Duzce University Journal of Science and Technology
Düzce Üniversitesi Bilim ve Teknoloji Dergisi
14 (1), 2026, 20-37.
https://doi.org/10.29130/dubited.1662505
e-ISSN: 2148-2446
https://dergipark.org.tr/en/pub/dubited

Research Article

COVID-19 Pandemic Prediction Using Hyper-Parameter-Tuned ANN, Bi-LSTM, and Stacked-LSTM in Türkiye

Serpil Özer[1*], Mustafa Göçken[2], Ayşe Tuğba Dosdoğru[2]

[1] OSTİM Technical University, Department of Industrial Engineering, Ankara, Türkiye. ror.org/049xhb141

[2] Adana Alparslan Türkeş Science and Technology University, Department of Industrial Engineering, Adana, Türkiye. ror.org/013z3yn41

*Corresponding author: serpil.ozer@ostimteknik.edu.tr

Received: 21.03.2025
Revision: 01.06.2025
Accepted: 09.07.2025
Available Online: 21.01.2026

Abstract: COVID-19 first appeared in China in December 2019. In Türkiye, the first case was seen on March 11, 2020. Since the emergence of COVID-19, it has rapidly affected the world and caused the death of many people. The rapid spread of the virus around the world and the fact that it poses a global threat have made it mandatory for the authorities to take quick measures and decisions. For this reason, the diagnosis, treatment, and prediction of the number of cases of COVID-19 disease is a vital issue. In this paper, daily deaths, daily cases, cumulative deaths, and cumulative case numbers were predicted using Türkiye's COVID-19 data. The prediction analysis for COVID-19 was conducted employing artificial intelligence (AI) techniques, specifically Stacked Long-Short Term Memory (Stacked-LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), and Artificial Neural Network (ANN) methods. Hyperparameter optimization using the Gray Wolf Optimizer Algorithm (GWO) was applied to optimize these models. To assess the accuracy and efficacy of prediction models, various performance metrics were employed, including Mean Absolute Percentage Error (MAPE), Coefficient of Determination (R^2 Score), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), and Explained Variance Score (EVS). The assessment of the accuracy and efficacy of the pandemic prediction models involved comparing the predicted data values with the actual values. Based on this analysis, the prediction model that exhibited the highest degree of similarity to the actual values was determined to be the most reliable and effective.

Keywords: *Data analysis, Deep learning, Artificial neural networks, Long short-term memory (LSTM), COVID-19, GWO*

I. INTRODUCTION

Throughout history, humanity has faced many pandemics that have led to serious health impacts and deaths. In December 2019, an increase in pneumonia cases in Wuhan, China, led to the discovery of a new coronavirus (2019-nCoV). This virus, dubbed SARS-CoV-2, quickly spread to 113 countries, and a pandemic was declared on March 11 (Tandon et. al., 2022). The first case in Türkiye was diagnosed in March 2020 (T.C. Sağlık Bakanlığı, 2021). The virus shows cold-like symptoms, and the risk is higher for people with cancer, diabetes, and heart disease. Quarantine and social isolation are in place to reduce the spread of the virus. The status of the pandemic is monitored by parameters such as the number of reproduction and incubation periods; however, since these parameters alone are not sufficient to explain the general situation of the pandemic, comprehensive research and analysis are needed (Scientific Committee Study COVID-19 (SARS-CoV-2) Infection Guide, 2020).

After examining the current literature on COVID-19, it becomes apparent that researchers extensively utilize statistical analysis and mathematical models in their investigations. Nevertheless, traditional analytical methodologies may prove inadequate, as real-time studies on the trajectory of the pandemic demand expeditious insights (World Health Organization Regional Office for Europe, 2021). Consequently, researchers have increasingly gravitated towards the adoption of Artificial Intelligence (AI) models, which have attracted considerable attention in recent years within the realm of COVID-19 research. Notably recognized for their capacity to yield practical and consequential outcomes in addressing everyday challenges, AI models have emerged as a frequently preferred approach among researchers. These methodologies offer compelling solutions to complex and challenging problems.

Concurrently, there exists a strong relationship between AI and the processes of acquiring knowledge from data, assessing the acquired information, and making informed decisions based on such knowledge. Within these techniques, classification, prediction, clustering, generalization, and learning functions are frequently favored approaches, enabling researchers to effectively address a diverse range of challenges. Deep learning approaches, notable for their flexibility, realism, ease of application, cost-effectiveness, and expeditiousness, have emerged as powerful tools in achieving rapid and practical outcomes. Hence, in this study, the objective is to employ AI techniques for prediction purposes and assess the performance of these models.

Cite as: Özer, S., Gökçen, M., & Dosdoğru, A. T. (2026). COVID-19 pandemic prediction using hyper-parameter-tuned ANN, Bi-LSTM, and Stacked-LSTM in Türkiye. *Duzce University Journal of Science and Technology, 14*(1), 20-37. https://www.doi.org/10.29130/dubited.1662505

This is an open access paper distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.

Throughout history, pandemics have had a profound impact on human life. Obtaining comprehensive pandemic data is critical for mitigating the effects of a pandemic, understanding its progression, and developing informed decisions and measures accordingly. Therefore, extensive research has been conducted on the effects and progression of the pandemic, and various analysis methods have been applied by utilizing pandemic data. For example, Khare et al. showed the global spread rate of COVID-19 with visual exploratory data analysis, aiming to raise awareness and examine the increasing number of infections. They used the SEIR (susceptible-exposed-contagious-infected-recovered) model to predict future cases and deaths and evaluated the performance of the model with sensitivity analysis (Khare et al., 2020). In a study conducted in Pakistan the SIR (susceptible-infected-recovered) model was applied for accurate prediction of case numbers (Waqas et al. 2020). Crokidakis used the SIQR (susceptible-infected-quarantined-recovered) model to predict COVID-19 cases in Brazil, taking into account variables such as the number of exposed, recovered, infected, and quarantined individuals (Crokidakis, 2020).

Tandon et al. examined the effects of the 60-day lockdown in Italy on the pandemic through parameter estimation using statistical analysis methods such as ARIMA (autoregressive integrated moving average). The study confirms that the lockdown and isolation measures positively affected the spread of the pandemic (Tandon et. al., 2022). AI and machine learning algorithms have become prominent approaches in decision-making processes related to the COVID-19 virus. These techniques have been widely adopted in many pandemic-related issues, such as early detection of infections, prediction of case and mortality rates, drug and vaccine development, and virus control. Vaishya emphasized the importance of AI and machine learning methods in these areas, highlighting their potential to provide valuable insights and solutions in the fight against COVID-19 (Vaishya et al., 2020).

Irmak et al. used three different models to predict the patient density in a hospital for the next month based on the number of patients admitted to the hospital. First, a time series analysis was performed using an exponential smoothing model, followed by an analysis using an ARIMA model, and finally a forecasting model was developed using an ANN (artificial neural network). Each analysis was performed independently by building different models, and a comparative evaluation was conducted to determine the most effective approach (Irmak et al., 2012). Similarly, Picariello and Aliani estimated total cases and mortality rates for the Lombardy region and the provinces of Bergamo and Brescia (Picariello & Aliani, 2020). Lalmuanawma et al. evaluated the accuracy and reliability of AI and deep learning approaches in epidemic forecasting and showed that AI and deep learning models can be used to accurately predict the spread of epidemics and reveal previously unknown hidden patterns in the data (Lalmuanawma et al., 2020).

Wang et al. compared traditional epidemic models with the LSTM model using data from Iran, Peru, and Russia and evaluated their performance in predicting trends in COVID-19 case numbers. The results showed that traditional epidemic analyses can predict the overall trend in case numbers with sufficient accuracy, but the LSTM technique more successfully predicts critical turning points where the spread of the virus will decrease or stop (Wang et al., 2020). Similarly, Bahri et al. predicted the number of recovered cases with the LSTM model, focusing on three selected cities from different continents. Based on data from the US, Italy, and India, the study aimed to predict the total number of recovered cases in 7 days and evaluated the accuracy of the model with performance measures such as MAE, MSE, MAPE, and RMSE (Bahri et al., 2020). Shahid et al. predicted the number of confirmed cases, recovered cases, and deaths using various models such as ARIMA, support vector regression (SVR), LSTM, and Bi-LSTM. The performance of the models was evaluated by measures such as MAE, RMSE, and R^2 score, and the best model was identified (Shahid et al., 2020).

Ayoobi et al. used various deep learning models such as LSTM, Bi-LSTM, Convolutional LSTM (Conv-LSTM), Bidirectional Convolutional LSTM (Bi-Conv-LSTM), GRU, and Bidirectional GRU (Bi-GRU) to predict COVID-19 cases in Austria and Iran. The researchers analyzed the predictive performance of the Bi-GRU method by predicting the number of cases 1, 3, and 7 days ahead in a 100-day period (Ayoobi et al., 2021). Istaiteh et al. used various models such as ARIMA, LSTM, ANN, and Convolutional Neural Networks (CNN) to predict the one-week course of the COVID-19 pandemic. They also tried to optimize the performance by investigating the effective use of hyperparameters in these models and evaluated the compatibility and accuracy of the models in predicting the outbreak (Istaiteh et al., 2020).

