COMBINING COVID-19 CASE PREDICTION AND ANALYSIS OF SEASONAL DATA IMPACTS USING DEEP LEARNING METHODS

YİĞİTCAN İPEKÇİ

MAY 2021

COMBINING COVID-19 CASE PREDICTION AND ANALYSIS OF SEASONAL DATA IMPACTS USING DEEP LEARNING METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY

BY
YİĞİTCAN İPEKÇİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING
DEPARTMENT

MAY 2021

ABSTRACT

COMBINING COVID-19 CASE PREDICTION AND ANALYSIS OF SEASONAL DATA IMPACTS USING DEEP LEARNING METHODS

İPEKÇİ, Yiğitcan
M.Sc., Department of Computer Engineering
Supervisor: Prof. Dr. Hayri SEVER

MAY 2021, 73 pages

The new coronavirus (Covid-19) epidemic, which has affected the whole world, has infected millions of people and caused the death of hundreds of thousands, even millions of people. This epidemic has greatly affected the economics and social life of many countries. The measures taken could not prevent this and the society was caught unprepared. Estimating the rate of increase in the number of cases, is of great importance especially in the planning of administrative processes related to health infrastructure. Mathematical models and deep learning methods are used for these predictions. It is also being developed in various artificial intelligence-based approaches. In this study, in order to predict the changes in the number of COVID-19 cases in Italy, a forward forecast is made with a long short-term memory (LSTM) based neural network approach. In the study, the number of daily cases, deaths and recovered patients in Italy between Feb 24 and Nov 1, 2020 were used. In addition to, the effects of seasonal changes on the epidemic are analyzed using the meteorological data of this period. In addition to this, considering the 14-day incubation period in the COVID19 outbreak, the effect of historical values of meteorological parameters on cases is demonstrated by experimental studies.

The results show that the Long short-term memory (LSTM) method can provide a significant advantage in case prediction to take preventive steps, and at the same time, seasonal data are added to the LSTM network, increasing the success rates in case prediction.

Keywords: Neural Network, Deep Learning, Long Short-Term Memory (LSTM), Pandemic, COVID-19.

ÖZ

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK ÖZGEÇMİŞLER ÜZERİNDE ANAHTAR KELİME ÇIKARIMI

İPEKÇİ, Yiğitcan
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi: Prof. Dr. Hayri SEVER

MAYIS 2021, 73 sayfa

Tüm dünyayı etkisi altına alan korona virüs (COVID-19) salgını milyonlarca insanı etkilemiş, yüzbinlerce hatta milyonlarca kişinin ölümüne neden olmuştur. Bu salgın birçok ülkenin ekonomik ve sosyal hayatını büyük ölçüde etkiledi. Alınan tedbirler bunu engelleyemedi ve toplum hazırlıksız yakalandı. Vaka sayısındaki artış oranının tahmin edilmesi, özellikle sağlık altyapısı ile ilgili idari süreçlerin planlanmasında büyük önem taşımaktadır. Bu tahminler için matematiksel modeller ve derin öğrenme yöntemleri kullanılır. Ayrıca farklı türdeki yapay zekâ temelli yaklaşımlarda geliştirilmektedir. Bu çalışmada, İtalya'daki COVID-19 vakalarının sayısındaki değişiklikleri tahmin etmek için, uzun kısa süreli bellek (LSTM) tabanlı sinir ağı yaklaşımı ile ileriye dönük bir tahmin yapılmıştır. Çalışmada 24 Şubat- 1 Kasım 2020 tarihleri arasında İtalya'da günlük vaka, ölüm ve iyileşen hasta sayıları kullanıldı. Ayrıca mevsimsel değişikliklerin salgın üzerindeki etkileri de bu döneme ait meteorolojik veriler kullanılarak analiz edilmiştir. Buna ek olarak, COVID19 salgınında 14 günlük kuluçka dönemi dikkate alınarak, meteorolojik parametrelerin geçmiş değerlerinin vakalar üzerindeki etkisi deneysel çalışmalarla ortaya konmuştur. Sonuçlar, uzun kısa süreli bellek (LSTM) yönteminin önleyici adımların atılması durumunda önemli bir avantaj sağlayabileceğini ve aynı zamanda LSTM ağına mevsimsel verilerin eklenerek vaka tahmininde başarı oranlarının arttığını göstermektedir.

Anahtar Kelimeler: Derin öğrenme, Yapay Sinir Ağları, Uzun Kısa Süreli Bellek, Pandemi, COVID-19.

ACKNOWLEDGEMENTS

Throughout the study, I would like to thank Prof. Dr. Hayri SEVER, who always guided me with his knowledge and experiences, and who did not refrain from his support and who tried to help me in all matters. It is a pleasure to express my special thanks to my family and my friends for their valuable support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS INDEX

ANN: Artificial Neural Network

MLP: With Multi-Layer Sensor

LSTM: Long-Short Term Memory (Long-Short Term Memory Network)

RNN: Recurrent Neural Network

GRU: Gated Recurrent Units

RMSE: Root Mean Squared Error

MSE: Mean Squared Error

MAE: Mean Absolute Error

MAPE: Mean Absolute Percentage Error

CHAPTER I

INTRODUCTION

The world has been struggling with a pandemic caused by a new type of coronavirus (SARS-CoV-2) since it was discovered in China in December 2019. Almost all countries have been affected by the new coronavirus (COVID-19) outbreak, and Italy is one of the most affected European countries. As of May 15, the total number of positive cases exceeded 223,885 and the number of deaths exceeded 31,000. Thanks to the use of the internet, new information and data are being created and their number is gradually increasing. The number of data increases considerably through social media and question-answer sites, forums, and sharing sites, and people want to find the right one for themselves.

The new coronavirus epidemic (COVID-19) has created a great chaos around the world in a very short time. As of October, more than 40 million official cases have been reported worldwide and the number of deaths due to COVID-19 has exceeded 1 million. Many countries are developing various policies to deal with this epidemic and to minimize its effects. It is very important to taking precaution for COVID-19 and similar outbreaks.

Science and technology contribute greatly to precautionary policies applied in this sense. One of the most important of these contributions is the ability to predict how the outbreak will progress in ongoing times.

In this context, two basic approaches emerge. The first of these are statistical approaches and mathematical models. The second approach is AI-based approaches that have gained more attention in recent years. Artificial intelligence-based algorithms, which have proven themselves in many fields, have been the focus of attention of researchers in outbreak prediction. Over the past 30 years, these algorithms have been used to predict many infectious diseases such as dengue, influenza, ebola

and malaria. In the dengue epidemic that occurred in India, Long-Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) methods were applied on the number of cases recorded between 2017 and 2018, and Some analyzes are provided on how the epidemic is affected by weather conditions.

1.1 GENERAL OVERVIEW

Again, on the dengue epidemic, epidemic estimation is made using data collected from 5 different states between 2010 and 2017 in Brazil. In this study, a comparative analysis with a total of 442 weeks of case numbers, air temperature, humidity and atmospheric pressure is presented, using various regression models [1].

The LSTM method is used for the prediction of the spread of influenza virus between 2002 and 2017 in the United States. Using the weekly number of cases between these dates as data, the results of various parameters and models for the LSTM network are presented [2]. In the malaria epidemic, the spread estimation is made using a data set that includes positive cases, epidemic formation and some weather data between 2011- 2014 in India. In this study, Naive Bayes, Random Forest and Logistic Regression methods were used for outbreak prediction [3]. The LSTM method has been applied using the number of cases in India for COVID-19 prediction. In the study, coronaviruses such as Ebola and Mers are presented in comparison with COVID-19. In addition, in the China region, LSTM and GRU methods are used in case prediction as a hybrid for COVID-19 spread All these approaches attempt to predict the future spread of the outbreak depending on the number of cases [4]. There are many external factors such as weather data that affect the spread of cases. Especially for the COVID-19 outbreak, weather data such as air temperature, humidity, atmospheric pressure are used to predict case rates [5]. Again, there is Google trend data on case rates. Also, there are case rates used in data such as social media, news headline analysis and google trend data. The effects of quarantine decisions and restrictions on the course of cases are being investigated. Statistical information about the age, gender and observed symptoms of the patients are also used in studies as parameters affecting the results in case estimation. In addition, studies are conducted to model the impact of the epidemic on healthy people and how it spreads [6][7].

When the existing studies are examined, it is seen that the LSTM method is included in most of the studies and gives successful results. Also, there are many studies on both direct case numbers and mixed data sets. In this study, a forward forecast for the COVID-19 outbreak is made with the LSTM method. The method applied in many countries, cases, estimates of changes in death rate and improving patient numbers. In addition, seasonal data are added to the LSTM network, and the impact of various weather data on the epidemic is analyzed. However, the effect of past seasonal data on cases is analyzed. The results obtained from the experimental studies show the success of the LSTM method. By determining the effects of meteorological parameters, the effects of the past values of these parameters on the predictions are also presented in experimental studies. This study prepared consists of 5 main parts. The first section includes a brief summary of the subject and a literature review on artificial intelligence approaches used in epidemic prediction. In the second part, information about epidemic diseases is given. All-important outbreaks in the literature since the first recorded epidemic are described in this section. It also contains information about various methods used in epidemic prediction. The third section contains detailed information about the methods to be used in experimental studies. Basic artificial neural network architecture and deep learning methods based on this architecture are explained in detail. The fourth section includes experimental studies. Information about the data set used and pre-treatment steps are explained in this section. All findings obtained are interpreted and presented in accordance with the purpose of the study. In the fifth and last section, all analyzes, and comparisons related to the study are interpreted. There are also some suggestions for future studies.

