.

In [33], the authors have proposed a deep assessment methodology and fractional calculus to predict the pandemic spread in 8 different countries (China, Italy, Turkey, Spain, France, Germany, UK, and the USA). The study included the period from 13 February 2020 to 19 June 2020. They built a deep assessment algorithm based on fractional calculus and deep LSTM. However, the daily confirmed cases curve is estimated to be a Gaussian function, which is not suitable for every country as they found. In [34], the authors have proposed a machine learning method based on exponential smoothing (ES) algorithm to predict the pandemic spread in 8 different countries (China, Italy, Turkey, Spain, France, Germany, UK, and the USA).

The study included the period from 22 January 2020 to 4 June 2020. The proposed algorithm is compared to other machine learning algorithms, such as linear regression (LR),
support vector machine (SVM), least absolute shrinkage, and selection operator (LASSO). Their study was applied on a global dataset, ignoring each country’s particular case study or comparing their findings with well-known statistical time series forecasting models. Their experiments showed that the ES performed its best performance when following it by LR and LASSO models. However, this strategy was more complex and required a high computational cost.

In [35], the authors proposed an SEIR model to predict COVID-19 confirmed cases. The study included the period from 22 February 2020 to 8 October 2020. Furthermore, the study included two countries, which are Egypt and Iraq. Their proposed model estimated that the peak value of the confirmed cases is 4254 cases in Iraq and 1534 cases in Egypt based on the Gaussian model, while their model estimated that the peak value is 490,900, and 105,000 in Iraq and Egypt, respectively based on the logistic model. However, their findings have not been compared with similar studies in the literature.

In [36], the authors proposed seven machine learning algorithms to predict the positive confirmed cases in Egypt. The study included the period from 15 February 2020, to 15 June 2020. They investigated the performance of different types of regression models, which are exponential polynomial, quadratic, third-degree, fourth-degree, fifth-degree, sixth-degree, and logit growth. However, their results were good, and we cannot generalize their study.

In [37], the authors introduced a COVID-19 spread prediction model based on google trend algorithm in India, USA, and the UK. The authors applied the study in the period from 24 February 2020 to 20 May 2020. The study included predicting the daily new cases, the cumulative cases, and the death cases. They optimized their proposed system based on Grey Wolf Optimizer (GWO) and compared their findings with the ARIMA prediction model.

In [38], the authors proposed a simple statistical model to predict positive confirmed cases globally. The authors applied the study in the period from 22 January 2020 to 21 May 2020. Their model has not been compared with other models. On the other hand, their forecasting results were negatively biased in the case of death cases predictions and positively biased when new spikes of death and confirmed cases happened. They mentioned that their forecasting model would perform better when the established wave remains stable.

In [39], the authors proposed a deep learning model to predict the confirmed cases in several countries, which are Brazil, India, Russia, South Africa, Mexico, Peru, Chile, Colombia, and Iran. The authors applied the study from 20 January 2020, to 3 August 2020. The deep learning models were based on multi-head attention and LSTM network. Then, the authors optimize their model hyper-parameters based on Bayesian optimization. Each country represents a particular case according to several characteristics, such as population, contact rate, number of tourists, opening and closing policies. Saudi Arabia’s population has reached about 34 million citizens as reported by the unified national platform [40]. Furthermore, there are millions of foreign workers that are existed in Saudi Arabia. As its importance to the muslims, there are about 2 million that are visiting Saudi Arabia every year to perform their Hajj. On the other hand, millions of Muslim people are visiting Saudi Arabia continuously whole the year to perform Umrah. Therefore, there is a real need to predict the future confirmed cases of COVID-19 to help policymakers make recommendations and procedures that can stop the spread of infectious cases. There are a few articles in the literature that have covered Saudi Arabia’s pandemic spread prediction.

In [41], the authors have proposed a statistical model based on ARIMA time-series forecasting to predict the COVID19 pandemic spread in Saudi Arabia. This study has covered the duration from 2 March 2020 to 20 April 2020. Their model achieved a lower RMSE value of 21.17. However, their model was examined for only the last 15 days. The authors did not present enough comparative analysis with other methods in the literature to prove their findings. They have reported only the actual cases vs. the predicted cases with no deep investigations.
In [42], the authors have proposed a deep learning algorithm based on the LSTM network to predict the COVID-19 pandemic spread in Saudi Arabia. This study has covered the duration from 2 March to 10 October 2020. The algorithm was examined at three different periods, and the authors have compared their findings with the other statistical models in the literature. However, they have achieved a high RMSE value, which reached 160.608 with confirmed cases prediction and 100.039 with death cases prediction.

In [43], the authors have proposed a system dynamic based on SIR and SIR-F models to predict the COVID-19 pandemic spread in Saudi Arabia. This study has covered the duration from 2 March to 12 June 2020. In addition, the study has investigated the COVID-19 pandemic spread for susceptible, infected and recovered people. However, the authors did not compare their findings with the other statistical or machine learning models in the literature.

In [44], the authors proposed a mathematical model for predicting the confirmed cases in Saudi Arabia based on fractal-fractional derivative. However, they investigate the COVID-19 cases for a brief duration from 1 March 2020 till 22 April 2021. They have not also investigated the model’s ability to be applied to other countries.

In [45], the authors proposed a SIR model to predict the cases in Saudi Arabia, the Philippines, Singapore, and Indonesia. They expand the study from 2 March to 23 December 2020. Their model achieved an R-squared value of 0.95. However, they did not utilize any of the error evaluation metrics to prove their findings.

In [46], the authors proposed a neural network with a self-organizing map for the spatial analysis of data. On the other hand, they utilized the fuzzy fractal technique to capture the temporal trends. They applied their model to Belgium, Italy, United States, and Mexico from 21 January to 30 January 2021. However, this period is concise to measure the model’s performance.

In [47], the authors proposed a hybrid mathematical model based on the short-term forecast (STF) and long-term forecast (LTF) models. Their system was applied to Jordan only. However, their model achieved an R-squared value of 0.84, which is very low related to the similar works in the literature.

In [48], the authors proposed a fractional SEIR-AHQ model to predict COVID-19 cases in Beijing, Chongqing, Tianjin, and Heilongjiang. They investigate the period from 22 January 2020 to 5 March 2020 based on the data of the first 10 (early stage), 20 (middle-stage), and 30 (late-stage) days, respectively. They proved that the non-medical intervention criteria play an essential role in COVID-19 control.

In [49], the authors proposed an LSTM deep learning model to predict COVID-19 spread cases in Egypt from 14 February 2020 to 15 August 2020. Unfortunately, their model was limited to only one country and has not been applied to other countries.

From the previous studies, there is more a real challenge to increase the studying period, investigate the first wave of COVID-19 pandemic spread at its starting and ending, increase R-Squared value and decrease the achieved RMSE value for the prediction results. Also, there is a real need to employ recent deep learning algorithms to increase forecasting performance.

The Encoder–Decoder long short-term memory (LSTM) was introduced for natural language processing (NLP) tasks. The Encoder–Decoder architecture is based on a recurrent neural network (RNN). It achieved a successful performance versus other methods in the literature, specifically in the area of text translation [50]. Recently, the Encoder–Decoder long short-term memory (LSTM) has been applied for several time series forecasting tasks, such as power consumption [51], metal temperature [52], air pollutant [53] behaviour prediction [54], and gas concentration [55]. However, the LSTM core for Encoder–Decoder architecture needs to be developed using recent deep units. Also, there is a real need to apply Encoder–Decoder architecture for pandemic spread prediction tasks.

In this paper, our contributions are as follows: we extend the studying period to cover the pandemic spread from its start point to almost its stopping in Saudi Arabia. We investigate the pandemic spread for (confirmed positive cases, death cases, and recovery
cases) in Saudi Arabia. We also investigated the pandemic spread during two periods at the first wave starting and ending. We proposed a deep network model based on encoder-decoder BiLSTM network to forecast the COVID-19 pandemic spread in Saudi Arabia. We optimized the hyper-parameters of our proposed network as follows: the utilized training optimizer, learning rate, units, and dropout ratio. We compared our proposed algorithm with more than fifteen time-series forecasting approaches; statistical, machine learning, and deep learning.

Our proposed system is compared with the other algorithms in the literature that have been applied to the same country in the literature. We examined the generalization of the capability of our proposed system to several countries. The proposed system will need to be applied to more countries to prove its superior performance. The hyper-parameters are critical parameters that need to be developed. Moreover, our proposed system has reflected its superior performance and achieved the best performance compared to the other literature methods.

3. Material and Methods

3.1. Material

In favor of the transparency of the health authorities in Saudi Arabia, there is a daily recording of the COVID-19 cases. The recording includes the confirmed cases, death cases, and recovered cases. The information of all Saudi cases exists on the Saudi ministry of health website (https://covid19.moh.gov.sa, accessed on 1 January 2021). All utilized data were acquired from this website, the existed interactive dashboards, and the available application program interface (API). We utilized the available dataset for the Saudi Arabia dataset at three dimensions of investigation; The first dimension is investigating all COVID-19 cases (confirmed, recovered, and death). The second dimension is specifying a short studying period at the COVID-19 pandemic starting from 2 March 2020 to 31 May 2020. The third dimension specifies long studying period at the COVID-19 pandemic ending from 2 March 2020 to 27 December 2020.