In this paper, COVID-19 data specific to Türkiye were sourced from the Turkovac website, comprising distinct datasets related to daily cases, daily deaths, cumulative cases, and cumulative deaths. The study aimed to predict these data groups using LSTM, Bi-LSTM, Stacked-LSTM, and ANNs methods, subsequently evaluating the performance of the employed models. While neural networks have shown proficiency in solving complex problems, they often present challenges during the training phase. However, studies indicate that incorporating optimization algorithms when training AI models yields more favorable outcomes (Zeroual et. al., 2020). The effectiveness of optimization algorithms relies on the control parameters employed, with each algorithm possessing unique characteristics in this regard. Notably, the Gray Wolf Optimizer Algorithm (GWO) algorithm minimizes the dependence on control parameters and is trained using population size and number of iterations as the key control parameters (Er & Işık, 2021). The Bi-LSTM, Stacked-LSTM, and ANN machine learning algorithms used in this study were optimized using GWO, and their performances were compared to determine the most accurate prediction model.

This study is aimed to make a significant contribution to the literature on the prediction of COVID-19 case and death numbers. In addition to widely used AI methods in the literature, the study conducts a comparative analysis of deep learning-based models such as Stacked-LSTM, Bi-LSTM, and ANN. Furthermore, the use of the GWO algorithm for hyperparameter optimization represents an innovative approach aimed at improving prediction accuracy. The integration of such optimization techniques with COVID-19 prediction models remains relatively limited in the existing literature. The novelty of this study lies in both the diversity of models employed and the systematic evaluation of hyperparameter optimization through various performance metrics.

In the first section of the study, titled Introduction, the general framework of the research is outlined and a comprehensive review of the relevant literature is presented. The second section provides detailed information on the methods and techniques employed in the study. The third section presents the research findings and the results obtained. Finally, in the Conclusion section, an overall evaluation of the study is provided, and interpretations are made in light of the findings.

II. MATERIAL AND METHOD

In this study, various AI techniques and time-dependent neural network architectures were utilized to predict daily and cumulative COVID-19 case and death numbers. Fundamental concepts such as data analysis, artificial intelligence, and supervised and unsupervised learning are explained in the following sections to establish the theoretical and practical foundation of the study. Explaining these concepts helps clarify the technical approaches and decision-making processes on which the applied methods are based.

AI and its subfields—such as ANNs, deep learning, and optimization algorithms—are particularly noteworthy for their ability to model complex relationships in datasets characterized by high uncertainty, nonlinearity, and temporal variability, as seen during the pandemic. The selection of these methods was based not only on their proven success in the literature but also on their adaptability to multidimensional and dynamic datasets like those related to COVID-19.

ANNs were employed in this study due to their capacity to work with multidimensional medical data and detect nonlinear relationships. LSTM, Stacked-LSTM, and Bi-LSTM models were selected for their superior performance in modeling dependencies in time series data. To enhance the training performance of ANNs, the GWO algorithm was used as a metaheuristic optimization technique. GWO effectively optimizes the model's weight parameters and minimizes error functions, thereby positively influencing model performance.

Traditional training algorithms, especially when applied to nonlinear and complex datasets such as those related to pandemics, are prone to getting stuck in local minima and often fail to generalize effectively. Inspired by the leadership hierarchy and hunting strategies of grey wolves, the GWO algorithm offers a powerful global search capability, leading to faster and more stable convergence. As a result, the integration of GWO significantly improved the ANN training process, yielding higher prediction accuracy, better generalization capability, and more stable model performance. GWO has proven particularly effective in forecasting multidimensional datasets such as daily and cumulative COVID-19 cases and deaths.

A. Data Analysis

Data refers to any object, such as numbers, words, photographs, music, etc., that can be stored in a computer (Uğur & Kınacı, 2006). Today, a large amount of data is collected with technological developments. Data analysis aims to produce useful information and results by processing and interpreting raw data, especially using statistical techniques. In this process, the purpose of the analysis, the method, and the size of the data are extremely important. Data is used by experts and organizations to make strategic decisions, improve product quality, and provide better services. In the field of medicine, data analysis is applied for many purposes, from disease diagnosis and treatment to predicting the number of patients, from evaluating intensive care occupancy rates to predicting hospital admission rates.

When the data is properly analyzed, it produces effective results, and their performance is improved. Within data analysis, various types of analysis are commonly employed, including Descriptive Analysis, Exploratory Analysis, Inferential Analysis, Forecasting Analysis, and Prescriptive Analysis, each serving specific purposes. AI techniques are commonly favored for data analysis, classification tasks, establishing data relationships, and making accurate predictions. In the realm of healthcare, AI techniques including expert systems, machine learning, deep learning, ANNs, genetic algorithms, and fuzzy logic find extensive application in various domains such as disease detection, disease treatment, image recognition, and drug development. AI encompasses the domains of machine learning and deep learning, as depicted in Figure 1. Subsequent sections provide detailed descriptions of these domains.

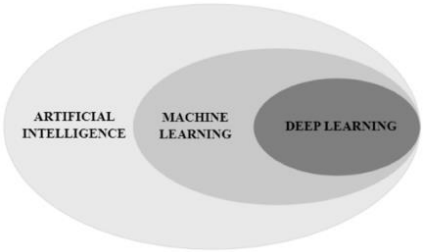


Figure 1. AI and its sub-branches

B. Artificial Intelligence

AI is a field of science and engineering that enables computer systems to perform human-like behaviors such as reasoning, understanding, comprehension, and learning from experience. Studies in this field aim to make computers approach human-like intelligence by utilizing complex algorithms, statistical models, and data analysis methods (Sazlı, 2006). In other words, it is a program that can show people's behavior in an intelligent way. AI is a computer science that works integrated with other fields of science to examine how the human brain works and how experiences and information are obtained (Er & Bal, 2020). AI demonstrates remarkable proficiency in solving intricate problems with speed and efficiency, enabling rapid acquisition of essential data. AI's exceptional problem-solving capabilities extend beyond complex tasks to encompass daily life problems. This remarkable ability has led to widespread utilization of AI across various domains, including modern medicine, biomedicine, energy, the economy, production, banking, and the service sectors (Uğur & Kınacı, 2006).

B.1. ANN

The continuous advancement of technology has led to an increased integration of machines in various aspects of our lives. This increased dependence on machines has necessitated the development of new methods and processes to effectively address emerging challenges. In this context, ANN algorithms and models have emerged as prominent tools for tackling multidimensional and complex problems. It is extensively applied across various domains of medicine, such as diagnosis of disease and development of treatments, radiology, pathology, CT (computerized tomography), and MR (magnetic resonance imaging), and contributes to solving medical problems (Sazlı, 2006). Therefore, the use of ANNs is increasing rapidly as reliable and fast information can be obtained, effective relationships can be established, and nonlinear complex behaviors can be detected (Sharma et al., 2017). The working principle of ANNs was established by observing the nerve cells called neurons in the human brain. ANNs facilitate the transmission of information through interconnected neurons, enabling the exchange and processing of information among them (Sharma et al., 2017). The impact of interconnections between elements in the neural network model is determined by assigning weights to each of them (Uğur & Kınacı, 2006).

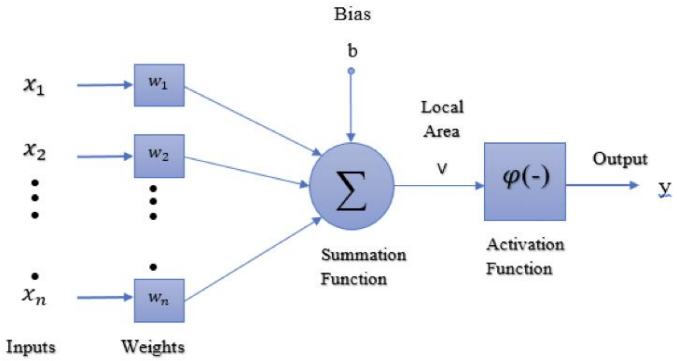


Figure 2. Neural Network Structure

Figure 2 shows the structure of a neuron in an ANN. As seen in Figure 2, the structure of the neural network consists of inputs, weights, outputs, addition functions, and activation functions. Input is the information or data that the neural network receives from outside. A neural network can have an unlimited amount of data. The weights indicate the importance of the incoming input data and the effect it has on the neuron. Input data is connected to neurons according to the effect of weights. Whether the weights are large or small does not indicate that the information is important or unimportant. This situation may vary according to the installed model. It is known that the initial value of the weights is random and is usually between -1 and +1 (Sharma et al., 2017). The aggregation function calculates the net value of the input data. The net input value is found by multiplying the input value by the weights (Yönem & Akay, 2020). The activation function determines the output data by processing the input data coming to the neuron. It is important to

choose the right activation function, as it significantly affects performance. The selection of the activation function is based on the specific learning objectives to be achieved from the input data. Accordingly, the commonly used activation functions are linear activation functions, step functions, tangent hyperbolic functions, threshold functions, sinus activation functions, sigmoid activation functions, softsingn activation functions, and softplus activation functions (Er & Bal, 2020). The output refers to the information obtained after processing the input data. When classifying ANNs based on the learning approach, they are categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning: Involves utilizing real-time historical data, where the expected output value corresponding to the input variable is known (Tüzüntürk, 2010). Initially, network weights are assigned randomly, and the actual output within the network is compared against the desired output value to assess the accuracy of the predicted output (Anderson & McNeill, 1992). This learning method is commonly employed in feed-forward networks.