1.2 RELATED WORKS

Sajadi et al: proposed a simplified model that presents a zone in which the risk of COVID-19 spread is increased. Using weather modeling, it allows for the intensification of public health efforts to contain the spread of the infectious virus by predicting areas at highest risk of significant community spread of COVID-19 in the coming weeks [8].

Demongeot et al: showed that the virulence of coronavirus diseases due to viruses such as SARS-Cov and MERS-Cov decreased in humid and hot weather conditions. The predicted temperature dependence of the new coronavirus contagion, namely COVID-19, has attracted great attention in the medical field.

Similarly, our study aims to identify important parameters from COVID-19 propagation dynamics, such as the coefficient of infection that increases with cold, warm or dry air, potentially temperature dependent parameters [9].

CHAPTER II

EPIDCEMICS AND DETECTION METHODS

Epidemics have claimed millions of lives throughout human history and have produced crises that require protracted struggles. While most of these outbreaks disappeared within a few years, some continued for decades, claiming millions of lives. Some of the microbes that carry these epidemics have adapted to live with humans, thanks to developing medical technologies and vaccines, have lost their lethality to a great extent and are still actively involved in our lives.

2.1 DISEASE FROM PAST TO PRESENT

From the past to the present, many epidemics that have deeply affected humanity have been recorded in every century. The Black Plague (Black Death), which started in the south west of the Asian continent in the 14th century and then became effective in Europe and destroyed a third of the population, is among the most devastating epidemics in history.

Smallpox, which emerged in the 17th century, killed more than 20 million people at that time. The first vaccine known in the 18th century was used for the treatment of smallpox. Cholera epidemic, which first appeared at the beginning of the 19th century, was effective in Asia and Europe. The cholera epidemic, which was also effective in our country between 1912 and 1913, caused more than 100 thousand people to die.

In addition to these, many epidemics such as hemorrhagic fever, smallpox, yellow fever, measles, typhus, malaria have put an end to the lives of millions of people until the 20th century. In addition, these epidemics caused great chaos for people at that time, both economically and socially.[11][12]

Table 2.1: Some of the past epidemics [13]

Epidemic	Date	Country	# Of Death
Black Plague	1347- 1357	Europe	100 million
Bleeding Fever	1545- 1548	Mexico	15 million
Smallpox	1600- 1762	Worldwide	20 million
Cholera	1817- 1824	Asia and Europe	100 thousand
Typhus Disease	1847- 1848	Canada	20 thousand

The Spanish flu is undoubtedly the most devastating epidemic in the last 100 years. This epidemic caused by the H1N1 virus infected 500 million people worldwide in a short period of eighteen months between 1918 and 1920 and killed nearly 50 million people. In the following years, there were many less deadly influenzas outbreaks caused by different mutated versions of this virus.[14]

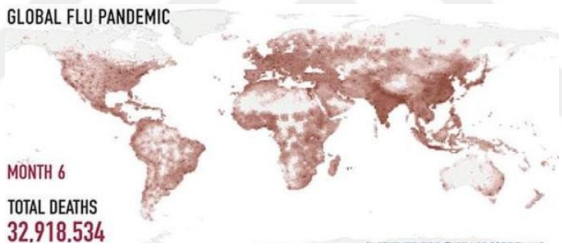


Figure 2.1: Simulation of the global Spanish Flu pandemic model [13]

The HIV / AIDS virus epidemic, whose effects are continuing today, was first recorded in 1981. This ongoing epidemic has killed more than 30 million people so far. Swine flu, which broke out in the USA and Mexico between 2009-2010, caused the death of more than 200 thousand people all over the world.

Table 2.2: Deadliest pandemics of recent times [13]

Epidemic	Date	Country	# Of Death
Spanish Flu	1918- 1920	Worldwide	50 million
Asian Flu	1957- 1958	Worldwide	2 million
HIV / AIDS	1906- ...	Worldwide	770 thousand
SARS	2002- 2003	West Africa	775
Swine Flu	2009- 2010	Asia - Canada	200 thousand
MERS	2012- ...	Worldwide	450
Covid-19	2019- ...	Worldwide	1 million

2.2 CORONAVIRUSES

These viruses, which were initially seen in animals, were first reported to be transmitted to humans in 2002. One of the ongoing coronaviruses today is the MERS virus. This virus, which has a killing rate of over 30%, first appeared in Arabia in 2012 and took 450 lives. This epidemic, called SARS, ended in 2003 and according to the records, 775 lives in 29 countries. Known human coronaviruses can be grouped into 7 groups:

i. HCoV-229E
ii. HCoV-OC43
iii. SARS-Cov
iv. HCoV-NL63
v. HKU1
vi. MERS-Cov
vii. COVID-19

The last coronavirus outbreak occurred in Wuhan, China, in December 2019. The rate of spread of this virus is very high, which is called COVID-19, Its infected millions of people in a short time and caused the death of 100 thousand of people [15][16].

Figure 2.2: Simulation of the global COVID-19 pandemic model [13]

2.3 METHOD AND ESTIMATION MODEL

There are two basic approaches for predicting forward in time-based data. These are statistical methods, mathematical models and artificial intelligence-based approaches.

The first statistical approaches for epidemic analysis were introduced in the early 1920's for process control. After this date, mathematical models and statistical analysis started to be used widely. Artificial intelligence-based approaches were first used in medicine in the 1970s as auxiliary programs in clinical decisions. Today, artificial intelligence approaches are increasingly accepted [17].

2.3.1 Times Series

Time series are data points obtained regularly over consecutive time intervals. It is a conceptual model that can make predictions about the future based on previously known information. Time series analysis aims to obtain meaningful statistics and new information from existing information. Observing some features of time series, such as seasonal data, trends or cyclical data, helps to understand the data and to choose correct forecasting methods [18][19].

2.3.2 Time Series Forecasting

Time series forecasting is the study of complex models to predict future values based on previously observed data. When we want to make predictions about the future using historical data, we use quantitative prediction techniques.

Prediction is a prediction of some future events or events. In order to analyze the performance of a model, it is necessary to know the difference between the estimated model and the actual value [19][20].

There are statistics used to measure the predictive accuracy (forecast performance) of the models. Among them, the methods we use are as follows.

$$\mathbf{E_t} = y_t - \hat{y}_t \quad (2.1)$$

Where:

- $\mathbf{E_t}$ is the residual (the prediction error at time t),
- y_t is the observed or actual value,
- $\hat{y}_t$ is the predicted value.

The of the most preferred performance criteria in the literature given in the figures below with error calculation metrics. 80% of the data set is reserved for training each algorithm and the remaining 20% is allocated to the test set to measure the accuracy of the predictions produced based on this training.

The test set was compared with the estimates of the algorithms and the values produced, and this comparison was evaluated with the performance metrics of mean absolute percent error (MAPE), root mean square deviation (RMSE) , mean absolute error (MAE) and Mean Squared Error (MSE).In line with the evaluations made with performance metrics, the performance of machine learning models was compared and suggestions were made on possible scenarios on the best performing model.

General formula for **Mean Squared Error (MSE)**. Simply, mean square error tells you how close a regression curve is to a set of points.

Let $\hat{x} = g(Y)$ be an estimator of the random variable x, given that we have observed the random variable y. The mean squared error (MSE) of this estimator is defined as in equation 2.2.

The formula is:

$$MSE = \frac{1}{N}\sum_{(x,y)\in D}^{n} (y - (prediction(x))^2 \quad (2.2)$$

Where:

- x: the set of features that the model used to make predictions,
- Prediction (x): function of the weights and bias in combination with the set of features x,
- y: the real value (for example, temperature),
- D: data set containing many labeled examples, which are (x, y) pairs,
- N: the number of example D.

The **Root Mean Square Error (RMSE)** standard deviation formula of the residuals (prediction errors). It indicates how dense that data is around the best fitting line given to us.

The formula is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)^2} \quad (2.3)$$

Where:

- n: number of observations,
- f: Actual value,
- o: predicted value,

The **Mean Absolute Error (MAE)** is the average of all absolute errors. Average absolute error is the measure of the difference between two continuous variables. The formula is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|X_i - X| \quad (2.4)$$

Where:

- n: number of observations,
- Σ: summation symbol (which means “add them all up”),
- X: predicted value,
- Xi: Actual value.

The **mean absolute percentage error (MAPE)** is a statistical measure of how accurate a forecast system. In regression and time series models, the mean percent absolute error is often used to measure the accuracy of predictions. The formula is:

Where:

- At: is the actual value,
- Ft: is the forecast value
- n: number of observations.

CHAPTER III

METHODOLOGY

Artificial learning is the ability of computers to gain problem solving ability by digitizing sample data and experiences obtained from existing information. In a broader sense, artificial learning is the imposition of a human-specific learning function to another object that is controlled. Since these objects are computers today, this controlled learning function is provided by the programming of computers [21].