We split the available dataset into 80% for the training process and 20% for testing the proposed model for each studying period. For the short studying period, the training samples are from 22 January 2020 to 17 May 2020, and the testing samples are from 18 May 2020 to 31 May 2020 for about 14 days. For the long studying period, the training samples are from 22 January 2020 to 27 October 2020, and the testing samples are from 28 October 2020 to 27 December 2020 for about 60 days. This splitting percentage reflects the robustness of our proposed network more than the other works in the literature.

To extend our investigations to other countries, we utilized the dataset available at the data repository for the 2019 Novel Coronavirus Visual Dashboard operated by the Johns Hopkins University Center for Systems Science and Engineering (JHU-CSSE). We investigated both deaths and confirmed cases in the countries of interest.

3.2. Methods

In this work, we employ three kinds of methodologies: statistical models, machine learning models, and deep learning models. Our proposed system is described in Figure 1. First, we collect the available COVID-19 time series data set with it three categories (confirmed cases, death cases, and recovered cases) for Saudi Arabia is split into 80% for training and 20% for testing. Then, we utilize a data pre-processing stage, which includes reading CSV files and data standardization of the available data.

In the hyper-parameter selection stage, we select the optimal parameters from the previous studies and investigate the optimal hyper-parameters to train our proposed network. After the training process for each model, we evaluate the prediction results based on statistical performance indices and visualization plots. The Saudi COVID-19 time series data is employed to optimize our proposed network hyper-parameters. Finally, our proposed optimized network is examined by forecasting the global dataset’s death and confirmed cases for several countries.
3.2.1. Data Pre-Processing

After reading the COVID-19 time-series data from a comma-separated values (CSV) files, we apply the feature scaling stage to the input time series. There are two primary data feature scaling methodologies that are applied to the time-series data, which are normalization [56] and standardization [57]. These methodologies play a crucial role in overall system performance, especially the time series data [58]. This study investigates which feature scaling technique is more suitable with the COVID-19 data time series.

The feature scaling using normalization technique can be defined as follows:

$$Y_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

where $Y_{\text{norm}}$ represents the normalized data between 0 and 1 for input data $X$. $X_{\text{max}}$ and $X_{\text{min}}$ represents the maximum value and the minimum value of the input $X$, respectively.

The feature scaling using standardization technique can be defined as follows:

$$Y_{\text{stand}} = \frac{X - \mu}{\sigma}$$

where $Y_{\text{stand}}$ represents the standardized data for the input data $X$. $\mu$ represents the mean value of the given input $X$ and $\sigma$ represents the standard deviation the given input $X$.

3.2.2. Our Proposed Approach

To investigate the superior performance of our proposed approach, we assess the comparison between our proposed approach and the other COVID-19 forecasting algorithms in the literature, such as machine learning, deep learning, and statistical algorithms. The
machine learning regression algorithms are linear regression, support vector regression (SVR), lasso, Huber, RANSAC, TheilSen, and ElasticNet.

We set the parameters of the machine learning algorithms as in [34]. The deep learning algorithms applied previously for COVID-19 time series forecasting were LSTM, BiLSTM, GRU, and Encoder–Decoder BiLSTM. We set the hyper-parameters for each deep learning network as in [32,42]. For the statistical approach, we investigate the performance of three ARIMA models in the literature to investigate the COVID-19 spread in Saudi Arabia. We set the (P,Q,D), which controls the performance of ARIMA model as in [27,41,59].

The traditional RNNs predict each input sample corresponds to an output sample for the same time step. The Encoder–Decoder deep architecture was demonstrated to solve the sequence-to-sequence mapping models [55]. Therefore, it does not correlate the input time series correspond to the predicted time series since the inputs and the predicted samples are not correlated, and their lengths can be different. Therefore, it maps the time series samples of different lengths to each other.

The disadvantage of this model is the challenge to summarize a long sequence into a single vector, and the model often forgets the earlier parts of the input sequence when processing the last parts. Therefore, we develop the traditional Encoder–Decoder model by replacing the LSTM unit by BiLSTM unit model to overcome forgetting earlier samples in the sequence. The Encoder–Decoder deep architecture aims to extract more valuable data representation in compressed form, enabling the network to obtain the most discriminated features of the training input data [54].

An auto-encoder consists of encoding and decoding stages as shown in Figure 2. First, the encoder is utilized to read the input time series data and encode it into a fixed-length vector. The second stage is called a decoder. The decoder is utilized for decoding the fixed-length vector. Then, it produces the forecast sequence. Our proposed approach is based on Encoder-Decoder BiLSTM architecture. The proposed approach consists of a BiLSTM layer for both encoding and decoding layers, RepeatVector layer, dropout layer, and time distributed layer as shown in Figure 2. The Encoder–Decoder is based on Recurrent Neural Networks (RNNs). We employ the deep BiLSTM as a particular type of RNN for encoding and decoding procedures in our proposed approach.

RNNs are networks that are organized into successive layers, and each layer of neurons represents nodes [60]. RNNs are a vanilla neural network consisting of input layers, hidden layers, and output layers where the neurons connect. However, in RNNs, each neuron is assigned to a fixed time step. Furthermore, each neuron in the hidden layer is also connected in a time-dependent direction. Finally, each input and output neuron is attached to the hidden layer with its corresponding time step. RNNs have several advantages for time series forecasting: not significantly affected with missed values, tracking complex time series patterns, and modeling data flexibility as each sample depends on the previous one. However, the vanilla RNN cannot forecast long period time series.

Long short term memory (LSTM) is a special kind of RNN deep learning model that was developed by [61]. LSTM has the advantage of predicting long-term time series. This is because it uses a particular combination of hidden units, element-wise products, and aggregates within units to execute gates responsible for controlling memory cells. First, each cell is created to hold information without alteration for long periods. Then, the learnable weight values must be updated to forecast the next step, which demands the preservation of information from the initial steps. Unfortunately, the simple structure of RNN makes it learn only a limited number of short-term relationships, and it does not perform well during forecasting the long-term series. However, LSTM can well forecast long-term series, and it overcomes the vanishing gradient that happens in the state-of-the-art RNN.
LSTM consists of a cell current state with three gates: input, output, and forget, as shown in Figure 2. The cell state is the network memory responsible for retrieving the sample along the input sequence. The input gate determines the relevant information to add the previous time steps. The forget gate keeps the previous time step and determines how the previous memory remembers and forgets. Finally, the output gate decides the value of the current time step.

The forget gate equation is computed as follows:

\[ f(t) = \sigma(x(t)U_f + h(t-1)W_f) \]  

(3)

The input gate equation is computed as follows:

\[ i_1(t) = \sigma(x(t)U_i + h(t-1)W_i) \]  

(4)

\[ i_2(t) = tanh(x(t)U_g + h(t-1)W_g) \]  

(5)

\[ i(t) = i_1(t) \ast i_2(t) \]  

(6)

The cell state equation is computed as follows:

\[ C(t) = \sigma(f(t) \ast C(t-1) + i(t)) \]  

(7)

The output gate equation is computed as follows:

\[ O(t) = \sigma(x(t) \ast U_O + h(t-1)W_O) \]  

(8)

\[ h(t) = tanh(C_t) \ast O(t) \]  

(9)
where \( i(t), f(t), \) and \( O(t) \) are the input, forget, and output gates at time \( t \), respectively; \( W_i \) and \( U_i \) represent the hidden layer weights that are the input of input gate; \( W_f, U_f \) represent the weights of the hidden layer corresponding to the forget gate; \( W_o \) and \( U_o \) represent the weights of the hidden layer corresponding to the output gate; and \( C_t \) and \( h_t \) represent the outcome of the cell and the outcome of the layer, respectively [62].

One of the few limitations founded on the LSTM cell is that it cannot see the future time sample. To overcome this limitation, in [63], the authors introduced the bidirectional-LSTM (BiLSTM) as shown in Figure 2. For the input sequence \( X(t) \) with time step \( t \), the output sequence \( y(t) \) is calculated from \( \overrightarrow{h}(t) \) in the forward direction, and \( \overleftarrow{h}(t) \) in the backward direction. Therefore, we replace the LSTM unit of the traditional Encoder–Decoder architecture with a BiLSTM unit in our proposed approach to take the advantages of BiLSTM over the LSTM unit.

The RepeatVector layer is essential to fit the encoder data dimension with the decoder data dimension. The RepeatVector is employed to repeat the one fixed-length data for each time step in the output sequence.

In our proposed approach, the RepeatVector layer repeats the outputs of the previous encoder stage for one time. Therefore, for the input shape with size \((1, 64)\), its output shape after the RepeatVector will be \((1, 1, 64)\), since the inputs were repeated only once. The input should be at least 3D, and the dimension of the index one will be considered to be the temporal dimension. Therefore, the 3D output can be processed later through the decoding stage.