Unsupervised Learning: Unsupervised learning is characterized by having only knowledge of the input variables entering the model, without information about the corresponding output data. In this learning approach, the neural network autonomously generates an output based on the input data without external guidance. The objective of unsupervised learning is to uncover patterns or structures within the dataset using the neural network model (Tüzüntürk, 2010). The weights of the network are adjusted based on its performance, without the need for external influences. While less commonly used compared to supervised learning, unsupervised learning finds its application in self-organizing neural networks (Tüzüntürk, 2010).

Reinforcement Learning: Involves a distinct learning approach where historical data is not utilized. Instead, the analysis of data is conducted through a process of trial and error. The system obtains outcomes based on real-world data and evaluates them against expected values (Marzouk et al., 2021). The primary objective of reinforcement learning is to identify the most rewarding or optimal actions by making decisions based on available information and effectively matching input and output data (Tapiwa et al., 2020). When classifying ANNs based on their structures, they can be categorized into two main types: feed-forward neural networks and feedback neural networks.

Feed-forward Neural Network: In this type of network, there is a forward link from one layer to the next, but no connection is observed between neurons in the same layer. In such neural networks, the information in the input layer is transmitted to the next layer without any calculation (Sharma et al., 2017). Figure 3 shows the feed-forward neural network structure. The number of neurons in the input and output layers depends on the specific problem and the nature of the data being analyzed. The determination of the number of hidden layers and the number of neurons within each layer in an ANN can be approached in two ways. Firstly, it can be determined based on the specific problem and domain knowledge, taking into consideration the complexity and characteristics of the data. Alternatively, an iterative process of trial and error can be employed to find the optimal configuration (Ataseven, 2013).

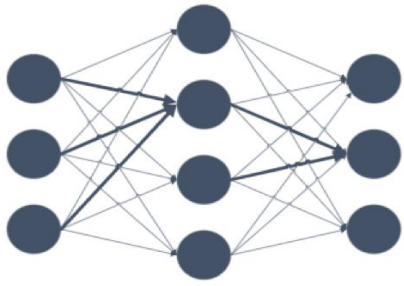


Figure 3. Feed-forward Neural Network Structure

Feedback neural networks: There is a backward link from the output layer to the input layer (Sengupta, 2020). In contrast to the feed-forward network structure, where input data only flows forward to the next layer, the structure of feedback neural networks allows for bidirectional information flow. In these networks, the input data not only transmits information to the subsequent layers but can also transmit information back to the previous layers.

In this type of network structure, known as the feedback neural network, the learning process involves comparing each input with the target output data. By calculating the error between the predicted output and the desired output, the error is propagated backward through the neural network model. This process is often referred to as backpropagation. Figure 4 illustrates the structure of the feedback neural network.

It is worth noting that learning in feedback neural networks tends to be slower compared to other network types.

These networks typically require a large amount of data for effective learning to occur. This is due to the iterative nature of backpropagation, where the network adjusts its weights and biases gradually to minimize the overall error. With more data, the network can fine-tune its parameters and improve its performance over time (Lalmuanawma et al., 2020).

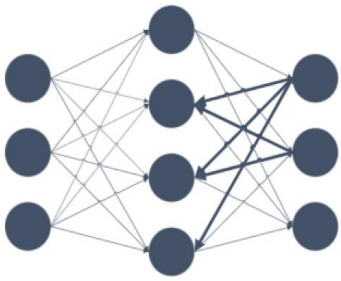


Figure 4. Back-Forward Neural Network Structure

When categorizing neural networks based on their intended use, they can be classified into two main categories: single-layer neural networks and multilayer neural networks.

Single-layer neural networks consist of only input and output layers. ANNs consisting of a single layer are used to estimate linear functions (Ataseven, 2013). Multilayer neural networks consist of at least one hidden layer. These multilayer neural network models are frequently preferred for solving complex problems. Additionally, many learning networks can be set up as multilayer neural networks (Ataseven, 2013).

LSTM was designed to manage long-term dependencies and memory, distinguishing it from traditional ANNs. As a specialized type of ANN, LSTM has been successful in tasks such as natural language processing and time series analysis. It can capture relationships over time more effectively and process longer-term dependencies. In this paper, in addition to ANN, LSTM techniques were also employed for the prediction of daily cases, daily deaths, cumulative cases, and cumulative deaths. The next section discusses LSTM techniques.

B.2. LSTM

LSTM was devised to address the temporal gap limitation in the RNN architecture, which is commonly employed for predicting subsequent states based on sequential states (Tapiwa et al., 2020). As RNN's capacity for retaining information over extended periods is inadequate, the LSTM technique was introduced to overcome this limitation. Consequently, unlike RNN, LSTM incorporates a memory layer. LSTM allows for the retention of long-term information while also facilitating the selective retention and discarding of specific elements within this information (Tapiwa et al., 2020). This characteristic of LSTM contributes to its effectiveness in various applications, including natural language processing, speech recognition, healthcare, and economics, where tasks such as analysis and forecasting are involved.

Since the desired outcome in RNN is depended on the previous result rather than the current input, the errors are calculated by backpropagation. As a result of this backpropagation, the vanishing gradient problem limits the models, so the training capacity of the system is lacking. To overcome this limitation, the LSTM architecture introduces a forget gate, which enables the handling of more complex structures (Marzouk et al., 2021). LSTM offers the advantage of processing input data as a series rather than individual data points. By utilizing memory cells to encode information and assess long-term relationships, LSTM emerges as a powerful tool, particularly in prediction studies (Bahri et al., 2020).

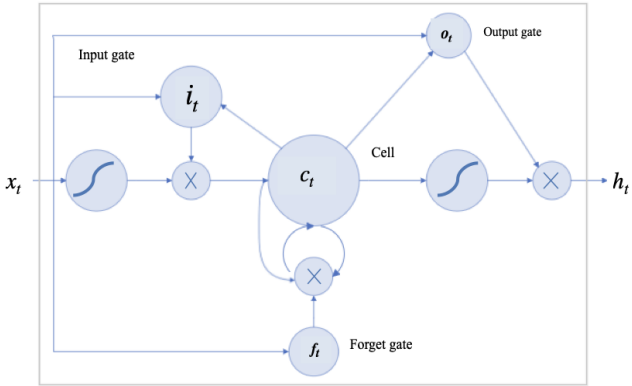


Figure 5. LSTM Model Structure

Figure 5 shows the structure of the LSTM model. LSTM cell architecture consists of memory cell, input gate, forget gate and output gate (Devaraj et al., 2021). Input, forget and output gates control the flow of information through cells in the neural network.

The input gate is responsible for selecting and incorporating relevant information into the cell. It is denoted as i_t and calculated according to equation 1. Here, x_t represents the input values at time t, and h_{t-1} refers to the cell output at time t-1. The Sigmoid function, operating within the range of 0 to 1, determines whether the information should be updated. On the other hand, the Tanh function assesses the significance of the information between -1 and 1 and assigns weights to the corresponding values.

The forget gate, denoted as f_t, determines which information from the previous cell should be forgotten and which should be retained. Its formula is provided in equation 2. By storing information in the memory cell, it can be preserved and passed on unchanged to the subsequent cell (Er & Işık, 2021). The LSTM mechanism ensures that relevant information is retained while unnecessary information is discarded from the information received by the neural network (Tapiwa et al., 2020). Additionally, equation 5 presents the formula for the output gate, denoted as o_t. The output gate determines the extent to which the output should be updated and plays a role in generating the output data and hidden layer information.

$$i_t = \sigma(w_i * [h_{t-1}, x_t] + b_i \quad (1)$$
$$f_t = \sigma(w_f * [h_{t-1}, x_t] + b_f \quad (2)$$
$$\tilde{c}_t = \tanh(w_f * [h_{t-1}, x_t] + b_c) \quad (3)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (4)$$
$$o_t = \sigma(w_o * [h_{t-1}, x_t] + b_0) \quad (5)$$
$$h_t = o_t * \tanh(c_t) \quad (6)$$

B.2.1. Stacked-LSTM

Stacked-LSTM is an extension of the LSTM model specifically designed for making predictions with time series data. Figure 6 shows the structure of the Stacked-LSTM model. Stacked-LSTM introduces the concept of multiple memory cells and hidden layers, thereby increasing the depth of the neural network. Unlike the traditional LSTM, Stacked-LSTM incorporates multiple layers, enabling the model to generate separate outputs for each cell instead of a single output for the input structure (Arora et al., 2020). This architectural enhancement improves the capacity of the model and its performance in handling complex temporal dependencies.