3.1 Artifical Neural Network

Artificial neural networks (ANN) have been developed inspired by the human brain. They are structures that are connected to each other by weighted links in their structure and each consisting of processing elements with its own memory. These structures process information in parallel and distributed manner. In other words, Artificial neural networks (ANNs) are computer systems that can learn by imitating the human brain and derive new information, make predictions using this learned and derived information, and produce reactions against events that occur in the system it is in [21][22]. ANNs started with the interest of people in neurobiology and were shaped by applying the information they obtained in this field to computer science. The working principles and functions of the human brain have been studied for many years.

The first work containing information about these functions of the brain was published in 1890. However, the first studies with engineering value started after the 1940's [22][23].

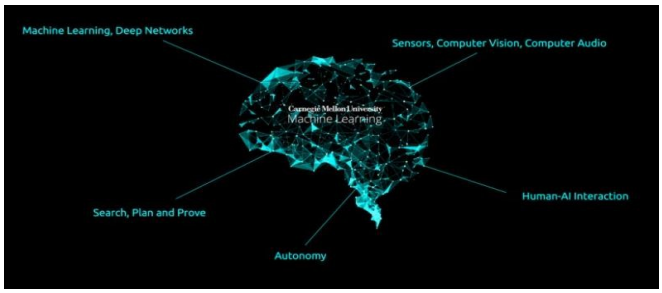


Figure 3.1: Simulation of the AI model [24][25]

3.1.1 Biological Neural Network Cell

Biological neural networks are structures formed by the combination of many nerve cells in our brain. These structures are realized by approximately 10 billion nerve cells in the human brain and 60 trillion connections that these neurons make with each other. These nerves receive input data from the sensory organs, process these signals through the carrier nerves and transfer them to the next nerve, allowing the signal to reach the central nervous system. After the central nervous system receives and interprets these signals, it generates response signals against inputs. In this way, the appropriate signals to the reaction organs against the information coming from the sensory organs are sent via the nervous system and reactions against the situations coming from the environment are produced.

Part of the neural network that enables the inputs from sensory organs to be transported to the central nervous system is shown in Figure 3.2 [26][27][28].

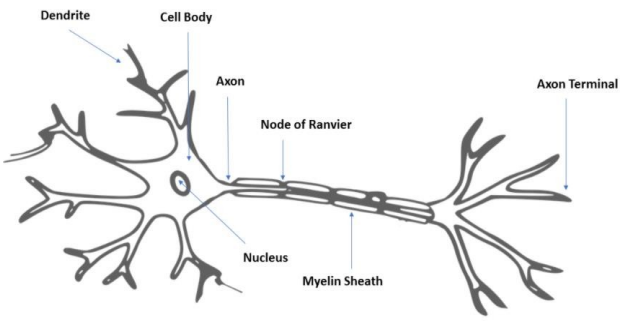


Figure 3.2: Biological Neuron [26][27][28].

3.1.2 Artificial Neural Network Cell

Learning in biological systems occurs by adjusting the synaptic (synaptic) connections between neurons. The brain is constantly developing in the process of people's learning with experience. As we live and gain new experiences, synaptic connections are updated, and new connections are formed when necessary. In this way, learning takes place. This situation is also valid for the Artificial Neural Network (ANN).

Learning takes place using examples that can be considered as the numerical equivalent of human experience by training the created artificial neural networks. In other words, the realization occurs by processing the input / output data, that is, by repeatedly adjusting the link weights using this data until the training algorithm converges.

The basic building block of artificial neural networks is the nerve cell. This cell consists of 5 Basic elements. These are inputs, weights, summation function, activation function and output. As seen in Figure 3.3, cells have multi-entry structures. Each of these inputs is multiplied by weights and transmitted to the addition function, then the information obtained is processed in the activation function and transmitted to the output of the neuron. The result obtained as a result of these processes creates the value of that neuron.

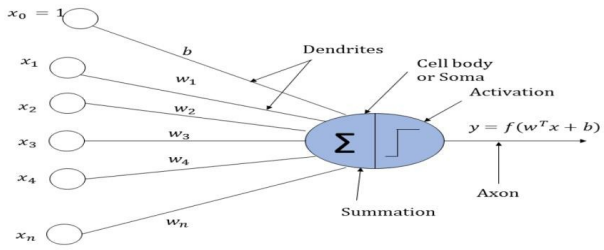


Figure 3.3: Example artificial neural network model [26][27][28].

And also, let's look at the structure of an artificial neural network, following diagram in Figure 3.4. Example artificial neural network model [26][27][28].

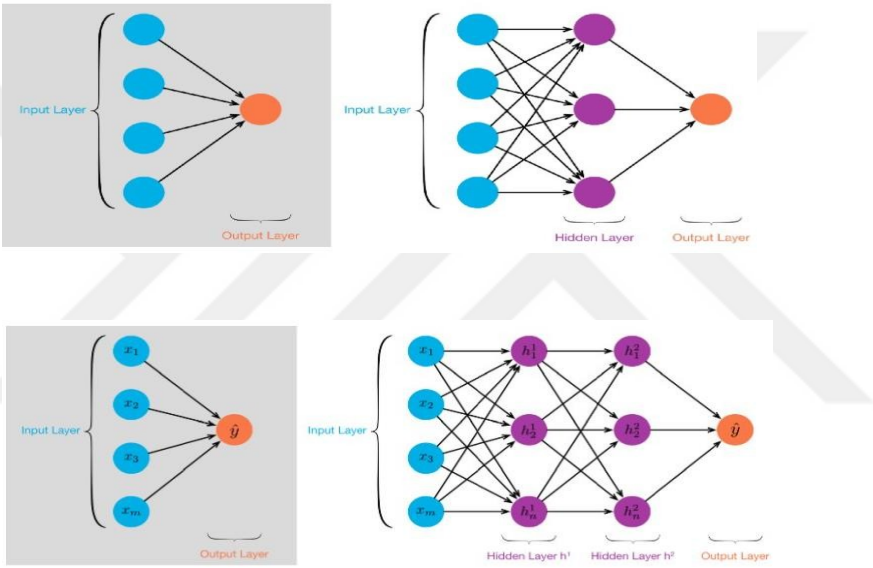


Figure 3.4: Example artificial neural network model [27][28][29].

The Inputs ($x1$, $x2$, …, xN) provides data input from the outside world. It is a dataset containing samples given to the network by the user. The weights ($w1$, $w2$, ..., wN) show how much the incoming input information for each neuron will affect the neuron's output. For example, the $w1$ weight shows the effect of input $x1$ on the output. The Summation Function is used to calculate the net input of a cell from all inputs. Various summation functions are used for this purpose. Most preferred is the weighted summation function. Other functions used are given in Table 3.1.

Table 3.1: Functions used in Artificial Neural Network (ANN) [27][28][29].

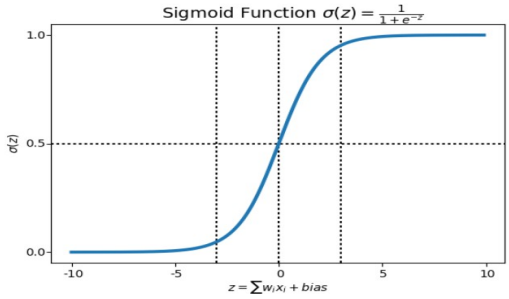
Function Name	Equation (i = 1,2, ..., n)
Weighted Total	$NET = \sum x_i w_i$
Multiplication	$NET = \prod x_i w_i$
Max	$NET = MAX(x_i, w_i)$
Min	$NET = MIN(x_i . w_i)$
Incremental Total	$NET = NET(k-1) + \sum x_i . w_i$

Table 3.2: Activation function used in ANN cells [28][29].

$$z = \sum_{i=1}^{m} w_i x_i + bias$$

Sigmoid Function is: $\sigma(z) = \frac{1}{1+e^{-z}}$

Sigmoid Function



The final output values of neurons are determined by the activation function. Table 3.2 gives the most preferred function among these functions is the sigmoid function. Output (y) is the value determined by the activation function. The output produced can be sent to another cell or to the outside world. In case of feedback, the cell can make feedback on itself and use its output value as input. However, feedback can also be made to another cell [28][29].

3.2 Learning In Artificial Neural Networks

3.2.1 Learning Strategies

In artificial neural networks, information is introduced by examples. These examples are given as input and output to the developed network structure.

The artificial neural network is aimed to learn the relational link between these inputs and outputs. After the learning takes place, the artificial neural network gains the experience to give outputs for different inputs. In other words, in line with the examples given, the artificial neural network learns the characteristics of the system.

It is not necessary to create samples by using input and output values together for each network. In some cases, learning of the network can be achieved by giving only inputs.[30] Depending on how these examples are created, artificial neural networks have 3 basic learning strategies [30].

Supervised Learning: As the name suggests, supervised learning takes place under the supervision of a teacher. Samples related to the event to be resolved are given to this system as an input / output set. With these examples, the system is aimed to solve the relationship between inputs and outputs. If there is a difference between the actual output and the desired/target output vector, an error signal is generated. On the basis of this error signal, the weights would be adjusted until the actual output is matched to the desired output [30][31] [32] [33].

Reinforcement Learning: In this strategy, a teacher helps the system. Input examples to the system are given. Then the system is expected to produce output for the given inputs, and a signal is sent to the artificial neural network, depending on whether the output is true or false. The system continues the learning process by taking this signal into account [30][31] [32] [33].