After the decoding stage, we apply the dropout layer to the decoding stage output. The dropout layer was introduced by [64] to increase the performance of the neural networks. This happens through dropping some of the weights inside the network as in Equation (10). In addition, the connections of the drop weights are skipped, as shown in Figure 3. The dropout technique supports the regularization strategy, reducing the risk of co-adaptation, and reducing the over-fitting. In this work, we investigate the optimal drop percentage and its effect on our proposed approach.

\[
\hat{w}_{ij} = \begin{cases} 
  w_{ij}, & \text{with } P(c) \\
  0, & \text{otherwise} 
\end{cases}
\]

Figure 3. (a) A network without dropout and (b) a network with dropout.

The Time-Distributed layer performs the same dense layer function to the output of the dropout layer for one time step at a time. Time-Distributed layer processes each received input sample using the dense layer. Therefore, we apply a dense layer to every
temporal slice of the input data with an index considered to be the temporal dimension. For the COVID-19 case time-series forecasting model, the dense layer output is set to be 1.

3.2.3. Prediction Results Evaluation Criteria

We assess the statistical performance in terms of three error measures, which are the mean absolute error (MAE), root mean square error (RMSE), and mean square error (MSE). The three metrics are defined in Equations (11)–(13), respectively. Moreover, we employ the coefficient of determinations (R-Squared) specified in this section for performance evaluation as in Equation (15). The low MSE, RMSE, and MAE values indicate the best forecasting performance. On the other hand, the high R-Squared value indicates the best forecasting performance. We also utilize the visualization plots for training, validation, and prediction plots. This proves the forecasting results trend stability with time-series data variation.

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |C_i - \hat{C}_i| 
\]

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (C_i - \hat{C}_i)^2} 
\]

\[
\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (C_i - \hat{C}_i)^2 
\]

\[
\text{R-Squared} = 1 - \frac{\frac{1}{N} \sum_{i=1}^{N} (C_i - \hat{C}_i)^2}{\frac{1}{N} \sum_{i=1}^{N} (C_i - \bar{C}_i)^2} 
\]

where \(C_i\) represents the true value of the COVID-19 cases, \(\hat{C}_i\) represents the predicted value of the COVID-19 cases at time \(i\), \(\bar{C}_i\) is defined as \(\bar{C}_i = \frac{1}{N} \sum_{i=1}^{N} C_i\), and \(N\) represents the number of samples.

Furthermore, we utilize the mean absolute scaled error (MASE), which is a scale-free error metric [65]. The low MASE value indicates the best forecasting performance. The error is represented in the MASE metric as a ratio compared to a baseline average error as in Equation (15).

\[
\text{MASE} = \frac{1}{I} \sum_{i=1}^{I} |q(t_i)| 
\]

\[
q(t_i) = \frac{e(t_i)}{\frac{1}{N} \sum_{n=2}^{N} |X(t_n) - X(t_{n-1})|} 
\]

where \(q(t_i)\) represents the scaled error value, \(e(t_i)\) represents the error value at time \(t_i\), is defined as \(e(t_i) = (C_i - \hat{C}_i)\), \(I\) are steps of forecasts, and \(X(t_n) = \{1, 2, \ldots, N\}\) are the existing observations that used for the training process of the forecasting model.

4. Results

During our experiment, we utilized the software Python 3.8 in the Spyder platform. The hardware system contained Quad-Core 2.9 GHz Intel i7 with 16 GB RAM. The GPU computation was performed through NVIDIA Ge-Force 840M with 4 GB built-in RAM and compute-capability 5.0.

We organized our experiments as follows: Experiment 1 aimed to find the optimal hyper-parameters for our proposed model. It also aims to find the best optimizer to train our proposed model. Experiment 2 examined the effect of features scaling strategy on the proposed model. Experiment 3 proposed a comparison between our proposed method and fifteen forecasting models from the literature. Furthermore, most of these models were employed in forecasting the Saudi Arabia COVID-19 spread. Experiments 1, 2, 3, and
4 were based on the Saudi Arabia COVID-19 dataset for confirmed, recovered, and death cases. Finally, Experiment 4 aims to prove the ability of our proposed model to be applied to other countries, including Brazil, India, South Africa, Germany, Italy, Turkey, and Spain.

4.1. Experiment 1

In Experiment 1, we investigate the optimal parameters to be utilized in the training process of our model. In addition, we investigate the effect of the optimization algorithms of our proposed model. We employ the Saudi Arabia COVID-19 dataset for this target.

Generally speaking, the pandemic wave starts with the first confirmed case reported, and it ends with about zero cases reported before the regrowth of confirmed cases again [66]. For example, in Saudi Arabia, the first case of COVID-19 was reported on 2 March 2020 [67]. With the end of December 2020, the first wave of COVID-19 spread was almost ended as reported in [68–70]. Therefore, during the studying period from 2 March 2020 until 27 December 2020, there was only one COVID-19 spread wave in Saudi Arabia, as reported in the literature. In our experiments, we investigate the model performance during two different time series in the start of the first wave and the ending of the first wave. Furthermore, we also investigate the confirmed, recovered, and death cases in all periods.

To optimize the hyper-parameters of our model, we investigate three model parameters: the hidden units, initial learning rate value, and drop percentage as in [42]. First, we fix the initial learning rate at 0.0005 and the drop percentage at 0.2. Then, we tune the number of hidden units with (16, 32, 64, and 128). Second, we fix the hidden unit at 128 and the drop percentage at 0.2. Then, we tune the initial learning rate value with (0.01, 0.001, 0.005, and 0.0005). Thirdly, we fix the units at 128 and the initial learning rate at 0.0005. Then, we tune the dropout percentage with values (0.1, 0.2, 0.3, and 0.4). We set the (Adaptive Moment Estimation Algorithm) Adam optimizer as the training optimizer for this experiment as it proved its superior performance for forecasting the same task [32,42,66]. Therefore, we can deliver the optimal parameters for our proposed model.

For the forecasting of confirmed cases of the COVID-19 first wave starting as shown in Table S1, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.9929$, $9.681 \times 10^2$, $8.031 \times 10^2$, and $9.371 \times 10^5$, respectively. For the forecasting of confirmed cases of the COVID-19 first wave ending as shown in Table S1, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage.

It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.9981$, $2.001 \times 10^2$, $1.711 \times 10^2$, and $3.981 \times 10^4$, respectively. The superior performance of our proposed model to forecast the confirmed cases for along 14 days during the first wave start is shown in Figure 4a. The superior performance of our proposed model to forecast the confirmed cases for along 60 days during the first wave end is shown in Figure 4b.

For the forecasting of recovered cases of COVID-19 first wave starting as shown in Table S2, our proposed model achieved the best performance with the following parameters: 16 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.9929$, $2.271 \times 10^3$, $2.081 \times 10^3$, and $5.171 \times 10^6$, respectively. For the forecasting of recovered cases of COVID-19 first wave ending as shown in Table S2, our proposed model achieved the best performance with the following parameters: 16 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.9981$, $2.451 \times 10^2$, $2.141 \times 10^2$, and $6.021 \times 10^4$, respectively.

The performance of our proposed model to forecast the recovered cases for along 14 days during the first wave starting is shown in Figure 4c. We noticed that the instability of the prediction line vs. the test line. The superior performance of our proposed model to forecast the recovered cases for along 60 days during the first wave ending is shown in Figure 4d.
For the forecasting of death cases of COVID-19 first wave starting as shown in Table S3, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9954, 3.96, 3.40, and 10.56, respectively. For the forecasting of death cases of COVID-19 first wave ending as shown in Table S3, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9989, 8.00, 6.55, and 60.39, respectively. The superior performance of our proposed model to forecast the death cases for along 14 days during the first wave starting is shown in Figure 4e. The superior performance of our proposed model to forecast the death cases for along 60 days during the first wave end is shown in Figure 4f.

In the literature, there is no deep investigation of the optimizer effect on the forecasting model [32,42]. Therefore, we investigate the effect of the optimization algorithms of our proposed model as shown in Tables S4 and S5 through two studies.

In study 1, we investigated the performance of different optimizers with the median hyper-parameter values, which are 48 hidden units, initial learning rate 0.003, and dropout percentage 0.25. We employ the median values introduced in the first experiment to have a fair comparison between the different optimizers. In study 2, we investigate the performance of our proposed optimal hyper-parameters, which are 128 hidden units, initial learning rate 0.0005, and dropout percentage 0.2 with different optimizers. These optimization algorithms are Root Mean Square Propagation (RMSprop), Stochastic Gradient Descent (SGD), Adam, Adaptive Maximum (Adamax), Adaptive Delta (Adadelta), and Adaptive Gradient (Adagrad).

We selected the confirmed cases during the first wave starting and ending to investigate the optimizer effect. In study 1, Adam optimizer achieved the best forecasting performance for forecasting the confirmed cases during the first wave start. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9908, $1.101 \times 10^3$, $9.281 \times 10^2$, and $1.221 \times 10^6$, respectively. For the forecasting of the confirmed cases during the first wave ending, Adam optimizer achieved the best forecasting performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9789, $6.581 \times 10^2$, $6.411 \times 10^2$, and $4.321 \times 10^5$, respectively.