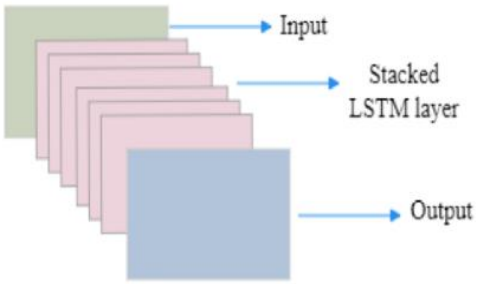


Figure 6. Stacked-LSTM Structure (Arora et al., 2020).

B.2.2. Bi-LSTM

Figure 7 shows structure of the Bi-LSTM model. The standard LSTM model operates in a unidirectional manner, processing inputs only from one direction. To overcome this limitation, the Bi-LSTM model has been developed. Bi-LSTM represents an enhanced version of LSTM that is trained in both forward and backward directions (Tapiwa et al., 2020). Moreover, Bi-LSTM comprises two layers: a forward layer and a backward layer, allowing it to leverage information from both past and future contexts. Bi-LSTM exhibits faster learning capabilities compared to traditional LSTM models (Ayoobi et al., 2021)

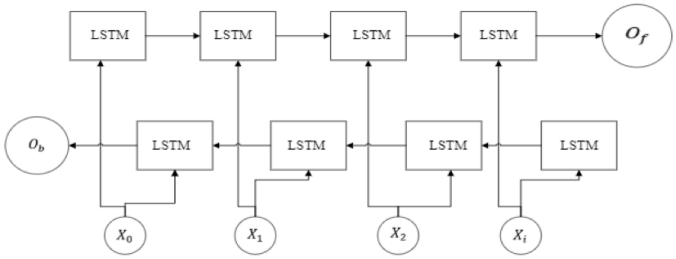


Figure 7. Bi-LSTM Model Structure

B.2.3. Conv-LSTM

The Conv-LSTM model exhibits a resemblance to biological neural networks in terms of its structure. Figure 8 shows structure of Conv-LSTM model. In Conv-LSTM, network neurons possess certain weights and are more tightly connected to each other compared to conventional neural networks. The structure of this method consists of convolutional, batch normalization, rectified linear unit (relu), pooling and a full link layer. The convolutional layer is responsible for feature extraction and feature map generation by filtering inputs. Pooling layer enables easier computation of data manipulation by reducing the size of the feature map and obtaining a subset to process the data. Relu layer is a kind of enabler to preserve non-linear data and size. Fully connected layer provides output by matching attribute outputs appropriately (Mirjalili et al., 2014).

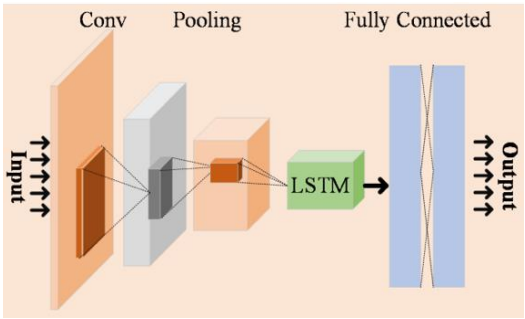


Figure 8. Conv-LSTM Structure (Bao et al., 2021).

B.2.4. GRU

The GRU is a streamlined variant of LSTM that requires less training time and has a smaller number of gates. The computations involved in analyzing data with the GRU model are less complex compared to LSTM. Figure 9 shows structure of GRU model. The GRU architecture features a hidden state cell that utilizes an update gate, which comprises a forget gate (f_t) and an input gate (x_t) (Ataseven, 2013). The update gate (z_t) plays a crucial role in determining the amount of previous information to be incorporated into the next step. It essentially decides the extent of information to be updated. When the update gate value is close to zero, the previous information is disregarded, and only the new information is given importance. In this model, the group cell state and hidden state serve as a single state (Tapiwa et al., 2020). The model operates using the reset gate (r_t). The role of the reset gate is to determine which portions of the previous state information should be deleted (Alkhatib et al., 2022).

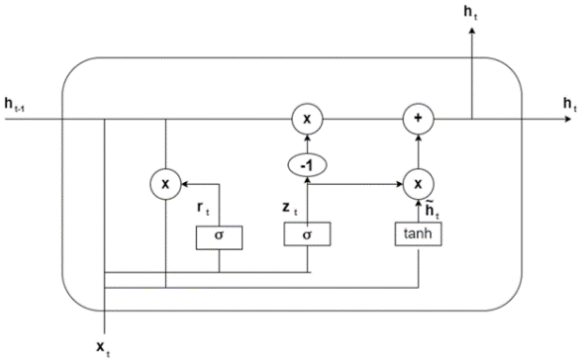


Figure 9. GRU Structure (Alkhatib et al., 2022).

B.3. GWO Algorithm

Mirjalili et al. introduced the GWO algorithm as a herd-based meta-heuristic algorithm. Inspired by the behavior of

gray wolves, such as herd hunting and social hierarchy. It was developed to mimic the collective behavior of wolves in a herd. This algorithm found application in optimizing ANNs in 2015 (Mirjalili et al., 2014).

Many AI algorithms are probabilistic and use control parameters such as population size and recurrence parameters when optimizing problems. The correct utilization of these control parameters is crucial for these algorithms to produce effective results. In the case of the GWO algorithm, the population size and number of iterations are recognized as control parameters. Furthermore, minimizing the control parameters of the GWO algorithm is an objective in itself (Mirjalili et al., 2014). Although neural networks are successful in solving complex problems, their training process can be challenging. Nevertheless, employing optimization algorithms during the training of AI systems lead to enhanced outcomes and greater success rates (Mirjalili et al., 2014). Table 1 shows pdeudocode of the GWO algorithm.

Table 1. Pseudocode of the GWO (Arora et al., 2020).

Algorithm 1 Pseudocode of the GWO

1: Set the maximum number of iterations L
2: Initialize the population $\boldsymbol{X_i}$ (I=1, 2, ….., n)
3: Initialize a, A and C
4: Calculate the fitness of wolves
5: $\boldsymbol{X_a}$ =the best search agent
6: $\boldsymbol{X_\beta}$ =the second-best search agent
7: $\boldsymbol{X_\delta}$ =the third best search agent
8: while (t < L) do
9: for each search agent do
10: Update the position of the current search agents
11: end for
12: Update the a, A and C
13: Calculate the fitness of all search agents
14: Update $\boldsymbol{X_a, X_\beta, and\ X_\delta}$
15: t=t+1
16: end while
17: return $\boldsymbol{X_a}$

B.4. Error Measures

As part of the COVID-19 study conducted in Türkiye, predictive models were developed using AI techniques to forecast the number of cases and fatalities. The effectiveness of these prediction models was assessed using performance metrics. The evaluation of the predictions involved comparing the actual values with the predicted values using specific metrics. Table 2 presents the performance criteria, formulas, and explanations employed in our study.

y_i: actual values, $\overline{y}$: the mean of the actual values, $\hat{y}_i$: predicted values, e_i=y_i-$\hat{y}_i$: error
n: size of dataset (Arora et al., 2020).

Table 2. Performance Criteria

PERFORMANCE CRITERIA	FORMULAS	EXPLANATION
MAE (Mean Absolute error)	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert e_i\rvert$	The absolute error mean is the measured distance between the actual value and the appropriate value. It measures the average size of errors in data prediction. Lower values indicate better performance.
MAPE (Mean Absolute Percentage Error)	$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{\lvert e_i\rvert}{\lvert y_i\rvert}$	The mean absolute percentage error is used to measure and compare the accuracy of the model's prediction.
MSE (Mean Squared Error)	$MSE = \frac{1}{n}\sum_{i=1}^{n} {e_i}^2$	The mean squared error measures the performance of the estimator. Predicted values with MSE values close to zero perform better.
MSLE (Mean Squared logarithmic Error)	$L(A,P) = \frac{1}{n}\sum_{i=0}^{n}(\log(y_i+1) + \log(\hat{y}_i+1))^2$	It is expressed as a measure of the ratio that exists between actual values and predicted values. It only cares about the percentage difference between the actual value and the predicted value.
RMSE (Root Mean Square Error)	$RMSE = \sqrt{\frac{\sum_{i=1}^{n} {e_i}^2}{n}}$	The RMSE (Root Mean Square Error) is a performance metric that quantifies the magnitude of the error between the predicted and true values of a dataset. It represents the standard deviation of the prediction errors. A value of zero for this performance criterion indicates that the model is nearly error-free.