Unsupervised Learning: As the name suggests, this type of learning is done without the supervision of a teacher. Only input examples to the system are shown. The system is expected to learn the relationship, similarity and differences between the input parameters of the samples on its own. However, labeling is made by the user to make sense of the outputs. This strategy is often used in classification problems [30][31][32][33].

There are 4 basic learning rules in artificial neural networks. These are Hebb, Hopfield, Delta and Kohonen rules [32][33].

Hebb Rule: the Hebb rule forms the basis for all other rules. According to this rule, if the mathematical sign of two consecutive cells is the same, the connection of these two cells must be strengthened. The strengthening of the connection is made by increasing the weight value between the two cells.

If the mathematical signs of the cells are different, then the connections between cells are weakened, in other words, the weights are reduced.

Hopfield Rule: Unlike the Hebb rule, the Hopfield Rule is determined by the learning coefficient how much the connections of two cells will be strengthened or weakened. Learning coefficient is generally chosen between 0 and 1 by the user.

Delta Rule: In this rule, the weight values are changed in such a way that the squared average of the error between the output value produced by the artificial neural network and the target value desired to be produced is at least.

Kohonen Rule: According to the Kohonen Rule, cells in the artificial neural network compete to change their weight. The cell with the highest output value wins the race and the weight values are allowed to be changed.

3.3 Artificial Neural Network Architectures

3.3.1 Feed-Forward ANN

Feedforward ANNs allow signals to go only in one direction: from input to output. There is no feedback (loop); that is, the output of any one layer does not affect the same layer. Feed-forward ANNs tend to be simple networks that relate inputs to outputs. They are widely used in pattern recognition. This type of organization is also called bottom-up or top-down. Also, Feed-forward neural networks are ideally suitable for modeling relationships between a set of predictor or input variables and one or more response or output variables [33][34].

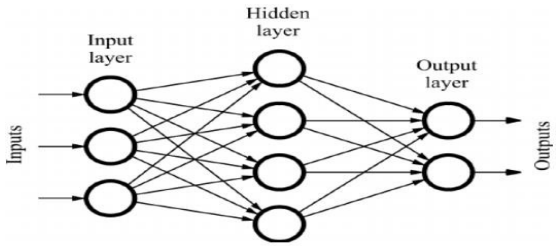


Figure 3.5: Feed-Forward ANN Model [33][34]

3.3.2 Feedback Artifical Neural Network

Feedback artificial neural networks, just like feed forward networks, consist of input, output and intermediate layers. However, in this structure, the signals obtained from the output layer and the outputs of the intermediate layers are fed back to the input units or to the previous intermediate layers [33][34].

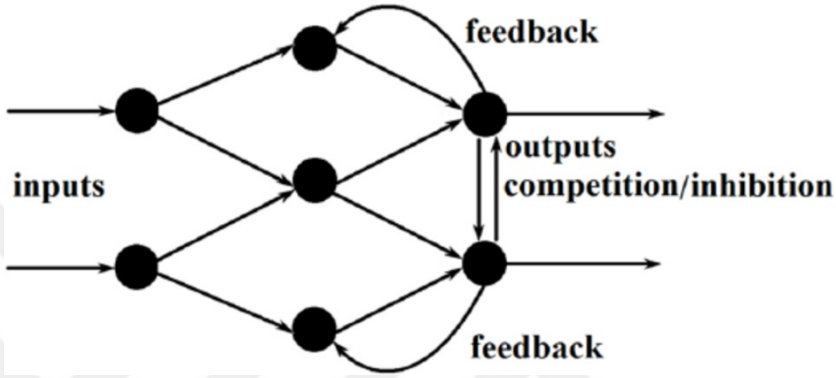


Figure 3.6: Feedback ANN Model [33][34]

3.4 Operation of Artificial Neural Networks

In artificial neural networks, information is given from the input layer to the network. This information link is strengthened by its weights, reaching the intermediate layers and processed from there to the output layer. The value of a neuron is found by multiplying the inputs from that neuron by coefficients and adding them. This result found is inserted into an activation function and According to the result of the function, it is also decided whether that neuron will be fired or not [35][36].

ANN operation steps:

1. Information is given to the network from the input layer.
2. With weighted connections, each information is processed in the neurons in thehidden layer and transmitted to the next layer.
3. It processes the information coming to each neuron input in the intermediatelayers by passing it through the gathering function.
4. The information coming to the output layer is passed out through the activationfunction.

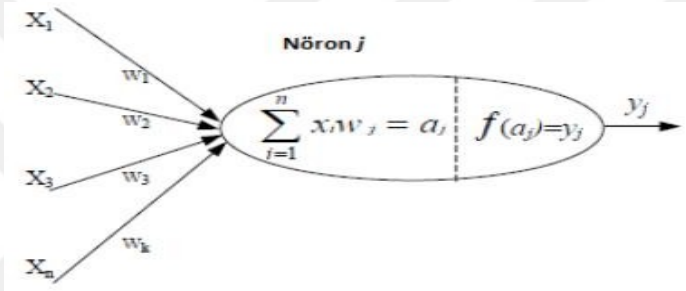


Figure 3.7: The Neuron Models [35][36].

For the neuron model given in Figure 3., all input information (x_i) is (w_i) multiplied by. We then obtain the sum of these products by the sum function:

$$\boldsymbol{a_j = \sum_{i=1}^{n} x_i w_{ij}} \quad (3.1)$$

The aj value that comes out of the addition function then passes through the threshold function, forming the output yi value of the neuron. Output value:

$$y_i = f(a_j) \quad (3.2)$$

3.5 Artifical Neural Networs Models

The number of layers, connection structures, activation functions used, different learning strategies and many neuron factors in ANN architectures have caused this architecture to be enriched with many different models. The multi-layer sensor model that forms the basis of our work is one of them.

Some of the most common ANN architectures are listed below:

- Perceptron
- Adaline
- With multi-layer sensor
- Radial based function
- Elman network
- Hopfieldnetwork.

3.5.1 Artificial Neural Network with Multi-Layer-Sensor

A multi-layer sensor (MCA) neural network model is shown in Figure 3.6. This model has become the most preferred artificial neural network model in engineering applications. The fact that many learning algorithms can also be used to train this network is one of the biggest factors in its widespread use. An MCA pattern consists of an input, an output, and one or more intermediate (hidden) layers. Processing elements (neurons) in each layer have a structure connected with each of the neurons in the next layer. The number of neurons in the input layer is determined depending on the number of data the applied problem will receive from the external environment. This information received from the input layer is processed in intermediate layers and transmitted to the output layer. In this way, the network is provided to give results to the external environment [35][36]. Examples of networks in multilayer networks and results associated with these examples are shown. The examples are applied to the input layer. Processed in intermediate layers, the sample results coming out are expected the error rate is determined by comparing with the results.

Until this ratio reaches the desired minimum value, the training continues by updating the weights with back propagation [35][36].

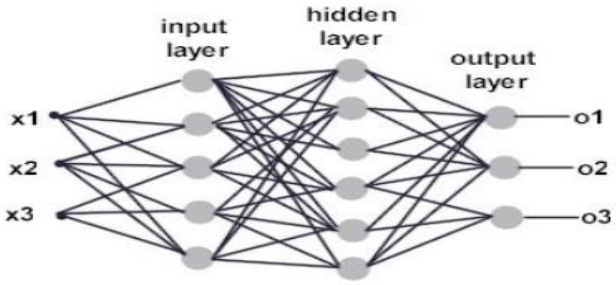


Figure 3.8: ANN Model With Multi-Layer-Sensor [35][36]

In multilayer networks, there are two different stages in solving a problem in artificial neural networks. The first one is Forward pass, the other is backward pass. When calculating in the backward pass direction, the connection weights of the neurons are updated.

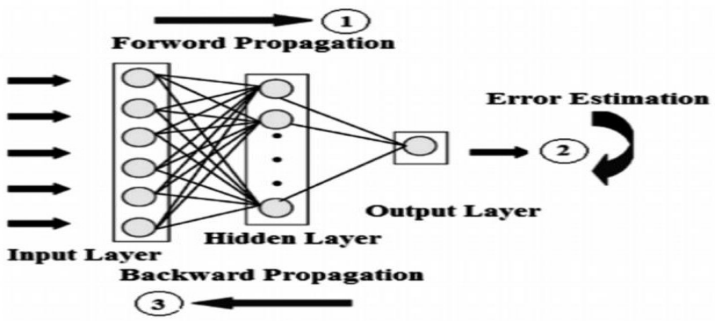


Figure 3.9: ANN Model With Multi-Layer-Sensor [35][36]

In the feed forward neural network, cells are arranged in layers and the outputsof cells in one layer are given as input to the next layer via weights. Feedforward ANN As the first step in the Advanced Computing phase, the samples prepared for training are given to the network through the input layer. These data given to the network are given by multiplying each neuron in the hidden layer by their connection weights.

The sum of these multiplications gives the NET input information of that neuron.

$$NETaj = \sum xi\ wi \quad (3.3)$$

$$Faj = f(NETaj) \quad (3.4)$$

The input NET obtained with the sum function in Equation 3.3, then given to theactivation function (Phage), the output value is calculated.

Feedback ANN While calculating in the Back Direction, the estimated outputsproduced by the network are compared with the targeted outputs. The difference givesthe error of the network. To reduce this error, the network performs back propagation. Back propagation progresses from the output layer to the input layer, updating all weights in order.

$$\varepsilon j = Tj - Fj \tag{3.5}$$

T_j: target output

F_j: the output generated by the network.