In study 2, we investigated the performance of different optimizers with the optimal hyper-parameters values, which are 128 hidden units, initial learning rate 0.0005, and dropout percentage 0.2. Adam optimizer achieved the best forecasting performance for forecasting the confirmed cases during the first wave start. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9929, $9.681 \times 10^2$, $8.031 \times 10^2$, and $9.371 \times 10^5$, respectively. For the forecasting of the confirmed cases during the first wave ending, Adam optimizer achieved the best forecasting performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9981, $2.001 \times 10^2$, $1.711 \times 10^2$, and $3.981 \times 10^4$, respectively. Adamax optimizer achieved the second rate. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9783, $6.661 \times 10^2$, $6.381 \times 10^2$, and $4.431 \times 10^5$, respectively. RMSprop optimizer achieved the third rate. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9783, $6.661 \times 10^2$, $6.381 \times 10^2$, and $4.431 \times 10^5$, respectively. However, SGD, Adadelta, and Adagrad achieved bad performance during our forecasting task. From this experiment, we concluded the importance of the three investigated hyper-parameters to control the performance of the proposed forecasting model. We noticed that the initial learning rate value variation was a critical parameter on our proposed model’s performance. However, the variation of hidden units or dropout percentage slightly varied our proposed model’s performance. It is recommended to set the parameter for our model with 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage.

We also noticed the stability of our model between the forecasting of the short time series during the COVID-19 first wave starting and forecasting of the long time series
during the COVID-19 first wave ending. The exact optimal hyper-parameters were the same for both first wave start and end.

Generally, the forecasting performance during the first wave end was better than during the first wave start. This can be explained by the high variation of all cases during the first wave start. Based on several statistical assessments, the forecasting performance was similar for the confirmed, recovered, and death cases with shallow variance errors.

We also concluded that the Adam optimizer was the best optimization algorithm to train our proposed model. A similar performance between the two studies was noticed for the performance of different optimizers. The Adamax optimizer achieved the second rank among optimizers. However, SGD, Adadelta, and Adagrad achieved a lousy performance during our forecasting task. We also noticed that the crucial role of investigating the optimization algorithms for deep learning models.

Figure 4. Cont.
Figure 4. The forecasting results of COVID-19 spread for our proposed method in Saudi Arabia. (a) Confirmed cases forecasting results in the start of the COVID-19 first wave in Saudi Arabia. (b) Confirmed cases forecasting results in the ending of the COVID-19 first wave in Saudi Arabia. (c) Recovered cases forecasting results in the start of the COVID-19 first wave in Saudi Arabia. (d) Recovered cases forecasting results in the ending of the COVID-19 first wave in Saudi Arabia. (e) Death cases forecasting results in the start of the COVID-19 first wave in Saudi Arabia. (f) Death cases forecasting results in the ending of the COVID-19 first wave in Saudi Arabia.

4.2. Experiment 2

In this experiment, we investigate the performance of our proposed model based on different feature scaling techniques. We investigate the features scaling techniques normalization and standardization. Based on several statistical assessment criteria, the features standardization is better than the features normalization technique as shown in Figure 5a,b. We select the confirmed cases during the first wave starting and ending to investigate the features scaling effect. We set the hyper-parameters as 128 hidden units, initial learning rate 0.0005, and dropout percentage 0.2.

We noticed that the standardization has a higher R-Squared than the normalization technique. We also noticed that the standardization has lower RMSE, MSE, and MAE than the normalization technique. Therefore, we conclude that the standardization is much better than the normalization technique for our proposed model.

Figure 5. Our proposed model performance with different features scaling techniques for confirmed cases in Saudi Arabia. (a) Our proposed model performance during the first wave starting. (b) Our proposed model performance during the first wave ending.
4.3. Experiment 3

This experiment compares our proposed approach and four deep learning methods, three statistical ARIMA models, seven machine learning algorithms, and one prophet model. ARIMA model is a statistical time-series method that has been extensively utilized to forecast infectious diseases. It gathers the Autoregressive (AR) model and Moving Average (MA) with integration based on the decomposition method. In which all current and historical residual series values in the present time series are expressed linearly [41]. The ARIMA model is controlled as ARIMA (p, d, q) values. p represents the auto-regressive seasonal order, d represents the autoregressive non-seasonal order, and q represents the non-seasonal moving average order. The prophet model was utilized to forecast COVID-19 cases in Saudi Arabia [71]. The prophet model utilized the Fourier spectral data to compute the seasonality impact for forecasting time series data in the medium-term and long-term [72].

This model has been successfully utilized in Saudi Arabia to predict COVID-19 cases. The supervised machine learning algorithms were successfully applied to predict COVID-19 cases based on regression models [34]. The deep learning RNN networks can automatically capture seasonal and trends characteristics of the time series [71]. In this experiment, we utilized four different deep architectures for time-series forecasting tasks: LSTM, BiLSTM, GRU, and Encoder–Decoder LSTM.

For each model parameters, we set their values as in the literature. The BiLSTM model was applied to Saudi Arabia [42]. The BiLSTM model hyper-parameters were set as 100 hidden units and 0.005 initial learning rate value. The GRU and LSTM hyper-parameters were generally applied in several countries [32]. They set the parameters as 128 hidden units, 0.001 initial learning rate value. Furthermore, we compared our proposed method based on the Encoder–Decoder BiLSTM model and the Encoder–Decoder LSTM one.

We set the optimal parameters as 128 units, 0.2 drop percentage, and 0.0005 initial learning rate for both models. In this way, we have a fair comparison to study the effect of using BiLSTM unit in the encoder decoder deep architecture. We utilized the Adam optimizer for all deep learning models training process. The three ARIMA models were applied to Saudi Arabia. Each model’s parameters were investigated previously in these studies. The (p,q,d) values were set in ARIMA Model1 [41] as (2,1,1) in ARIMA Model2 [27] as (1,1,1) in ARIMA Model3 [59] as (0,2,0). The machine learning models setup parameters were utilized as in [34] and [32].

For the forecasting of confirmed cases of COVID-19 first wave starting as shown in Table 1, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9929, $9.681 \times 10^2$, $8.031 \times 10^2$, and $9.371 \times 10^5$, respectively. The Encoder–Decoder LSTM model achieved the second rank. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9812, $1.5811 \times 10^3$, $1.5311 \times 10^3$, and $2.501 \times 10^6$, respectively. GRU model achieved the third rank. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9497, $2.041 \times 10^3$, $6.661 \times 10^6$, and $2.581 \times 10^3$, respectively.

ARIMA Model3 achieved the third rank among all compared models and the first rank in all ARIMA Models. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9143, $3.501 \times 10^3$, $3.141 \times 10^3$, and $1.231 \times 10^7$, respectively. Among deep learning models, the LSTM model achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.562, $5.431 \times 10^3$, $4.711 \times 10^3$, and $2.951 \times 10^7$, respectively. For all machine learning models, we noticed bad forecasting results with high error values and negative coefficients of determination. TheilSen regression algorithm achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of $-212.97$, $9.181 \times 10^4$, $8.141 \times 10^4$, and $8.431 \times 10^9$, respectively.
For the forecasting of the confirmed cases of COVID-19 first wave ending as shown in Table 2, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RSME, MAE, and MSE of 0.9981, 21 × 10⁻⁵, 1.711 × 10⁵, and 3.981 × 10⁴, respectively. BiLSTM model achieved the second rank. It demonstrated an R-Squared, RSME, MAE, and MSE of 0.9886, 4.821 × 10⁻⁵, 4.511 × 10⁴, and 2.321 × 10⁻⁵, respectively. In spite of, BiLSTM model is slightly higher than our proposed model, the BiLSTM model has higher error values.

### Table 2. COVID-19 spread forecasting results in countries with double spread wave based statistical evaluation criteria.