Table 2. (cont) Performance Criteria

R^2 SCORE	$R^2\ (y, \hat{y}) = 1 - \frac{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y_i})^2}{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \bar{y})^2}$	R^2 Score interprets to what extent the independent variable of one model affects the variance value of the other dependent variable [45]. It is preferred when deciding how accurate the predictive value of the model used is. The point value can be up to 1, and the model is successful when the measured value is close to 1.
EVS (Explained Variance Score)	$EVS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i - \bar{e_i})^2}$	It is utilized to assess the degree of fit between model data and actual data. In essence, it measures the variability in the estimates, determining whether there is a strong correlation between the data and a reliable forecast, as indicated by high EVS values.

III. RESULTS AND DISCUSSIONS

In this paper, COVID-19 data for Türkiye was acquired from the Turkovac website, and predictions were made for the future number of cases. Figure 10 illustrates the overall flowchart of the study. Initially, the relevant data to be used in the research was determined and collected. Following data acquisition, necessary preprocessing was performed to facilitate analysis, and the dataset was split into 70% for training purposes and 30% for testing. The selection of the prediction model and the choice of the appropriate optimization algorithm were determined in this study. The predicted outcomes of the model were then compared against the actual values, and the performance of the models was evaluated based on the defined criteria. The analysis was conducted using an Intel (R) Core (TM) i7-8700K CPU @3.70 GHz and 64.0 GB of RAM. For estimating the number of COVID-19 cases, this study utilized Stacked-LSTM, Bi-LSTM, and ANN techniques. The dataset encompassed the number of daily cases, daily deaths, cumulative cases, and cumulative deaths. Specifically, the data for daily deaths were selected from June 13, 2021, while the data for cumulative deaths were chosen from June 11, 2021, to December 7, 2021. As for daily cases, the analysis considered data starting from August 24, 2021, and for cumulative cases, data from August 22, 2021, to December 7, 2021. The data analysis involved using a single input variable to obtain a single output variable.

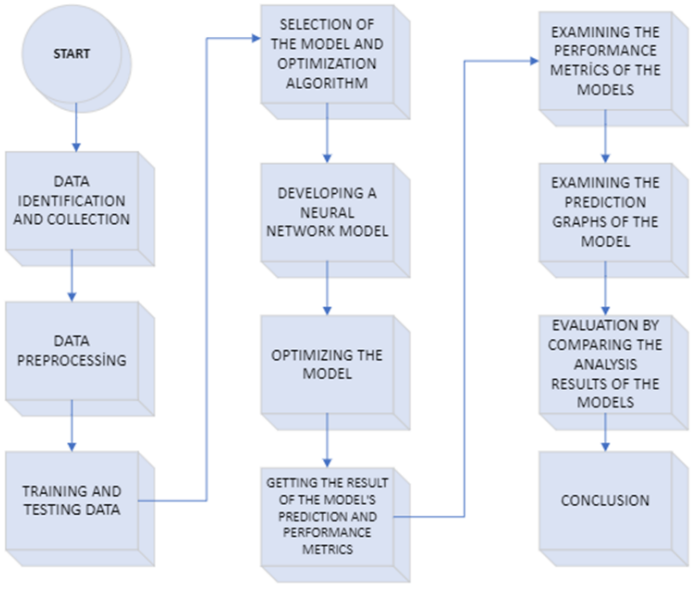


Figure 10. Flowchart of the Study

The GWO algorithm was utilized in our study to optimize the model parameters. The GWO algorithm represents a swarm group, and the specific parameter values used in the study can be found in Table 3 The population size was set to 20, indicating the number of results obtained for each iteration. In this study, the GWO algorithm was run for 100 iterations to optimize the model.

Table 3. GWO Parameters

Parameters	**Parameters values of the GWO**
Number of Iterations	100
Number of Populations	20

Table 4 provides the parameter values used in the study to optimize the LSTM, Bi-LSTM, and Stacked-LSTM models. The table includes information about the following algorithm parameters: node size, optimizer, activation function, step size, epoch size, batch size, and learning rates. The activation functions used in the models include relu, sigmoid, softplus, softsign, and tanh functions. Refer to Table 4 for detailed information on the parameter values.

Table 4. LSTM Parameters

Parameters	Parameter Values of the LSTM Algorithm
Node size	[8, 16, 32, 64, 128]
Optimizers	['Adadelta', 'Adagrad', 'Adam', 'Adamax', 'Ftrl', 'Nadam', 'RMSprop', 'SGD']
Activation functions	['relu', 'sigmoid', 'softplus', 'softsign', 'tanh']
Step size	[2, 4, 6, 8, 10, 12, 14]
Epoch size	[100, 500, 1000]
Batch size	[8, 16, 32, 64]
Learning rate	[0.01, 0.001, 0.0001, 0.00001]

A. Prediction Results using Stacked-LSTM

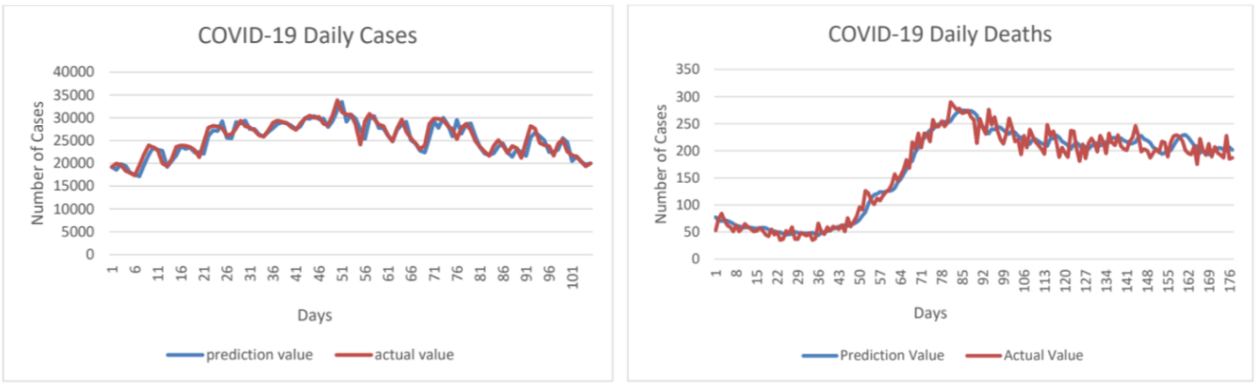
The prediction results for the Stacked-LSTM model were assessed by analyzing the number of daily cases, daily deaths, cumulative cases, and cumulative deaths. Table 5 presents the optimal parameter values selected for the Stacked-LSTM model. It was determined that the model should consist of three hidden layers and three activation functions. The Nadam optimizer was found to be the most suitable optimizer for predicting daily cases, while the Adam optimizer was selected for predicting daily deaths, cumulative cases, and cumulative deaths. The learning rate was set to 0.001 for daily cases and daily deaths, and 0.01 for cumulative cases and cumulative deaths. The epoch size was set to 500 for daily cases, cumulative cases, and cumulative deaths, while for daily deaths, it was set to 100. Table 6 provides information about the CPU time required for running the model.

Table 5. Optimized Stacked-LSTM Parameter Values

	Datasets			
	Daily cases	**Daily Deaths**	**Cumulative cases**	**Cumulative Deaths**
1st Hidden Layer	8	16	16	16
2nd Hidden Layer	128	16	64	8
3rd Hidden Layer	64	8	128	32
Optimizers	Nadam	Adam	Adam	Adam
1st Activation Functions	Softplus	Relu	Softsign	Sigmoid
2nd Activation Functions	Tanh	Sigmoid	Tanh	Softsign
3rd Activation Functions	Tanh	Sigmoid	Tanh	Tanh
Epoch Size	500	100	500	500
Step Size	8	12	8	10
Batch Size	8	8	8	8
Learning Rate	0.001	0.001	0.01	0.01

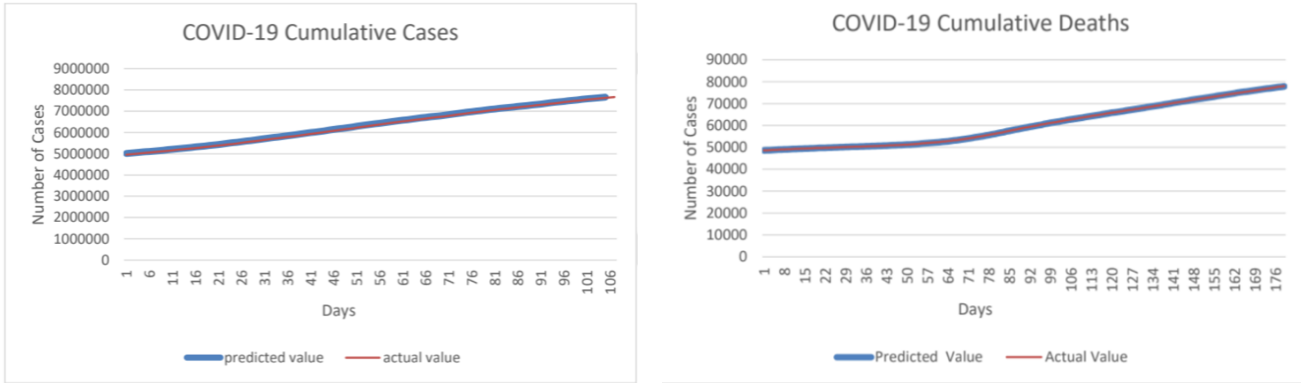
Table 6. CPU times (in seconds) for all datasets

Datasets			
#Of Daily Cases	**# Of Daily Deaths**	**# Of Cumulative Cases**	**# Of Cumulative Deaths**
131584.5	122934.7	126515.4	249028.9



(a) (b)

Figure 11. Predicted Values and Actual Values of COVID-19 Daily Cases (a) and Daily Deaths (b)



(a) (b)

Figure 12. Predicted Values and Actual Values of COVID-19 Cumulative Deaths (a) and Cumulative Cases (b)

Figure 11 and Figure 12 presents the actual and predicted values for daily cases, daily deaths, cumulative cases, and cumulative deaths using the stacked-LSTM model. It is clearly observed from the graph that the predicted values are quite close to the actual values.