To eliminate the negative consequences of the error, the two sides of theequation are squared.

$$\Delta w_{aj,ci} = -\eta_{aj,ci} \frac{\partial \varepsilon_i^2}{\partial w_{aj,ci}} \tag{3.6}$$

In: equation.

η: learning coefficient,

ε_i: i is the error of the output (If the weights between hidden layers or between input layer and hidden layer are changing, ε indicates the total error).

3.6 Deep Learning

Deep learning is an artificial neural network-based machine learning technique that has attracted great interest from researchers in recent years. Both the performance of traditional machine learning methods and the capacity to analyze much larger data have made this approach indispensable in a short time.

Deep learning architectures such as deep belief networks, recurrent neural networks and convolutional neural networks are applied in many areas such as computer vision, natural language processing, speech recognition, and social network analysis.

With this technology, computer models are trained using data and neural network architectures containing many layers. As a result, computer models can learn and develop algorithms on their own.[37][38]

3.6.1 Recurrent neural network

Recurrent neural network (RNN) is a feedback ANN model. In RNN architecture, the output of at least one cell is given as input to itself or to othercells. This feedback can be between cells or between layers. The main purpose of recurrent neural networks is to use sequential information.[38][39]

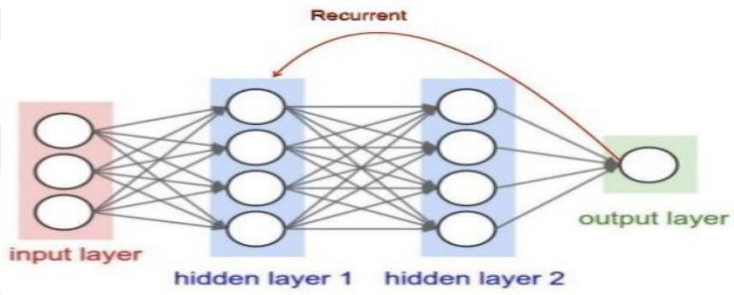


Figure 3.10: RNN model structure [39]

The way the RNN architecture works is in the following structure: it has an input layer, two hidden layers and an output layer. All of these layers work independently. Therefore, each layer has a specific function,and each layer performs a different function [38][39].

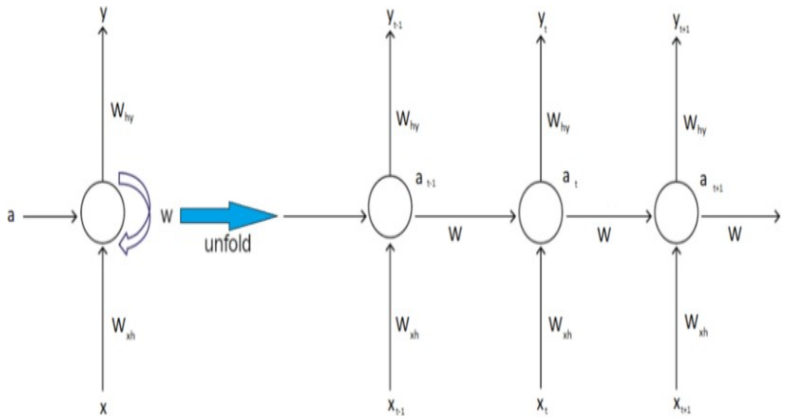


Figure 3.11: Recurrent Neural Networks (RNNs) Architecture [39]

W_{xh}: is weights for the connection of the input layer to the hidden layer.

W: is weights for the connection of the hidden layer to the hidden layer.

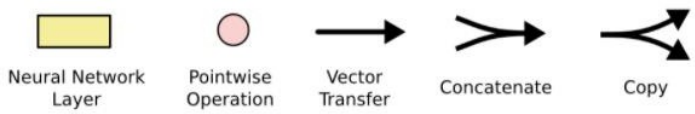
W_{hy}: are the weights for the connection of the hidden-layer-to-output-layer layer.

a: is the activation of the layer.

3.6.2 Long-Short Term Memory Networks

Long-Short Term Memory (LSTM) is a derivative of RNN that can learn longterm information and dependencies. When processing time series data, LSTMs can access information at different times. RNN and LSTM architectures shown together. In the LSTM structure, instead of a single network layer, there are 4 layers connectedto each other. With the help of gates using LSTM, it is decided what information the cell will store, and the time when operations such as reading, writing and deleting areperformed.

These gates contain a network structure in which the activation function isimplemented. These weights are calculated during the learning phase of the recurrent network. With the help of this network structure, the cell learns the processes of receiving, transmitting or deleting incoming data. So, LSTM algorithm is to decide which input data to pass through the cell [40][41].



Module is a standard RNN and consists of a single layer. LSTM preserves theerror value from different layers in back propagation in the neural network.

Thus, by providing a fixed error value after a certain number of steps, it ensures that the learningsteps of repetitive networks continue. It does this by opening a new layer between input and output [41][42].

Long-Short Term Memory Networks are explicitly designed to avoid the long-term dependency problem. Remembering information for a long time is practically their default behavior, not something they have difficulty learning. All recurring neuralnetworks have the form of a chain of repeating modules of the neural network. In standard RNNs, this repeating module will have a very simple structure like a single tanh layer [41][42].

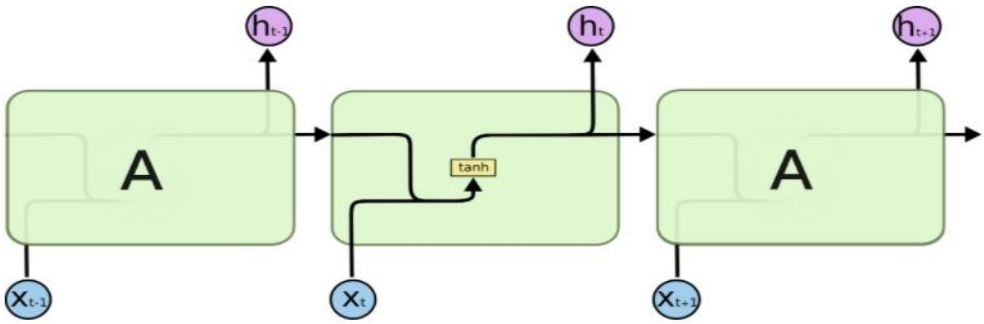


Figure 3.12: Standard RNN and Architecture [43][44]

In standard RNN architecture, there is a door in each cell. In LSTM, it has a more complex structure with four gates that ensure the continuity of information.

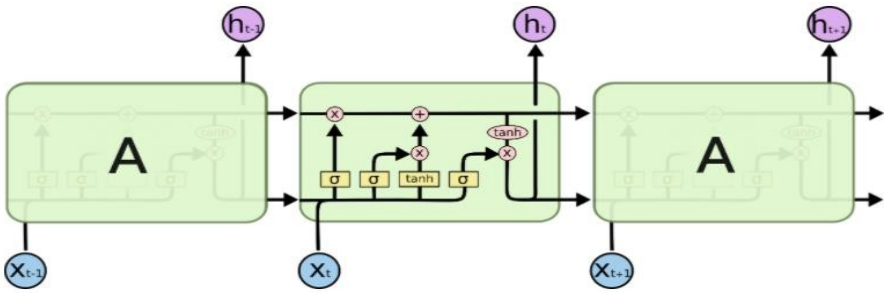


Figure 3.13 LSTM contains four interacting layers [43][44]

The key to the LSTMs is the cell state, which is the horizontal line running across the diagram.

The cell condition is like a kind of conveyor belt. It goes straight along the entire chain, with only some minor linear interactions. It is very easy for information to flow unchanged. There are two important components in an LSTM cell.

These are the Cell state and gates. Cell state Ct makes it possible for information to pass through the cell unchanged or with some small interaction.

Gates, on the other hand, protect the cell state and control this interaction as shown in the figure 3.14.

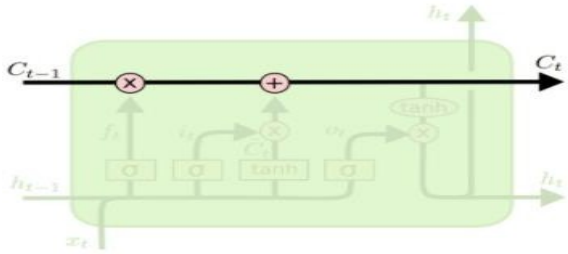
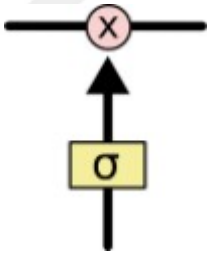


Figure 3.14 Cell status information and gate display structure [43][44]

LSTM has the ability to remove or add information to the cell state carefully regulated by structures called gates. Gates are an optional way of transmitting information. They consist of a sigmoid neural network layer and point multiplication.