| Country | Cases          | Period       | MSE     | MAE     | RMSE    | MAE     | R_Squared |
|---------|----------------|--------------|---------|---------|---------|---------|-----------|
| Germany | Death Cases    | Period 1     | 5.701 × 10⁻² | 2.081 × 10⁻¹ | 2.391 × 10⁻¹ | 12.2 | 0.9881 |
|         |                | Period 2     | 9.891 × 10⁻⁵ | 8.151 × 10⁻² | 9.941 × 10⁻² | 83.27 | 0.9723 |
|         | Confirmed Cases| Period 1     | 2.931 × 10⁻⁵ | 4.541 × 10⁻² | 5.411 × 10⁻² | 8.79 | 0.9633 |
|         |                | Period 2     | 4.701 × 10⁻⁴ | 6.821 × 10⁻⁴ | 6.851 × 10⁻⁴ | 45.3 | 0.9595 |
| Italy   | Death Cases    | Period 1     | 4.091 × 10⁻³ | 4.931 × 10⁻¹ | 6.391 × 10⁻¹ | 0.2  | 0.9904 |
|         |                | Period 2     | 1.541 × 10⁻⁵ | 3.021 × 10⁻³ | 3.921 × 10⁻² | 4.87 | 0.9987 |
|         | Confirmed Cases| Period 1     | 2.581 × 10⁻⁵ | 4.581 × 10⁻² | 5.081 × 10⁻² | 0.06 | 0.9742 |
|         |                | Period 2     | 3.951 × 10⁻⁶ | 6.181 × 10⁻⁴ | 6.281 × 10⁻⁴ | 33.09 | 0.9797 |
| Turkey  | Death Cases    | Period 1     | 3.651 × 10⁻² | 1.601 × 10⁻¹ | 1.911 × 10⁻¹ | 2.57 | 0.9864 |
|         |                | Period 2     | 9.601 × 10⁻⁴ | 2.271 × 10⁻² | 3.101 × 10⁻² | 0.69 | 0.9884 |
|         | Confirmed Cases| Period 1     | 8.221 × 10⁻⁵ | 7.531 × 10⁻² | 9.071 × 10⁻² | 0.87 | 0.976 |
|         |                | Period 2     | 1.221 × 10⁻⁴ | 6.591 × 10⁻⁴ | 1.101 × 10⁻⁴ | 47.88 | 0.9731 |
| Spain   | Death Cases    | Period 1     | 3.321 × 10⁻³ | 4.911 × 10⁻¹ | 5.761 × 10⁻¹ | 0.96 | 0.9898 |
|         |                | Period 2     | 5.721 × 10⁻⁵ | 5.801 × 10⁻² | 7.561 × 10⁻² | 3.47 | 0.9707 |
|         | Confirmed Cases| Period 1     | 8.461 × 10⁻⁵ | 8.381 × 10⁻² | 9.201 × 10⁻² | 0.71 | 0.9141 |
|         |                | Period 2     | 2.421 × 10⁻⁴ | 4.871 × 10⁻⁴ | 4.921 × 10⁻⁴ | 7.93 | 0.9418 |
an R-Squared, RMSE, MAE, and MSE of $-2.1222, 7.991 \times 10^3, 7.961 \times 10^3,$ and $6.381 \times 10^7$, respectively. For all examined machine learning models, we noticed bad forecasting results with high error values and negative coefficients of determination.

In addition, the prophet model achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of $-6090.89, 3.561 \times 10^5, 3.561 \times 10^5,$ and $1.271 \times 10^{11}$, respectively. The performance of our proposed model to forecast the confirmed cases for along 14 days during the first wave start is shown in Figure 6a. We noticed superior performance of our proposed model to forecast the confirmed cases for along 60 days during the first wave end as shown in Figure 6b.

For the forecasting of recovered cases of COVID-19 first wave starting are shown in Table 3. Our proposed model achieved the best performance with the following parameters: 16 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.965, 2.271 \times 10^3, 2.081 \times 10^3,$ and $5.171 \times 10^6$, respectively. The GRU model achieved the second rank. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.8713, 4.411 \times 10^3, 4.361 \times 10^3,$ and $1.941 \times 10^7$, respectively. The Encoder–Decoder LSTM model achieved a good performance relative to the other methods. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.8653, 4.511 \times 10^3, 3.8711 \times 10^3,$ and $2.031 \times 10^7$, respectively.

ARIMA Model3 achieved the third rank among all compared models and the first rank in all ARIMA Models. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.8234, 5.511 \times 10^3, 4.341 \times 10^3,$ and $3.041 \times 10^7$, respectively. Among deep learning models, the LSTM model achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of $-0.553, 9.721 \times 10^3, 7.871 \times 10^3,$ and $9.451 \times 10^7$, respectively. Among machine learning models, the Lasso regression model achieved the highest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.773, 3.921 \times 10^3, 3.271 \times 10^3,$ and $1.541 \times 10^7$, respectively. We noticed bad forecasting results with high error values and negative coefficients of determination. Prophet model achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of $-9.3176, 4.211 \times 10^4, 4.001 \times 10^4,$ and $1.771 \times 10^9$, respectively.

| Forecasting Model | COVID-19 First Wave Starting | COVID-19 First Wave Ending |
|-------------------|-----------------------------|-----------------------------|
|                  | MSE     | MAE     | RMSE    | R-squared | MSE     | MAE     | RMSE    | R-squared |
| Prophet Model     | $1.771 \times 10^6$ | $4.001 \times 10^3$ | $4.211 \times 10^4$ | $-9.3176$ | $1.191 \times 10^3$ | $3.451 \times 10^3$ | $3.451 \times 10^3$ | $-3067.49$ |
| ARIMA_Model1      | $5.541 \times 10^6$ | $5.951 \times 10^3$ | $7.441 \times 10^3$ | $0.6777$ | $3.141 \times 10^3$ | $4.271 \times 10^3$ | $5.611 \times 10^3$ | $0.1881$ |
| ARIMA_Model2      | $3.681 \times 10^7$ | $4.731 \times 10^3$ | $6.061 \times 10^3$ | $0.7862$ | $6.811 \times 10^3$ | $6.401 \times 10^3$ | $8.251 \times 10^3$ | $-0.7888$ |
| ARIMA_Model3      | $3.041 \times 10^7$ | $4.341 \times 10^3$ | $5.511 \times 10^3$ | $0.8234$ | $1.111 \times 10^3$ | $8.541 \times 10^3$ | $1.051 \times 10^4$ | $-1.8675$ |
| Lasso             | $1.541 \times 10^7$ | $3.271 \times 10^3$ | $3.921 \times 10^3$ | $0.7733$ | $2.681 \times 10^3$ | $1.641 \times 10^3$ | $1.641 \times 10^3$ | $-976.72$ |
| RANSACRegressor   | $1.811 \times 10^8$ | $1.341 \times 10^4$ | $1.351 \times 10^4$ | $-1.6702$ | $1.381 \times 10^3$ | $3.711 \times 10^4$ | $3.711 \times 10^4$ | $-49.2069$ |
| HuberRegressor    | $2.051 \times 10^7$ | $3.731 \times 10^3$ | $4.531 \times 10^3$ | $0.6984$ | $1.051 \times 10^4$ | $3.231 \times 10^3$ | $3.231 \times 10^3$ | $-37.0974$ |
| LinearRegression  | $2.381 \times 10^7$ | $3.991 \times 10^3$ | $4.881 \times 10^3$ | $0.649$ | $1.381 \times 10^3$ | $3.711 \times 10^4$ | $3.711 \times 10^4$ | $-49.2069$ |
| SVR_linearity     | $3.831 \times 10^7$ | $4.961 \times 10^3$ | $6.191 \times 10^3$ | $0.4357$ | $6.911 \times 10^3$ | $2.631 \times 10^4$ | $2.631 \times 10^4$ | $-24.1915$ |
| ElasticNet        | $1.591 \times 10^8$ | $1.261 \times 10^4$ | $1.261 \times 10^4$ | $-1.3486$ | $9.541 \times 10^4$ | $9.761 \times 10^4$ | $9.771 \times 10^4$ | $-346.681$ |
| TheilSenRegressor | $4.141 \times 10^7$ | $5.191 \times 10^3$ | $6.431 \times 10^3$ | $0.3905$ | $5.421 \times 10^3$ | $7.361 \times 10^3$ | $7.361 \times 10^3$ | $-196.374$ |
| GRU               | $1.941 \times 10^8$ | $4.361 \times 10^3$ | $4.411 \times 10^3$ | $0.8713$ | $1.951 \times 10^3$ | $4.410 \times 10^3$ | $4.410 \times 10^3$ | $0.4877$ |
| BiLSTM            | $1.211 \times 10^8$ | $1.011 \times 10^4$ | $1.101 \times 10^4$ | $0.197$ | $8.571 \times 10^3$ | $2.471 \times 10^3$ | $2.931 \times 10^3$ | $0.9978$ |
| LSTM              | $9.451 \times 10^7$ | $7.871 \times 10^3$ | $9.721 \times 10^3$ | $-0.553$ | $9.401 \times 10^3$ | $2.781 \times 10^3$ | $3.071 \times 10^3$ | $0.7271$ |
| Encoder–Decoder-LSTM | $2.031 \times 10^7$ | $3.871 \times 10^3$ | $4.511 \times 10^3$ | $0.8653$ | $5.311 \times 10^3$ | $2.281 \times 10^3$ | $2.301 \times 10^3$ | $0.8608$ |
| **Our Proposed Model** | $5.171 \times 10^6$ | $2.081 \times 10^3$ | $2.271 \times 10^3$ | $0.9657$ | $6.021 \times 10^4$ | $2.141 \times 10^2$ | $2.451 \times 10^2$ | $0.9984$ |

For the forecasting of recovered cases of COVID-19 first wave ending as shown in Table 3, our proposed model achieved the best performance with the following parameters: 16 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of $0.9984, 2.451 \times 10^2, 2.141 \times 10^2,$ and $6.021 \times 10^4$, respectively. BiLSTM model achieved the second rank. It demonstrated an R-Squared,
The Encoder–Decoder LSTM model achieved a low performance among all compared models in forecasting of recovered cases of long time-series. The best ARIMA Model is Model1. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9671, 11.7139, 7.1448, and 137.216, respectively. We noticed bad forecasting results for all machine learning models with high error values and negative coefficients of determination.