B. Prediction Results using Bi-LSTM

In this section, the results of the prediction analysis using the Bi-LSTM model are discussed. The performance of the model was evaluated by comparing the actual values and the predicted results of the number of daily cases, daily deaths, cumulative cases, and cumulative deaths. In Table 7, the parameter results optimized with the GWO algorithm of the Bi-LSTM model are given. 3 hidden layers and 3 activation functions have been determined for the Bi-LSTM model.

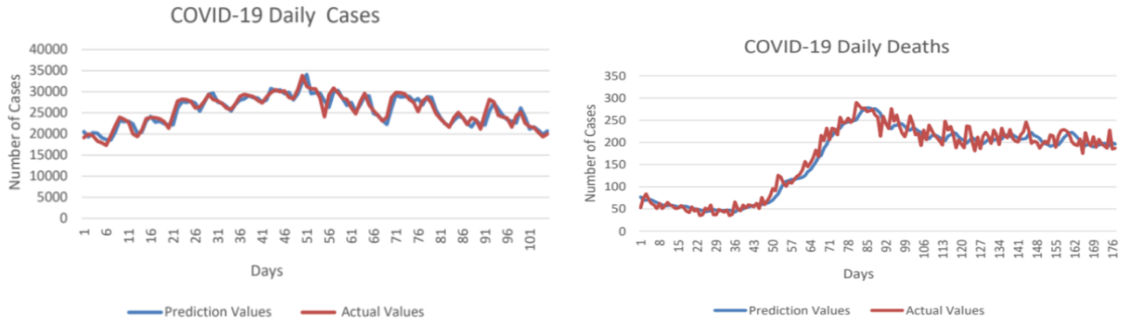
The best optimizer for daily cases, daily deaths, and cumulative cases has been determined to be RMSprop. For cumulative deaths, the best optimizer is Adamax. In the Bi-LSTM model, the epoch size is set to 100 for daily cases, daily deaths, and cumulative cases, while it is set to 1000 for cumulative deaths. The batch size is set to 8 for daily cases, daily deaths, cumulative cases, and cumulative deaths. The learning rate is set to 0.01 for daily cases, 0.00001 for daily deaths, 0.001 for cumulative cases, and 0.001 for cumulative deaths. The activation functions used in the Bi-LSTM model are relu and tanh for daily cases and daily deaths, softsign, relu, and sigmoid for cumulative cases, and sigmoid and tanh for cumulative deaths. Table 8 presents the CPU time of the Bi-LSTM model.

Table 7. Optimized Bi-LSTM Parameter Values

	Datasets			
	Daily Cases	**Daily Deaths**	**Cumulative Cases**	**Cumulative Deaths**
1st Hidden Layer	128	32	16	64
2nd Hidden Layer	128	128	32	32
3rd Hidden Layer	64	128	64	128
Optimizers	RMSprop	RMSprop	RMSprop	Adamax
1st Activation Functions	Relu	Relu	Relu	Sigmoid
2nd Activation Functions	Tanh	Tanh	Softsign	Tanh
3rd Activation Functions	Tanh	Tanh	Sigmoid	Tanh
Epoch Size	100	100	100	1000
Step Size	8	12	6	14
Batch Size	8	8	8	8
Learning Rate	0.01	0.00001	0.0001	0.001

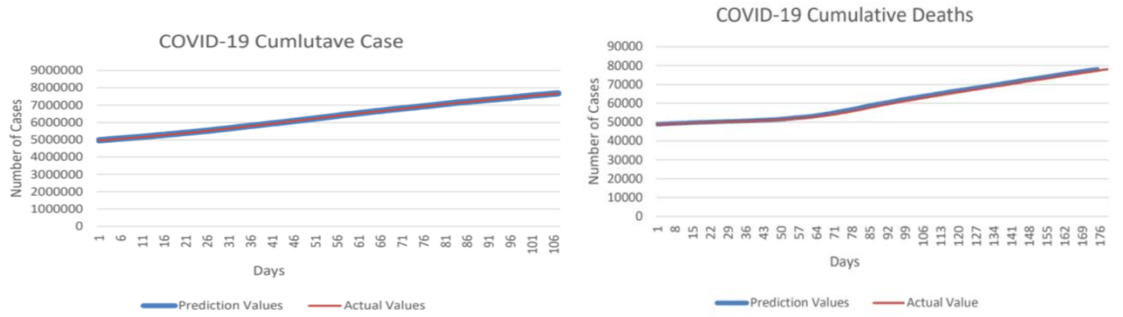
Table 8. CPU Times (in seconds) for All Datasets

Datasets			
#Of Daily Cases	**#Of Daily Deaths**	**#Of Cumulative Cases**	**#Of Cumulative Deaths**
227045.8	233771.8	158317.5	458506.6



(a) (b)

Figure 13. Predicted Values and Actual Values of COVID-19 Daily Cases (a) and Daily Deaths (b)



(a) (b)

Figure 14. Predicted Values and Actual Values of COVID-19 Cumulative Cases (a) and Cumulative Deaths (b)

Figure 13, and Figure 14 presents the actual and predicted values for daily cases, daily deaths, cumulative cases, and cumulative deaths using the Bi-LSTM model. Figure 13 (b) shows the number of deaths stabilizes at around 50 deaths during the first 40 days and this number gradually increases between 40 and 85 days. From the figure 13 (b) it is clearly seen that the actual deaths values fluctuate while the predicted values are more stable. It is clearly observed from the graph that the predicted values are quite close to the actual values.

C. Prediction Results using ANN

In our study, we initially made predictions using the Stacked-LSTM and Bi-LSTM models and evaluated the results. The final prediction model in our study was implemented using the ANN technique. We started by predicting COVID-19 data using the Stacked-LSTM and Bi-LSTM models, which are variants of the LSTM model. The performance of these models was assessed by comparing them. Additionally, we compared the ANN model, which is commonly used in real-life prediction problems in literature, with the LSTM techniques. We examined the accuracy of the models when making predictions. The dataset used for the ANN model consisted of daily cases, daily deaths, cumulative cases, and cumulative deaths.

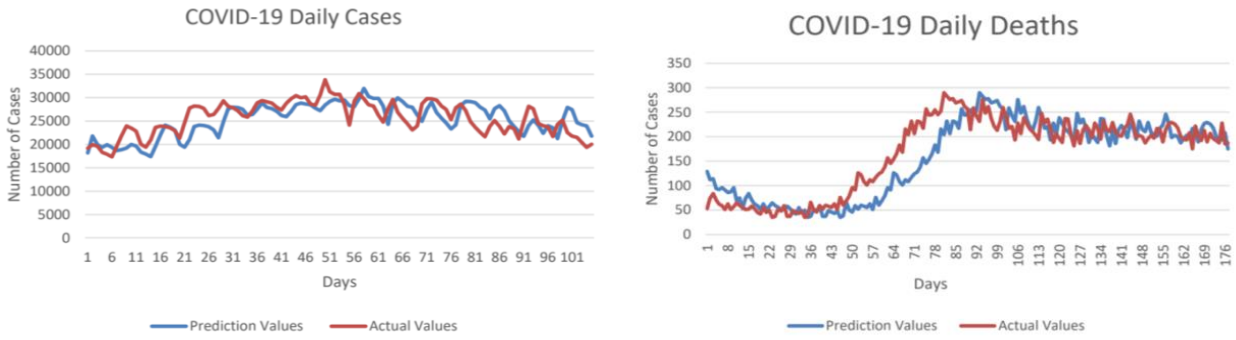
Table 9 provides the optimized parameter values for the ANN model using the GWO algorithm. Like the LSTM and Bi-LSTM models, the ANN model also utilizes three activation functions and three hidden layers. The learning rate and batch size parameters were set to 0.01 and 8, respectively, for the entire dataset. The epoch size was determined as 128 for the input dataset. In the ANN model, adadelta and adam optimizers were found to be the best choices for the daily cases and cumulative cases, respectively. For daily deaths and cumulative deaths, the Adamax optimizer was selected as the optimal choice. Additionally, Table 10 presents the CPU time of the ANN model.