Sigmoid layer gives the numbers between zero and one describing how much of each component should be allowed to pass. If the value of zero "don't let anything go through" while a value of one "let everything pass!". An LSTM has three of these gates to maintain and control cell status

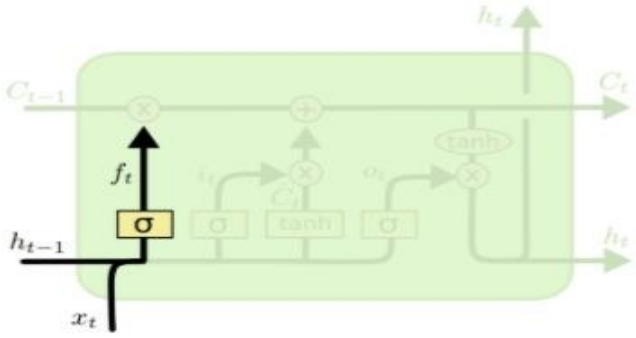


Figure 3.15 LSTM first gate [43][44].

$$S(Z) = \frac{1}{1+e^{-z}} \tag{3.7}$$

The first gate in the LSTM cell was the forget door, which consists of a sigmoid layer and decides what to keep from the previous $ct-1$ state. Where: $ht-1$ previous output value taking the input of xt with It gives the ft output a value between 0 and 1.

There is the mathematical model of the first gate. Where is the weight vectorof this gate, b_f is the bias value.

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_f\right) \tag{3.8}$$

The second gate consists of a combination of a sigmoid layer called the entrance layer and a tanh layer.

This gate decides what new information to add and update to the cell state. Sigmoid layer selects update values from the previous state. The next step decides which new information will be stored on the cell state. This step consists of two stages. In the first stage, a sigmoid layer, called the input layer, decideswhich values to update. Next, a tanh layer creates the new candidate values vector to be added to the cell.

$$i_t = \sigma\left(W_i . [h_{t-1}, x_t] + b_i\right) \tag{3.9}$$

(Note: The tanh layer creates the vector containing candidate values to be added.)

$$\hat{C}_t = tanh\,(W_c\,.[h_{t-1}, x_t] + b_c) \quad (3.10)$$

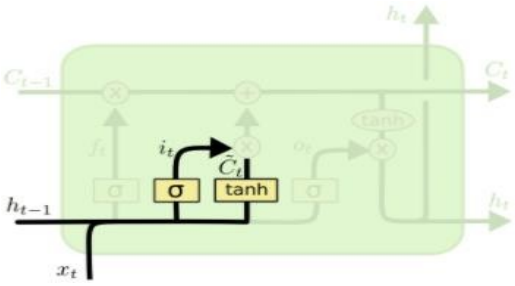


Figure 3.16 LSTM second gate [43][44].

Using the previous figures and definitions, the LSTM cell status is updated from "$ct-1$ " to "ct ". Lastly, we should have to decide what to output. This output will depend on our cell state, but it should be a filtered version. The scheme of the second gates in the LSTM cell is shown in Figure 3.13.

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (3.11)$$

This is where we will add new information or cell status as we decided in theprevious steps

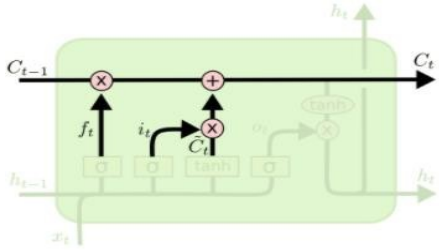


Figure 3.17 LSTM new cell status [43][44].

Next, multiply the cell state by tanh (to push the values between (- 1 and 1) the output of the sigmoid gate so we subtract only the parts we decided on.

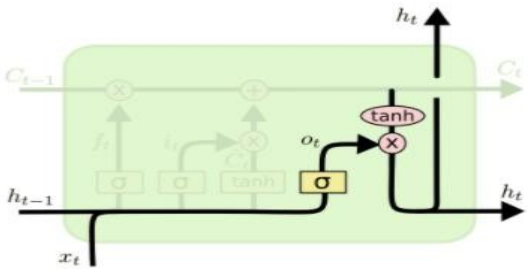


Figure 3.18 LSTM last gate [43][44].

In the last gate, a sigmoid layer is run that decides which parts of the cell state will be output. In this cell state, multiplied by the sigmoid gate output through the tanh layer.

$$\sigma_t = \sigma\left(W_{O.}\left[h_{t-1}, x_t\right] + b_O\right) \tag{3.12}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{3.13}$$

In Generally, an LSTM cell receives new information with information fromthe previous state. Also, it decides what information to store or discard for long-termmemory and learning takes place [47].

CHAPTER IV

METHODOLOGY AND MATERIALS

4.1 Dataset

For this study, we use the regional dataset showing the daily progress of the novel coronavirus outbreak in regions of Italy. Thus, I also use the publicly available Italian COVID-19 time series dataset provided by the Italian Ministry of Civil Protection. It can be downloaded from the website, built as a national response effort for the coronavirus emergency. The regional data set provides detailed figures for all 21 regions (19 regions and 2 autonomous provinces) as of 24 February and is updated daily [45][46][48][49].

The data set here has multiple characteristics (translated into English): Date, country, region code, region name, latitude, longitude, hospitalized with symptoms, intensive care patients, total hospitalized patients, home isolation, total positives (current positives), total positive, new positives, recovered (discharged), deceased, total cases, tests performed, total number of people tested, notes in Italian, notes in English.

We researched Italy data and worked, because different countries are testing COVID-19 very differently; We were hoping to partially control this by looking at a country. Our weather dataset comes from Dark Sky, collected in Google Map latitude and longitude pair assigned to each region in Italy, and our case numbers are taken from the Dati COVID-19 Italia GitHub repository reliable data it provides from Sito del Dipartimento della Protezione Civile- Emergenza Coronavirus: la risposta nazionale.The characteristics of the regions data set are as follows: Date, country, region name, total positives with symptoms (current positives) and new positives, total cases, death toll. Features "date, country, region code and name and etc." are contextual attributes, those are acceptable behavioral characteristics.

All these features were used during modeling, except for unnecessary and mostly empty features. The "tests performed." feature is a part of the government's response measures and varies significantly between regions depending on the policies implemented by each regional government in Italy.

The data set used in the study consists of 2 parts. In the first part of the data set; date, country, region name, number of positive with symptoms (current positives), number of recovery, number of deaths. The first part of the data set is shown in Table 4.1. This table is listed region name.

Table 4.1: A demonstration of COVID-19 daily raw dataset [48][49].

	REGION NAME								
	CAMPANIA			LAZIO			VENETO		
DATE	# OF POSITIVE	# OF Recovering Patients	# OF DEATHS	# OF POSITIVE	# OF Recovering Patients	# OF DEATHS	# OF POSITIVE	# OF Recovering Patients	# OF DEATHS
2/24/20	0	0	0	2	1	0	32	0	1
2/25/20	0	0	0	0	0	0	10	0	0
2/26/20	0	0	0	0	2	0	28	0	1
2/27/20	3	0	0	0	0	0	40	0	1
2/28/20	1	0	0	0	0	0	40	0	0
2/29/20	9	0	0	3	0	0	40	0	0
3/01/20	4	0	0	0	0	0	72	0	0
3/02/20	0	0	0	1	0	0	10	0	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
10/28/20	2427	365	17	1963	140	19	2143	49	11
10/29/20	3103	265	20	1995	214	15	2109	67	16
10/30/20	3186	525	15	2246	194	17	3012	74	17
10/31/20	3669	275	14	2289	108	22	2697	52	13
11/01/20	3860	409	3	2351	132	19	2300	48	17

Table 4.1: A demonstration of COVID-19 daily raw dataset [48][49] (continues).

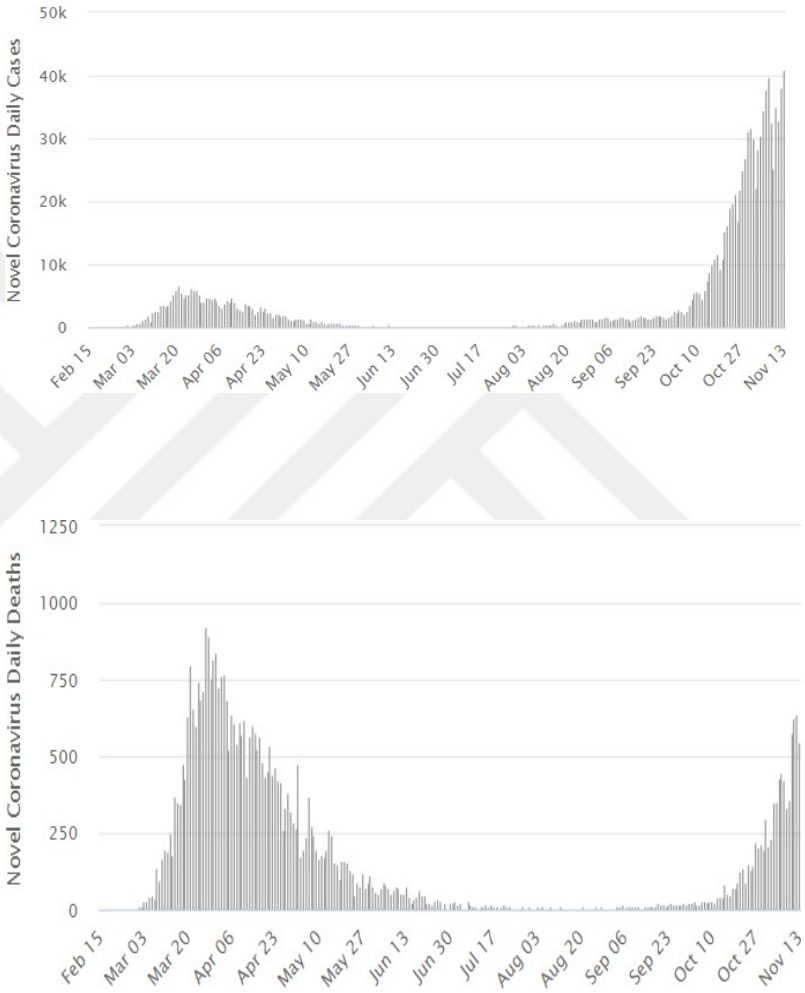
	PIEMONT			LOMBARDIA		
DATE	# OF POSITIVE	# OF Recovering Patients	# OF DEATHS	# OF POSITIVE	# OF Recovering Patients	# OF DEATHS
2/24/2020	3	0	0	166	0	6
2/25/2020	0	0	0	68	0	3
2/26/2020	0	0	0	18	0	0
2/27/2020	9	0	0	145	40	5
2/28/2020	11	0	0	128	0	3
2/29/2020	9	0	0	84	0	6
3/01/2020	38	0	0	369	33	1
3/02./020	2	0	0	270	66	14
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
10/28/2020	2547	261	19	7558	804	57
10/29/2020	2104	454	27	7339	1567	48
10/30/2020	2497	199	23	8960	2064	73
10/31/2020	2436	423	28	8919	489	54
11/01/2020	1590	423	11	8607	889	46