In addition, the ARIMA Model1 achieved the first rank in all ARIMA Models. However, it is low compared to the other models. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9743, 9.3672, 9.3539, and 87.7449, respectively. Among deep learning models, the LSTM model achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.1881, 5.611 × 10^3, 3.451 × 10^3, and 6.811 × 10^7, respectively. For all machine learning models, we noticed bad forecasting results with high error values and negative coefficients of determination.

For the forecasting of death cases of COVID-19 first wave ending as shown in Table 4, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9978, 2.931 × 10^2, 2.471 × 10^2, and 8.571 × 10^5, respectively. The Encoder–Decoder LSTM model achieved the third rank. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9984, 2.451 × 10^2, 3.851 × 10^2, and 6.021 × 10^4, respectively. In addition, we noticed bad forecasting results for all machine learning models with high error values and negative coefficients of determination.

For the forecasting of death cases of COVID-19 first wave starting as shown in Table 4, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9671, 11.7139, 7.1448, and 137.216, respectively. Among deep learning models, the LSTM model achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9671, 11.7139, 7.1448, and 137.216, respectively. We noticed bad forecasting results with high error values and negative coefficients of determination. However, the Lasso regression model achieved acceptable results. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.45, 32.93, 33.92, and 8.431 × 10^6, respectively. For all machine learning models, we noticed bad forecasting results with high error values and negative coefficients of determination. In addition, the prophet model achieved the lowest performance. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9978, 2.931 × 10^2, 2.471 × 10^2, and 8.571 × 10^4, respectively.

For the forecasting of death cases of COVID-19 first wave starting as shown in Table 4, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9978, 2.931 × 10^2, 2.471 × 10^2, and 8.571 × 10^4, respectively. The BiLSTM model achieved the second rank. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9984, 2.451 × 10^2, 3.511 × 10^2, and 6.811 × 10^7, respectively. The performance of our proposed model to forecast the death cases of COVID-19 first wave ending as shown in Figure 6d.

For the forecasting of death cases of COVID-19 first wave starting as shown in Table 4, our proposed model achieved the best performance with the following parameters: 128 hidden units, 0.0005 initial learning rate, and 0.2 dropout percentage. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9978, 2.931 × 10^2, 2.471 × 10^2, and 8.571 × 10^4, respectively.
cases for along 14 days during the first wave starting is shown in Figure 6e. We noticed the superior performance of our proposed model to forecast the confirmed cases for along 60 days during the first wave ending as shown in Figure 6f.

**Table 4.** A comparative analysis of our proposed approach with the previous methods based statistical evaluation criteria for death cases in Saudi Arabia.

| Forecasting Model       | COVID-19 First Wave Starting | COVID-19 First Wave Ending |
|-------------------------|-----------------------------|-----------------------------|
|                         | MSE  | MAE  | RMSE | R-squared | MSE  | MAE  | RMSE | R-squared |
| Prophet Model           | 80.698 | 276.6609 | 284.0755 | -18.3656 | 3.281 × 10^7 | 5725.99 | 5731.193 | -549.971 |
| ARIMA_Model1            | 165.524 | 8.139 | 12.8656 | 0.9603 | 1.361 × 10^4 | 85.9144 | 116.7805 | 0.7712 |
| ARIMA_Model2            | 155.3177 | 7.9651 | 12.4627 | 0.9627 | 1.111 × 10^4 | 73.9975 | 105.474 | 0.8134 |
| ARIMA_Model3            | 137.216 | 7.1448 | 11.7139 | 0.9671 | 4.911 × 10^4 | 167.2461 | 221.6259 | 0.1761 |
| Lasso                   | 1150.8528 | 32.9337 | 33.9242 | 0.4582 | 8.651 × 10^6 | 2936.2 | 2941.442 | -202.413 |
| RANSACRegressor         | 2396.0025 | 47.5562 | 48.949 | -0.128 | 5.061 × 10^4 | 703.9295 | 711.3375 | -10.8963 |
| HuberRegressor          | 3396.2835 | 56.6015 | 58.2776 | -0.5989 | 5.091 × 10^4 | 705.7586 | 713.593 | -10.9718 |
| LinearRegression        | 4308.111 | 63.7156 | 65.6362 | -1.0281 | 5.061 × 10^4 | 703.9295 | 711.3375 | -10.8963 |
| SVR_linear              | 4769.5801 | 67.4808 | 69.0621 | -1.2454 | 5.501 × 10^5 | 734.0821 | 741.8875 | -11.94 |
| ElasticNet              | 5292.7366 | 71.4414 | 72.7512 | -1.4916 | 3.461 × 10^6 | 1858.153 | 1860.545 | -80.384 |
| TheilSenRegressor       | 5913.6893 | 75.4037 | 76.9005 | -1.784 | 8.541 × 10^5 | 919.0879 | 923.8529 | -19.0662 |
| GRU                     | 32.7193 | 4.8849 | 5.7201 | 0.9904 | 5.921 × 10^4 | 240.6431 | 243.2687 | -0.0133 |
| BiLSTM                  | 677.8149 | 24.9696 | 26.0349 | 0.8016 | 4.241 × 10^3 | 5.7041 | 6.512 | 0.9993 |
| LSTM                    | 1030.9777 | 30.1047 | 32.1088 | 0.0894 | 4.121 × 10^3 | 52.8893 | 64.1746 | 0.9192 |
| Encoder–Decoder-LSTM   | 87.7449 | 9.3539 | 9.3672 | 0.97 | 1.431 × 10^4 | 119.71 | 119.73 | 0.754 |
| Our Proposed Model      | 16.1006 | 3.1147 | 4.0126 | 0.9953 | 6.391 × 10^3 | 6.5497 | 7.9967 | 0.999 |

From these results, we noticed that our proposed encoding–decoding deep learning network based on BiLSTM was better than the other deep learning networks. Moreover, our proposed method achieved a higher performance than the traditional Encoder–Decoder LSTM model. We noticed that the performance of the Encoder–Decoder model during the first wave starting was better than its performance during the first wave ending. This can be explained as, in each time step, our proposed method generates a point in forecasting, and it searches for a set of time series positions in the source of the time series where the most relevant information is intensified.

Thus, the model can forecast a target time series point based on the context vectors associated with these source positions and all the previous generated target points. We noticed that the BiLSTM model performance is better than LSTM model. Despite the GRU model performing well in the forecasting of the confirmed cases for short time series, it presented bad performance with forecasting of the confirmed cases for long time series. We noticed a similar performance between forecasting models for confirmed and death cases.

We observed the variation of the LSTM performance between forecasting of the short time-series and long time-series. ARIMA models are a powerful statistical forecasting tools. However, their (p,q,d) values control their performance. For time-series forecasting evaluation criteria, it is crucial to take into consideration the value of R-Squared with the achieved error values.

### 4.4. Experiment 4

In this experiment, we utilized our proposed model to perform COVID-19 forecasting in two categories of countries. Generally, the pandemic spread wave was followed by significant confirmed and death cases. This wave over-time is visualized as an exponential growth in the cumulative epidemic graph. In our study, we investigate how our model will perform with different spread forms. The first category A includes the counties with a single COVID-19 spread wave, which are Brazil, India, South Africa, and Saudi Arabia during the studying period, as shown in Table 5.
Figure 6. The forecasting results of our proposed method vs. the previous methods for confirmed, recovered, and death cases in Saudi Arabia. (a) Confirmed case forecasting results in the starting of the first COVID-19 wave in Saudi Arabia. (b) Confirmed case forecasting results in the ending of the first COVID-19 wave in Saudi Arabia. (c) Recovered case forecasting results in the starting of the first COVID-19 wave in Saudi Arabia. (d) Recovered case forecasting results in the ending of the first COVID-19 wave in Saudi Arabia. (e) Death case forecasting results in the starting of the first COVID-19 wave in Saudi Arabia. (f) Death case forecasting results in the ending of the first COVID-19 wave in Saudi Arabia.
The second category B includes the countries with double COVID-19 spread waves, which are Germany, Italy, Turkey, and Spain as shown in Table 2. In this experiment, we also utilize our proposed model to investigate both deaths and confirmed cases in two different periods, Period 1 starting from 22 JAN, 2020 to 31 May, 2020, and Period 2 starting from 22 JAN, 2020 to 27 Dec, 2020. Each period is split into 80% for training and 20% for testing.

As shown in Table 5, the forecasting results of Saudi Arabia’s death cases achieved the best performance. This country demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9989, 63.9, 6.55, and 8, respectively. Brazil achieved the second rank of the confirmed cases forecasting. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9942, 6.591 × 10³, 5.881 × 10³, and 4.341 × 10⁷, respectively. India achieved the lowest forecasting performance for confirmed cases during period 1. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.8662, 1.231 × 10⁴, 9.661 × 10³, and 1.501 × 10⁸, respectively.

As shown in Table 5, the forecasting results of Saudi Arabia’s death cases achieved the lowest MASE error value of 0.18 for the forecasting of death cases in period 2. Italy achieved the lowest MASE error value of 0.46 for the forecasting of death cases in period 2. South Africa achieved the highest MASE error value of 9.691 × 10⁴, 6.821 × 10⁴, and 9.601 × 10⁴, respectively. Spain achieved the lowest forecasting performance for confirmed cases during period 1. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9141, 9.201 × 10², 8.381 × 10², and 8.461 × 10², respectively.