Table 9. Optimized ANN Parameter Values

	Datasets			
	Daily Cases	**Daily Deaths**	**Cumulative Cases**	**Cumulative Deaths**
1st Hidden Layer	128	128	128	128
2nd Hidden Layer	32	16	16	8
3rd Hidden Layer	8	16	16	8
Optimizers	Adadelta	Adamax	Adamax	Adam
1st Activation Functions	Relu	Softsign	Softsign	Softplus
2nd Activation Functions	Sigmoid	Softsign	Softsign	Relu
3rd Activation Functions	Softsign	Softplus	Softplus	Softplus
Epoch Size	128	128	128	128
Step Size	500	1000	1000	1000
Batch Size	8	8	8	8
Learning Rate	0.01	0.01	0.01	0.01

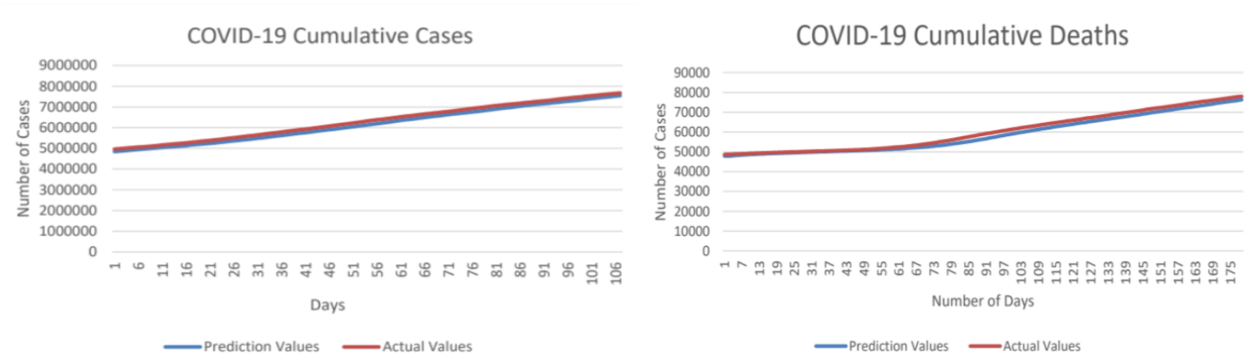
Table 10. CPU times (in seconds) for All Datasets

Datasets			
#Of Daily Cases	**#Of Daily Deaths**	**#Of Cumulative Cases**	**#Of Cumulative Deaths**
74128.81	107528.9	107528.9	161778.8



(a) (b)

Figure 15. Predicted Values and Actual Values of COVID-19 Daily Cases (a) and Daily Deaths (b)



(a) (b)

Figure 16. Predicted Values and Actual Values of COVID-19 Cumulative Cases (a) and Cumulative Deaths (b)

Figure 15 and Figure 16 presents the actual and predicted values for daily cases, daily deaths, cumulative cases, and cumulative deaths using the ANN model. Figure 15 (b) shows the predicted values between Day 50 and Day 90 are far from the real values compared to the whole dataset. It is seen that the predicted values between Day 100 and Day 176 are very close to the actual values.

D. Comparison of Performance Metrics across Models

The performance metrics of the models were analyzed separately for each dataset to determine the model that achieved the highest prediction success. The Bi-LSTM, Stacked-LSTM, and ANN models were evaluated for the daily cases, daily deaths, cumulative cases, and cumulative deaths, respectively.

Table 11. Comparison of Performance Metrics for Daily Cases across Models

	Stacked-LSTM	Bi-LSTM	ANN
EVS	0.956	0.954	0.999*
MAE	0.039	0.0405	0.001*
MAPE	0.096	0.096	0.004*
MSE	0.002	0.003	5.4E-06*
MSLE	0.001	0.001	2.5E-06*
RMSE	0.051	0.053	0.002*
R^2	0.956	0.952	0.999*

Table 11 presents a comparison of the performance metrics for daily measurements across the different prediction models. The R^2 value, which represents the coefficient of determination, is particularly noteworthy. R^2 value close to 1 suggests a better fit between the predicted and actual values. Thus, the ANN model stands out with an R^2 value of 0.999, indicating a strong performance. When considering other performance criteria for daily measurements, it is evident that the ANN model consistently yields more successful results.

Table 12. Comparison of Performance Metrics for Daily Deaths across Models

	Stacked-LSTM	Bi-LSTM	ANN
EVS	0.956	0.954	0.999*
MAE	0.039	0.0405	0.001*
MAPE	0.096	0.096	0.004*
MSE	0.002	0.003	5.4E-06*
MSLE	0.001	0.001	2.5E-06*
RMSE	0.051	0.053	0.002*
R^2	0.956	0.952	0.999*

The performance measures of the models for daily cases are presented in Table 12 When examining the EVS and R^2 values, it is observed that the Bi-LSTM model exhibits higher compatibility compared to the Stacked-LSTM and ANN models. Higher EVS and R^2 values indicate a better fit of the model to the data. Additionally, the Stacked-LSTM model demonstrates a lower MAPE value, which measures prediction accuracy, indicating strong performance in terms of predicting daily cases. On the other hand, when considering MAE, MSE, MSLE, and RMSE values, it becomes evident that the Bi-LSTM model outperforms the other models and achieves more accurate predictions.

Table 13. Comparison of Performance Measures for Cumulative Cases across Models

	Stacked-LSTM	Bi-LSTM	ANN
EVS	0.999	0.999	0.999
MAE	0.001*	0.003	0.004
MAPE	0.003*	0.008	0.004
MSE	2.00E-07*	1.11E-05	5.4E-06
MSLE	1.00E-06*	5.33E-06	2.5E-06
RMSE	0.001*	0.003	0.002
R^2	0.999	0.999	0.999

Table 13 presents a comparison of the performance measures for cumulative cases across the models. Upon examining the EVS and R^2 values in the table, it is observed that all models yield similar results. These high EVS and R^2 values indicate strong performance for all models in predicting cumulative cases. Additionally, when analyzing the MAE, MAPE, MSE, MSLE, and RMSE values, it becomes evident that the models exhibit excellent performance as these values are consistently low. However, upon considering these performance metrics, the Stacked-LSTM model emerges as the most successful model for predicting cumulative cases.

Table 14. Comparison of Performance Metrics for Cumulative Deaths across Models

	Stacked-LSTM	Bi-LSTM	ANN
EVS	0.999	0.999	0.999
MAE	8.32E-04	7.52E-04*	0.001
MAPE	0.002*	0.002*	0.004
MSE	1.40E-06	9.00E-07*	1.8E-06
MSLE	5.00E-07*	5.00E-07*	1.1E-06
RMSE	0.001	0.001	0.001
R^2	0.999	0.999	0.999

Table 14 provides a comparison of the performance metrics for cumulative death measurements across the models. When examining the EVS, RMSE, and R^2 performance values, it is evident that all models yield similar results, making it challenging to distinguish between them based on these metrics. However, when considering the MAE value, the Bi-LSTM model stands out as the most successful, surpassing the other models. Furthermore, in terms of MAPE and MSLE values, both the Stacked-LSTM and Bi-LSTM models demonstrate superior predictive capabilities compared to the ANN model. Moreover, a low MSE value indicates strong performance as it approaches zero. In this regard, when evaluating the models based on the MSE criterion for cumulative death measurements, the Bi-LSTM model demonstrates more successful outcomes.

IV. CONCLUSION

The current COVID-19 pandemic has rapidly spread across the globe and continues to pose a significant threat to human health. In response, authorities have conducted extensive research to mitigate the pandemic, gain insights into

the virus, and develop effective treatment strategies. One crucial area of study focuses on estimating the disease's transmission rate. In this regard, techniques such as AI are commonly employed to provide accurate estimations.

In this research, we aimed to predict the number of COVID-19 cases and deaths in Türkiye using the Bi-LSTM, Stacked-LSTM, and ANN techniques. The COVID-19 data used in the study was obtained from the Turkovac website. Additionally, we employed the GWO algorithm to optimize the models. The analysis was conducted separately for daily and cumulative case and death numbers. By comparing the actual and predicted values, we evaluated the performance of the models. Furthermore, the GWO algorithm was applied to optimize the prediction models, considering three different hidden layers and activation functions. Upon analyzing the MAPE values, it was determined that the Stacked-LSTM model exhibited the highest prediction accuracy when applied to the dataset with cumulative measurements. On the other hand, the Bi-LSTM model demonstrated the best prediction accuracy for the cumulative deaths' dataset. In terms of the ANN model, it was observed that the dataset with the lowest prediction accuracy was the daily cases dataset. Notably, the cumulative deaths dataset consistently yielded the highest prediction accuracy across all models, which is further supported by the MAPE values. In general, machine learning, deep learning, and AI-based approaches such as ANN offer valuable tools for making strategic decisions and taking measures during unexpected situations like pandemics. Evaluating the performance values of the LSTM, Bi-LSTM, and ANN models used in this study, it was observed that these models achieved successful outcomes when applied to pandemic data. Interestingly, each technique excelled in different datasets. The Bi-LSTM model demonstrated the most accurate predictions for daily cases, while the ANN model performed best for daily deaths. However, the ANN model exhibited lower performance for cumulative cases and cumulative deaths. Overall, this study highlighted the effectiveness of LSTM and ANN techniques in estimating the dynamics of an epidemic situation.