Table 4.2: A cross section of daily meteorological raw dataset [48][49].

REGION NAME									
	CAMPANIA			LAZIO			VENETO		
DATE	MAX	MIN	WIND	MAX	MIN	WIND	MAX	MIN	WIND
2/24/2020	10	4	5 km/h	15	12	8 km/h	12	8	10 km/h
2/25/2020	14	12	12km/h	16	12	15km/h	11	4	5km/h
2/26/2020	15	11	35km/h	14	9	35km/h	13	8	22km/h
2/27/2020	13	7	15km/h	12	5	12km/h	9	4	14km/h
2/28/2020	14	10	17km/h	14	9	17km/h	9	5	6km/h
2/29/2020	14	10	13km/h	13	8	21km/h	11	4	10km/h
3.01.2020	15	12	24km/h	15	9	26km/h	8	7	4km/h
3.02.2020	16	13	21km/h	15	10	36km/h	10	1	13km/h
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
8/28/2020	31	27	8km/h	30	26	17km/h	27	22	13km/h
8/29/2020	32	27	9km/h	32	23	29km/h	28	21	17km/h
8/30/2020	31	28	4km/h	29	25	24km/h	24	20	15km/h
8/31/2020	28	22	15km/h	23	13	10km/h	22	15	10km/h
9/01/2020	26	22	9km/h	27	16	12km/h	25	16	12km/h

	PIEMONT			LOMBARDIA		
DATE	MAX	MIN	WIND	MAX	MIN	WIND
2/24/2020	19	11	3 km/h	15	10	8 km/h
2/25/2020	13	8	4km/h	11	10	6km/h
2/26/2020	9	3	21km/h	11	7	32km/h
2/27/2020	9	2	12km/h	12	6	20km/h
2/28/2020	11	3	5km/h	13	6	11km/h
2/29/2020	9	4	6km/h	9	3	6km/h
3.01.2020	12	4	4km/h	9	7	5km/h
3.02.2020	10	5	5km/h	10	5	1km/h
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
8/28/2020	26	19	6km/h	27	22	10km/h
8/29/2020	27	18	5km/h	26	21	14km/h
8/30/2020	24	15	4km/h	25	19	9km/h
8/31/2020	25	14	6km/h	25	18	4km/h
9/01/2020	25	15	8km/h	26	18	8km/h

In the meteorology data (Table 4.2 Continues)**,** 5 major cities of the 5 major states that have approximately 75% of COVID-19 cases were selected. All data of these cities are usedas a separate feature in the LSTM network. The change of daily cases of the epidemic and the number of patients recovering, cases and death shown. In the graph, it is seen that the number of daily cases and death reached a peak in about a month with fluctuatio ns.



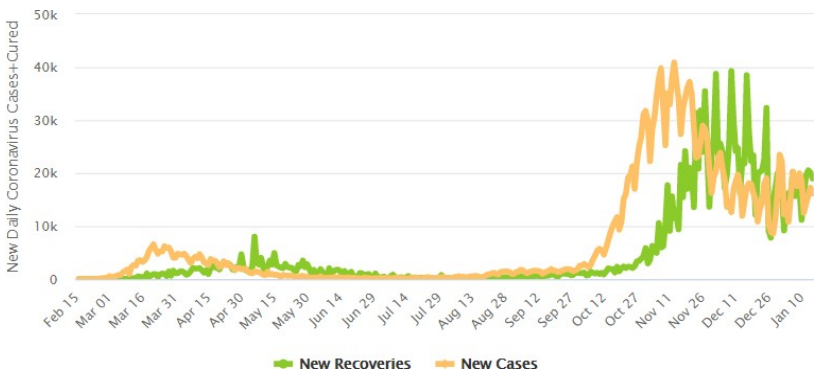


Figure 4.1: (Line 1) Number of daily cases, (Line 2) deaths and (Line 3) new caseand recovered [50].

The number of patients recovering, on the other hand, reaches its highest value rapidly from March and mid-April until the end of the month and the number of deaths started to decrease. There is a similar graph with the number of cases. In middle of the March and April, the death toll reached its maximum value and then decreased with a lower slope.

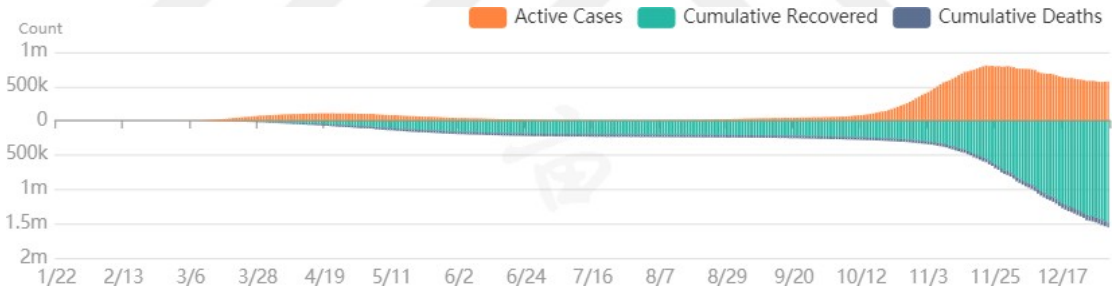


Figure 4.2: Number of active cases [50].

4.2 Dataset Pre-Processing Steps

The flow diagram of the experimental studies is given in Figure 4.5. The first step is to prepare the data. At this stage, first, all incomplete and repetitive records in the data sets are arranged and normalized before being trained with the LSTM network.

In the analysis of the results, firstly, the results are obtained with the COVID-19 data, which is the main part of the data set. Then, seasonal data is added to these results and how the forecast results of the LSTM network are observed and interpreted.

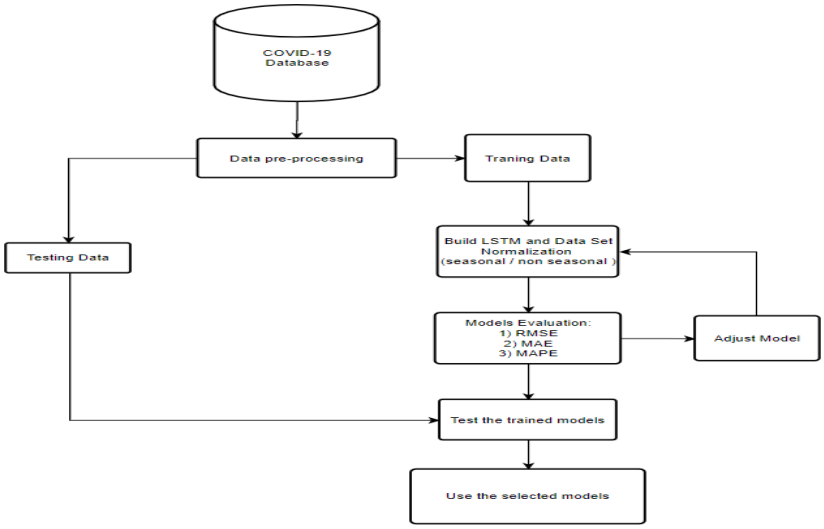


Figure 4.3: Conceptive structure of the proposed prediction methods.

Data pre-preprocessing: Before setting up the models, corrections on the data set, completing missing data, removing duplicate data, transforming, integrating, cleaning, normalizing, etc are transactions. At this stage, information discovery is unavoidably made on the data. Training data consist of 251 days. Next, we scale training data from zero to one to improve performance and training.

Moreover, each training example contains a string and a label for the actual value the model should estimate and generates real sequences for time series data to be fed into our LSTM model, which operates by inputting a series of numbers or vectors and outputs a number, as required in this model. We use it to create layers and initialize all the ancillary data. In Addition, to get the sequences, we pass all the sequences through the LSTM layer one at a time. The output from the last step is passed through the linear layer to get the estimate.

With the cross-validation method is a method that contributes to the optimization of the algorithm's runtime and results. The data is randomly divided into different parts. Each group is separated to be used as test data, and the remaining group is used as a training set.