As shown in Table 2, the forecasting results of Italy’s death cases achieved the best performance. During Period 2, it had R-Squared, RMSE, MAE, and MSE of 0.9987, 3.921 × 10², 3.021 × 10², and 1.541 × 10⁵, respectively. During Period 1, it had R-Squared, RMSE, MAE, and MSE of 0.9904, 60.39, 40.93, and 4.091 × 10³, respectively. Turkey achieved good forecasting results for death cases during period 2. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9884, 3.101 × 10², 2.271 × 10², and 9.601 × 10⁴, respectively. Spain achieved the lowest forecasting performance for confirmed cases during period 1. It demonstrated an R-Squared, RMSE, MAE, and MSE of 0.9141, 9.201 × 10², 8.381 × 10², and 8.461 × 10², respectively.

### Table 5. COVID-19 spread forecasting results in countries with single spread wave based statistical evaluation criteria.

| Country    | Cases       | Period          | MSE        | MAE         | RMSE        | MASE        | R_Squared |
|------------|-------------|-----------------|------------|-------------|-------------|-------------|-----------|
| Brazil     | Death Cases | Period 1        | 9.581 × 10⁵ | 9.691 × 10⁵ | 9.791 × 10² | 0.96        | 0.9581    |
|            |             | Period 2        | 1.271 × 10⁶ | 1.111 × 10³ | 1.131 × 10³ | 3.47        | 0.9863    |
|            | Confirmed Cases | Period 1      | 4.341 × 10⁷ | 5.881 × 10³ | 6.591 × 10³ | 0.71        | 0.9942    |
|            |             | Period 2        | 1.211 × 10¹⁰| 1.101 × 10⁵ | 1.101 × 10⁵ | 7.93        | 0.9677    |
| India      | Death Cases | Period 1        | 4.911 × 10³ | 6.281 × 10¹ | 7.011 × 10¹ | 3.8         | 0.9922    |
|            |             | Period 2        | 8.071 × 10⁵ | 8.671 × 10² | 8.981 × 10² | 2.63        | 0.9879    |
|            | Confirmed Cases | Period 1   | 1.501 × 10⁸ | 9.661 × 10³ | 1.231 × 10⁴ | 3.72        | 0.8662    |
|            |             | Period 2        | 1.841 × 10¹⁰| 1.141 × 10⁵ | 1.361 × 10⁵ | 3.59        | 0.9551    |
| South Africa | Death Cases | Period 1        | 1.211 × 10³ | 2.331 × 10¹ | 3.471 × 10¹ | 4.15        | 0.9334    |
|            |             | Period 2        | 4.851 × 10⁴ | 2.041 × 10² | 2.201 × 10² | 16.04       | 0.9878    |
|            | Confirmed Cases | Period 1   | 2.411 × 10⁶ | 1.501 × 10³ | 1.551 × 10³ | 2.31        | 0.9237    |
|            |             | Period 2        | 1.711 × 10⁸ | 1.201 × 10⁴ | 1.311 × 10⁴ | 5.56        | 0.9691    |
|            | Death Cases | Period 1        | 6.391 × 10¹ | 6.551 × 10⁰ | 8.001 × 10⁰ | 1.78        | 0.9989    |
|            |             | Period 2        | 1.561 × 10¹ | 3.401 × 10⁰ | 3.961 × 10⁰ | 0.46        | 0.9954    |
| Saudi Arabia | Confirmed Cases | Period 1   | 9.371 × 10⁵ | 8.031 × 10² | 9.681 × 10² | 2.24        | 0.9929    |
|            |             | Period 2        | 3.981 × 10⁴ | 1.711 × 10² | 2.001 × 10² | 0.18        | 0.9981    |

Saudi Arabia achieved the lowest MASE error value of 0.18 for the forecasting of confirmed cases in period 2. South Africa achieved the highest MASE error value of 16.04 for the forecasting of death cases in period 2. Italy achieved the lowest MASE error value of 0.2 for the forecasting of death cases in period 1. Germany achieved the highest MASE error value of 83.27 for the forecasting of death cases in period 2.

We visualize the forecasting graphs for the second countries category, which are Turkey, Italy, Germany, and Spain. All forecasting results are shown in Figure 7a-d. Our proposed model achieved high performance to forecast the confirmed cases for along 14 days during the first spread wave for all four countries. On the other hand, it also achieved a high performance to forecast the Germany, Italy, and Spain confirmed cases during for 60 days during the second spread wave. However, the Turkey forecasting results achieved a low performance during the last 15 days.
From Experiment 4, we have proven the generalization capabilities of our proposed model and its superior performance. Our proposed model was able to forecast both deaths and confirmed cases for eight different countries, according to these counties’ variation of control policies. Some of them witnessed two spread waves during the same year, and some have witnessed a single spread wave. Our proposed model can forecast both confirmed and death cases for the two kinds of spread models. In addition, we noticed the superior performance of our proposed model to forecast the confirmed cases for 60 days during the first and second waves. We conclude that the countries from category A achieved lower MASE values than those from category B.

![Graphs showing COVID-19 confirmed cases spread forecasting in several countries.](image)

**Figure 7.** COVID-19 confirmed cases spread forecasting in several countries. (a) Turkey. (b) Italy. (c) Germany. (d) Spain.

5. Conclusions

In this paper, an Encoder–Decoder BiLSTM deep learning model was utilized to forecast the COVID-19 confirmed cases, recovered cases, and death cases. First, we optimized the hyper-parameters for our proposed model based on Saudi Arabia’s reported cases. Primarily, our proposed model optimal values of the initial learning rate, hidden units, and dropout percentage were 0.0005, 128, and 0.2, respectively. We also investigated the effect of the features scaling technique, and we proved that the standardization technique was more suitable with COVID-19 time series forecasting performance.

We also investigated the COVID-19 spread through the year for two different periods. The forecasting evaluation was assessed based on 14 days and 60 days. We compared our proposed model and the fifteen previous methods in the literature, and our proposed model was the most stable. The Encoder–Decoder BiLSTM achieved higher performance than the traditional Encoder–Decoder BiLSTM model. Moreover, our proposed model achieved higher performance than traditional machine learning techniques, ARIMA models, and deep learning models applied to the same countries.
Our proposed model was applied to forecast several countries that witnessed a single-wave spread, Saudi Arabia, India, Brazil, South Africa, as well as double-wave spreads, Italy, Spain, Germany, and Turkey. Our proposed model achieved lower RMSE, MSE, and MAE values. Furthermore, it achieved the highest R-Squared values. For future work, it is recommended to study the capability of our proposed model to simulate the COVID-19 pandemic spread with different policies and lockdown regulations. Furthermore, there is a real need to investigate the recent hyper-parameter optimization techniques to speed up hyper-parameter selection. Our proposed model will be applied to other pandemic forecasting problems.

**Supplementary Materials:** The following are available online at https://www.mdpi.com/article/10.3390/fractalfract5040175/s1, Table S1: Statistical evaluation criteria of our proposed approach with network’s parameters tuning for confirmed cases, Table S2: Statistical evaluation criteria of our proposed approach with network’s parameters tuning for recovered cases, Table S3: Statistical evaluation criteria of our proposed approach with network’s parameters tuning for death cases, Table S4: Statistical evaluation criteria of our proposed approach with different training optimization algorithms for confirmed cases in Saudi Arabia with the median hyper-parameters, Table S5: Statistical evaluation criteria of our proposed approach with different training optimization algorithms for confirmed cases in Saudi Arabia with the optimal hyper-parameters.