DECLARATIONS

Acknowledgements: The author/authors do not wish to acknowledge any individual or institution.

Author Contributions: Conceptualization, M.G., A.T.D., S.O.; Methodology, M.G., A.T.D., S.O.; Validation, M.G., A.T.D., S.O.; Investigation, S.O.; Resources, S.O.; Data Curation, S.O.; Writing—Original Draft, S.O.; Writing—Review & Editing, M.G., A.T.D.; Supervision, M.G.

Conflict of Interest: The author/authors declare no conflict of interest.

Supporting Institutions: This research received no external funding.

Ethical Approval: This study does not involve human or animal participants. All procedures followed scientific and ethical principles, and all referenced studies are appropriately cited.

Plagiarism Statement: This article has been evaluated for plagiarism and no instances of plagiarism were detected.

Use of AI Tools: The author/authors declare that no Artificial Intelligence (AI) tools were used in the creation of this article.

REFERENCES

Alkhatib, K., Khazaleh, H., Alkhazaleh, H. A., Alsoud, A. R., & Abualigah, L. (2022). A new stock price forecasting method using active deep learning approach. Journal of Open Innovation: Technology, Market, and Complexity, 8(2), Article 96. https://doi.org/10.3390/joitmc8020096

Anderson, D., & McNeill, G. (1992). *Artificial neural networks technology* (Technical Report). Kaman Sciences Corporation.

Arora, P., Kumar, H., & Panigrahi, B. K. (2020). Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India. *Chaos, Solitons & Fractals*, *139*, Article 110017. https://doi.org/10.1016/j.chaos.2020.110017

Ataseven, B. (2013). Yapay sinir ağlari ile öngörü modellemesi. *Öneri Dergisi*, *10*(39), 101-115.

Ayoobi, N., Sharifrazi, D., Alizadehsani, R., Shoeibi, A., Gorriz, J. M., Moosaei, H., Khosravi, A., Nahavandi, S., Gholamzadeh Chofreh, A., Goni, F. A., Klemeš, J. J., & Mosavi, A. (2021). Time series forecasting of new cases and new deaths rate for COVID-19 using deep learning methods. *Results in physics*, *27*, Article 104495. https://doi.org/10.1016/j.rinp.2021.104495

Bahri, S., Kdayem, M., & Zoghlami, N. (2020, December). Deep learning for COVID-19 prediction. In *2020 4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)* (pp. 406-411). IEEE.

Bao, B., Xu, Z., Li, C., Sun, Z., Liu, S., & Li, Y. (2021). TDTS: Three-dimensional traffic scheduling in optical fronthaul networks with Conv-LSTM. *Photonics, 8*(10), Article 451. https://doi.org/10.3390/photonics8100451

Crokidakis, N. (2020). Modeling the early evolution of the COVID-19 in Brazil: Results from a Susceptible–Infectious–Quarantined–Recovered (SIQR) model. *International Journal of Modern Physics C*, *31*(10), Article 2050135. https://doi.org/10.1142/S0129183120501351

Devaraj, J., Elavarasan, R. M., Pugazhendhi, R., Shafiullah, G. M., Ganesan, S., Jeysree, A. K., Khan, I. A., & Hossain, E. (2021). Forecasting of COVID-19 cases using deep learning models: Is it reliable and practically significant? *Results in Physics*, *21*, Article 103817. https://doi.org/10.1016/j.rinp.2021.103817

Er, F., & Bal, C. (2020). COVID-19 ve Grafiksel Veri Analizi. *Osmangazi Tıp Dergisi*, *42*(4), 450-461.

Er, M. B., & Işık, İ. (2021). LSTM tabanlı derin ağlar kullanılarak diyabet hastalığı tahmini. *Türk Doğa ve Fen Dergisi*, *10*(1), 68-74.

Irmak, S., Köksal, C. D., & Asilkan, Ö. (2012). Hastanelerin gelecekteki hasta yoğunluklarının veri madenciliği yöntemleri ile tahmin edilmesi. *Uluslararası Alanya İşletme Fakültesi Dergisi*, *4*(1), 101-114.

Istaiteh, O., Owais, T., Al-Madi, N., & Abu-Soud, S. (2020, October). Machine learning approaches for covid-19 forecasting. In *2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)* (pp. 50-57). IEEE.

Khare, N., Jha, M., Mathur, R., & Jha, A. K. (2020). Data analysis for COVID-19. *The International Journal of Analytical and Experimental Modal Analysis*, *12*(5), 960-974.

Lalmuanawma, S., Hussain, J., & Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons & Fractals*, *139*, Article 110059. https://doi.org/10.1016/j.chaos.2020.110059

Marzouk, M., Elshaboury, N., Abdel-Latif, A., & Azab, S. (2021). Deep learning model for forecasting COVID-19 outbreak in Egypt. *Process Safety and Environmental Protection*, *153*, 363-375. https://doi.org/10.1016/j.psep.2021.07.034

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Picariello, M., & Aliani, P. (2020). Covid-19: Data analysis of the Lombardy region and the provinces of Bergamo and Brescia. *arXiv. https://arxiv.org/abs/2003.10518*

Sazlı, M. H. (2006). A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, *50*(01), 11-17. https://doi.org/10.1501/commua1-2_0000000026

Scientific Committee Study COVID-19 (SARS-CoV-2) Infection Guide. (2020). *T.C. Ministry of Health General Directorate of Public Health.*

Sengupta, S. (2020). Forecasting the peak of covid-19 daily cases in India using time series analysis and multivariate LSTM [Preprint]. *EasyChair.*

Shahid, F., Zameer, A., & Muneeb, M. (2020). Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. *Chaos, Solitons & Fractals*, *140*, Article 110212. https://doi.org/10.1016/j.chaos.2020.110212

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Science*, *6*(12), 310-316.

T.C. Sağlık Bakanlığı. (2021, December 7). COVID-19 bilgilendirme platformu. https://covid19.saglik.gov.tr/

Tandon, H., Ranjan, P., Chakraborty, T., & Suhag, V. (2022). Coronavirus (COVID-19): ARIMA-based time-series analysis to forecast near future and the effect of school reopening in India. *Journal of Health Management*, *24*(3), 373-388. https://doi.org/10.1177/09720634221109087

Tapiwa, G., Cécile, K., Dongxuan, C., Andrea, T., Christel, F., Jacco, W., & Niel, H. (2020). Estimating the generation interval for COVID-19 based on symptom onset data [Preprint]. *medRxiv.* https://www.medrxiv.org/content/10.1101/2020.03.05.20031815v1

Tüzüntürk, S. (2010). Veri madenciliği ve istatistik. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, *29*(1), 65-90.

Uğur, A., & Kınacı, A. C. (2006). Yapay zeka teknikleri ve yapay sinir ağları kullanılarak web sayfalarının sınıflandırılması. In *XI. Türkiye'de İnternet Konferansı (inet-tr'06),* Ankara, Türkiye.

Vaishya, R., Javaid, M., Khan, I. H., & Haleem, A. (2020). Artificial intelligence (AI) applications for COVID-19 pandemic. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, *14*(4), 337-339. https://doi.org/10.1016/j.dsx.2020.04.012

Wang, P., Zheng, X., Ai, G., Liu, D., & Zhu, B. (2020). Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran. *Chaos, Solitons & Fractals*, *140*, Article 110214. https://doi.org/10.1016/j.chaos.2020.110214

Waqas, M., Farooq, M., Ahmad, R., & Ahmad, A. (2020). Analysis and prediction of COVID-19 pandemic in Pakistan using time-dependent SIR model [Preprint]. *arXiv.* https://arxiv.org/abs/2005.02353

World Health Organization Regional Office for Europe. (n.d.). *Coronavirus disease (COVID-19) outbreak: About the virus*. Retrieved December 7, 2021, from https://www.euro.who.int/en/health-topics/health-emergencies/coronavirus-covid-19/novel-coronavirus-2019-ncov

Yönem, E., & Akay, R. (2020). Yapay arı koloni algoritması ile eğitilmiş tekrarlayıcı sinir ağlarının robot navigasyonu için kullanılması. *Avrupa Bilim ve Teknoloji Dergisi*, 318-324. https://doi.org/10.31590/ejosat.araconf41

Zeroual, A., Harrou, F., Dairi, A., & Sun, Y. (2020). Deep learning methods for forecasting COVID-19 time-series data: A comparative study. *Chaos, Solitons & Fractals*, *140*, Article 110121. https://doi.org/10.1016/j.chaos.2020.110121