**Author Contributions:** Conceptualization, A.I.S. and S.A.; methodology, A.I.S.; validation. A.I.S.; formal analysis, A.I.S.; investigation, S.A.; resources, S.A.; data curation, S.A.; writing—original draft preparation, A.I.S.; writing—review and editing, A.I.S., S.A.; visualization, A.I.S.; supervision, S.A.; project administration, S.A.; funding acquisition, S.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors extend their appreciation to the deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IFP-2020-17).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset is public and the code is available on request from A.I.S.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Arden, M.A.; Chilcot, J. Health psychology and the coronavirus (COVID-19) global pandemic: A call for research. *Br. J. Health Psychol.* 2020, 25, 231–232. [CrossRef]
2. Manderson, L.; Levine, S. COVID-19, risk, fear, and fall-out. *Med. Anthropol.* 2020, 39, 367–370. [CrossRef]
3. Sornette, D.; Mearns, E.; Schatz, M.; Wu, K.; Darcey, D. Interpreting, analysing and modelling COVID-19 mortality data. *Nonlinear Dyn.* 2020, 101, 1751–1776. [CrossRef]
4. Bansal, A.; Padappayil, R.P.; Garg, C.; Singal, A.; Gupta, M.; Klein, A. Utility of artificial intelligence amidst the COVID 19 pandemic: A review. *J. Med. Syst.* 2020, 44, 1–6. [CrossRef]
5. Chen, J.; See, K.C. Artificial Intelligence for COVID-19: Rapid Review. *J. Med. Internet Res.* 2020, 22, e21476. [CrossRef] [PubMed]
6. Jamshidi, M.; Lalbakhsh, A.; Talla, J.; Peroutka, Z.; Hadjilooei, F.; Lalbakhsh, P.; Jamshidi, M.; La Spada, L.; Mirmozafari, M.; Dehghani, M.; et al. Artificial intelligence and COVID-19: Deep learning approaches for diagnosis and treatment. *IEEE Access* 2020, 8, 109581–109595. [CrossRef] [PubMed]
7. Pham, Q.V.; Nguyen, D.C.; Hwang, W.J.; Pathirana, P.N. Artificial intelligence (AI) and big data for coronavirus (COVID-19) pandemic: A survey on the state-of-the-arts. *IEEE Access* 2020, 8, 130820–130839. [CrossRef]
8. Desai, S.B.; Pareek, A.; Lungren, M.P. Deep learning and its role in COVID-19 medical imaging. *Intell.-Based Med.* 2020, 3, 100013. [CrossRef]
9. Jakhar, D.; Kaur, I. Current applications of artificial intelligence for COVID-19. *Dermatol. Ther.* 2020. [CrossRef]
10. Lalmuanawma, S.; Hussain, J.; Chhakchhuak, L. Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons Fractals* 2020, 139, 110059. [CrossRef]
11. Tayarani-N, M.H. Applications of artificial intelligence in battling against Covid-19: A literature review. *Chaos, Solitons Fractals* 2020, 142, 110338. [CrossRef]
12. Vaishya, R.; Javaid, M.; Khan, I.H.; Haleem, A. Artificial Intelligence (AI) applications for COVID-19 pandemic. *Diabetes Metab. Syndr. Clin. Res. Rev.* 2020, 14, 337–339. [CrossRef]

13. Srinivasa Rao, A.S.; Vazquez, J.A. Identification of COVID-19 can be quicker through artificial intelligence framework using a mobile phone-based survey when cities and towns are under quarantine. *Infect. Control. Hosp. Epidemiol.* 2020, 41, 826–830. [CrossRef]

14. Mbunge, E. Integrating emerging technologies into COVID-19 contact tracing: Opportunities, challenges and pitfalls. *Diabetes Metab. Syndr. Clin. Res. Rev.* 2020, 14, 1631–1636. [CrossRef]

15. Shi, F.; Wang, J.; Shi, J.; Wu, Z.; Wang, Q.; Tang, Z.; He, K.; Shi, Y.; Shen, D. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE Rev. Biomed. Eng.* 2020, 14, 4–15. [CrossRef]

16. Ke, Y.Y.; Peng, T.T.; Yeh, T.K.; Huang, W.Z.; Chang, S.E.; Wu, S.H.; Hung, H.C.; Hsu, T.A.; Lee, S.J.; Song, J.S.; others. Artificial intelligence approach fighting COVID-19 with repurposing drugs. *Biomed. J.* 2020, 43, 355–362. [CrossRef] [PubMed]

17. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* 2021, 167, 108288. [CrossRef] [PubMed]

18. Rezaei, M.; Azarmi, M. Deepsocial: Social distancing monitoring and infection risk assessment in covid-19 pandemic. *Appl. Sci.* 2020, 10, 7514. [CrossRef]

19. Car, Z.; Baresti Šegota, S.; Andelić, N.; Lorenčin, I.; Mrzljak, V. Modeling the spread of COVID-19 infection using a multilayer perceptron. *Comput. Math. Methods Med.* 2020, 2020, 5714714. [CrossRef] [PubMed]

20. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, 521, 436–444. [CrossRef] [PubMed]

21. Shahin, A.I.; Almotairi, S. An accurate and fast cardio-views classification system based on fused deep features and LSTM. *IEEE Access* 2020, 8, 135184–135194. [CrossRef]

22. Aslan, M.F.; Unlersen, M.F.; Sabanci, K.; Durdu, A. CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection. *Appl. Soft Comput.* 2021, 98, 106912. [CrossRef] [PubMed]

23. Kumar, A.; Gupta, P.K.; Srivastava, A. A review of modern technologies for tackling COVID-19 pandemic. *Diabetes Metab. Syndr. Clin. Res. Rev.* 2020, 14, 569–573. [CrossRef] [PubMed]

24. Ceylan, Z. Estimation of COVID-19 prevalence in Italy, Spain, and France. *Sci. Total. Environ.* 2020, 729, 138817. [CrossRef]

25. Fanelli, D.; Piazza, F. Analysis and forecast of COVID-19 spreading in China, Italy and France. *Chaos, Solitons Fractals* 2020, 134, 109761. [CrossRef]

26. Gupta, R.; Pal, S.K. Trend Analysis and Forecasting of COVID-19 outbreak in India. *MedRxiv* 2020.

27. Abuhasel, K.A.; Khadr, M.; Alquraish, M.M. Analyzing and forecasting COVID-19 pandemic in the Kingdom of Saudi Arabia using ARIMA and SIR models. *Comput. Intell.* 2020, 2020, 1–14. [CrossRef]

28. Ribeiro, M.H.D.M.; da Silva, R.G.; Mariani, V.C.; dos Santos Coelho, L. Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil. *Chaos Solitons Fractals* 2020, 135, 109853. [CrossRef]

29. Atangana, A.; Araz, S.I. Mathematical model of COVID-19 spread in Turkey and South Africa: Theory, methods, and applications. *Adv. Differ. Eq.* 2020, 2020, 1–89. [CrossRef]

30. Zeroual, A.; Harrou, F.; Dairi, A.; Sun, Y. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos Solitons Fractals* 2020, 140, 110227. [CrossRef]

31. Shastry, S.; Singh, K.; Kumar, S.; Kour, P.; Mansotra, V. Time series forecasting of COVID-19 using deep learning models: India-USA comparative case study. *Chaos Solitons Fractals* 2020, 140, 110227. [CrossRef]

32. Zouroll, A.; Harrou, F.; Dairi, A.; Sun, Y. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos Solitons Fractals* 2020, 140, 110121. [CrossRef] [PubMed]

33. Shahid, F.; Zameer, A.; Muneeb, M. Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos Solitons Fractals* 2020, 140, 110212. [CrossRef] [PubMed]

34. Karaçuha, E.; Oral, N.O.; Ergün, E.; Tabatabadez, V.; Alkaş, H.; Karaçuha, K.; Tontuş, H.O.; Nu, N.V.N. Modeling and Prediction of the Covid-19 Cases With Deep Assessment Methodology and Fractional Calculus. *IEEE Access* 2020, 8, 164012–164034. [CrossRef]

35. Rustam, F.; Reshi, A.A.; Mehmood, A.; Ullah, S.; On, B.W.; Aslam, W.; Choi, G.S. COVID-19 future forecasting using supervised machine learning models. *IEEE Access* 2020, 8, 101489–101499. [CrossRef]

36. Ibrahim, M.A.; Al-Najafi, A. Modeling, Control, and Prediction of the Spread of COVID-19 Using Compartmental, Logistic, and Gauss Models: A Case Study in Iraq and Egypt. *Processes* 2020, 8, 1400. [CrossRef]

37. Amar, L.A.; Taha, A.A.; Mohamed, M.Y. Prediction of the final size for COVID-19 epidemic using machine learning: A case study of Egypt. *Infect. Dis. Model.* 2020, 5, 622–634. [CrossRef] [PubMed]

38. Prasanth, S.; Singh, U.; Kumar, A.; Tikkival, V.A.; Chong, P.H. Forecasting spread of COVID-19 using Google Trends: A hybrid GWO-Deep learning approach. *Chaos Solitons Fractals* 2021, 142, 110336. [CrossRef] [PubMed]

39. Petropoulos, F.; Makridakis, S.; Stylianou, N. COVID-19: Forecasting confirmed cases and deaths with a simple time series model. *Int. J. Forecast.* 2020. [CrossRef] [PubMed]

40. Abbasimehr, H.; Paki, R. Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization. *Chaos Solitons Fractals* 2021, 142, 110511. [CrossRef] [PubMed]

41. Middle, I. Regional variations in the prevalence of consanguinity in Saudi Arabia. *Saudi. Med. J.* 2007, 28, 1881–1884.

42. Alzahrani, S.I.; Aljamaan, I.A.; Al-Fakih, E.A. Forecasting the spread of the COVID-19 pandemic in Saudi Arabia using ARIMA prediction model under current public health interventions. *J. Infect. Public Health* 2020, 13, 914–919. [CrossRef] [PubMed]
71. Khayyat, M.; Laabidi, K.; Almalki, N.; Al-zahrani, M. Time Series Facebook Prophet Model and Python for COVID-19 Outbreak Prediction. *CMC-Comput. Mater. Contin.* 2021, 67, 3781–3793. [CrossRef]

72. Devaraj, J.; Elavarasan, R.M.; Pugazhendhi, R.; Shafiullah, G.; Ganesan, S.; Jeysree, A.K.; Khan, I.A.; Hossain, E. Forecasting of COVID-19 cases using deep learning models: Is it reliable and practically significant? *Results Phys.* 2021, 21, 103817. [CrossRef]