In other words, the algorithm trains the data times and tests its results and the accuracy of the method applied is the average of the results of each test data.

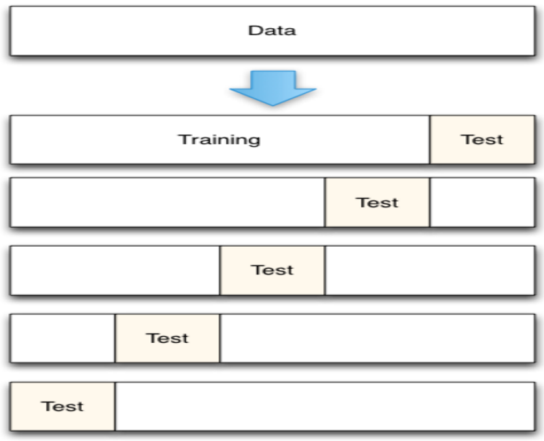


Figure 4.4: **Cross Validation Structure (5-Fold)**

On the other hand, unlike the cross-validation method, the data division method divides the data into two training and test sets with certain percentages. For example, as shown in Figure 4.7, the data is divided into two as 80% and 20%, and 80% of it is used as training and 20% of it is used as test data. The performance of the algorithm is determined as the performance criterion shown in the test data.

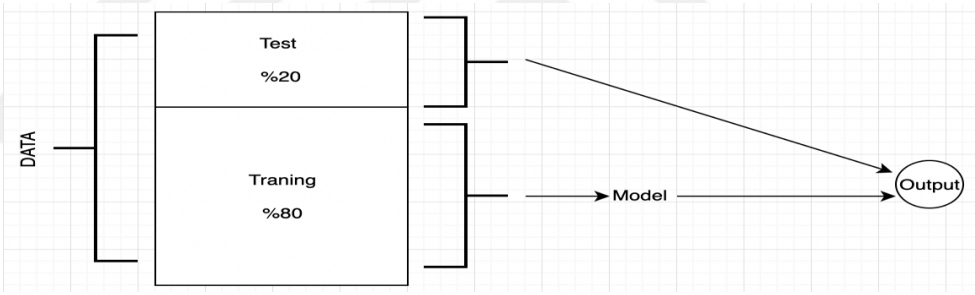


Figure 4.5: **Data Division Method Structure**.

$$Z = \frac{x_i - \mu}{\sigma} \qquad (4.1)$$

Where:

x_i : our dataset value,

μ: the average in the data set,

σ: standard deviation represents.

4.2.1 Experimental Study Results (Without Seasonal Data)

RMSE values, which are widely preferred and one of the most popular calculations in similar studies in the literature, are used as performance evaluation criteria in experimental studies.

In the experiments, the first part of the data is used for training and the last 7 days are estimated. The relationship between these estimates and actual values can be determined by RMSE values and graphics. To obtain the results, the optimum parameter values of the LSTM network are determined by trial-and-error method. All RMSE values shared in the graphs are presented by taking the average values obtained from ten consecutive runs of the LSTM network.

Table 4.3 Number of cases, deaths and recovered patients.

Date	# OF CASE	RESULT	# OF RECOVERED	RESULT
10/26/2020	1366	1461	314	617
10/27/2020	1409	1498	225	455
10/28/2020	1460	1646	348	458
10/29/2020	1444	1508	1322	1667
10/30/2020	1365	1428	312	766
10/31/2020	996	1101	883	632
11/01/2020	975	1079	291	251

Date	# OF DEATHS	RESULT
10/26/2020	13	11
10/27/2020	5	7
10/28/2020	9	8
10/29/2020	1	2
10/30/2020	4	6
10/31/2020	6	10
11/01/2020	8	7

The number of cases, deaths and recovered patients in the data set of the last seven days in Table 4.3 estimates are given. The average RMSE values of the results are presented in Table 4.4.

Table 4.4 Average RMSE values.

	# OF CASE	# OF DEATHS	# OF RECOVERED
RMSE	107.86764	2.10442	279.42569

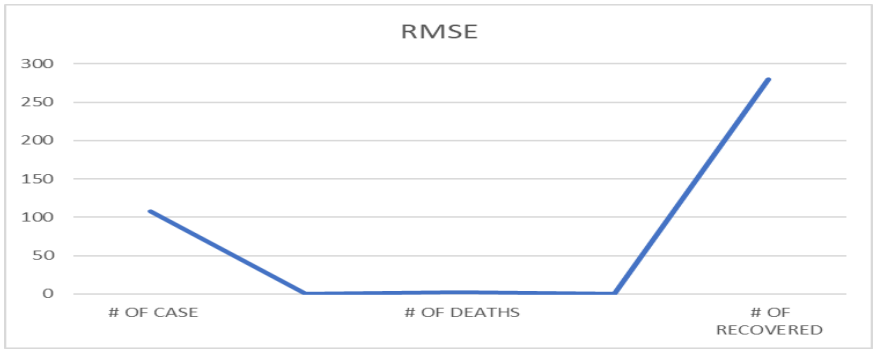


Figure 4.6 Average RMSE values.

The success of the LSTM method in predictions can be seen visually in the graph. There, the case and death rates are very close to the real values. Improved patient charts, on the other hand, show a lower proximity rate than case and death charts.

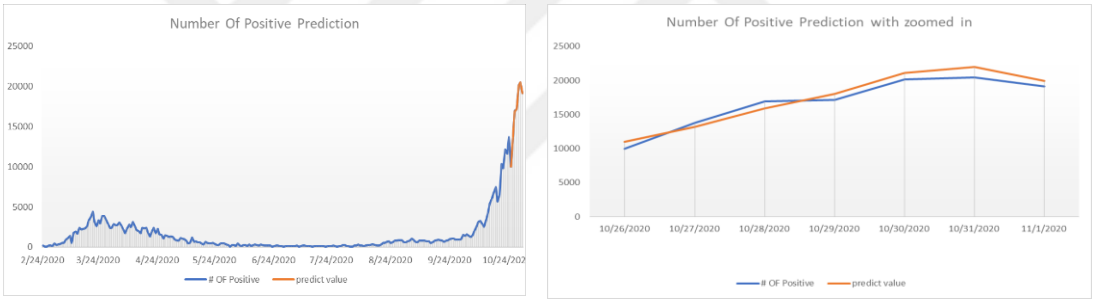


Figure 4.7: COVID-19 prediction graphs for the monthly and last 7 days for positive

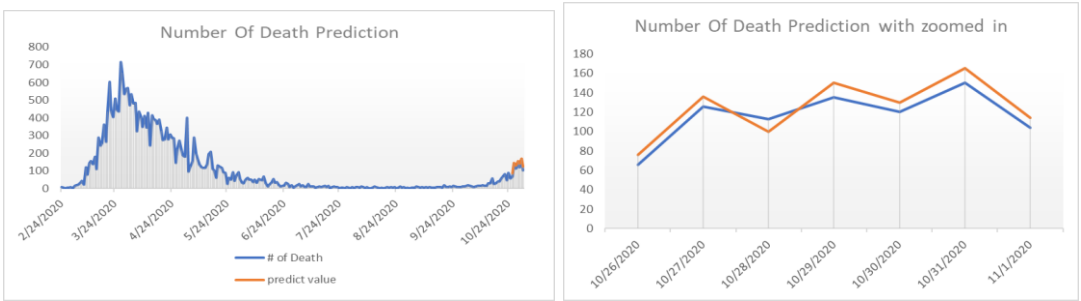


Figure 4.8: COVID-19 prediction graphs for the monthly and last 7 days fordeath

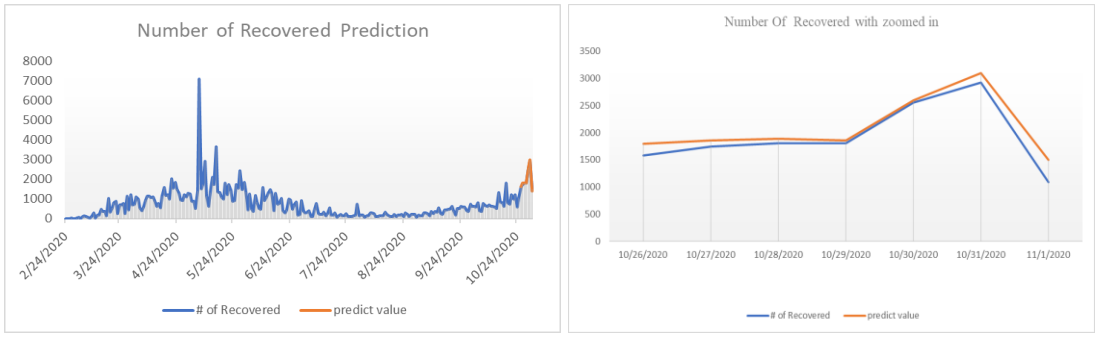


Figure 4.9: COVID-19 prediction graphs for the monthly and last 7 days forrecovered.

Corresponding the data graphics given in Figures (4.1) and (4.2) are examined, it is seen that the number of cases and deaths reached their highest values in mid-April and mid-May, then they started to decrease, and the number of patients who recovered similarly reached its peak at the end of April. Accordingly, in order to see the effect of the method in different time intervals, the data set will be divided into 4 parts and then the data will be partially calculated. For example: March- April, May- June, July-August, September- October and November.

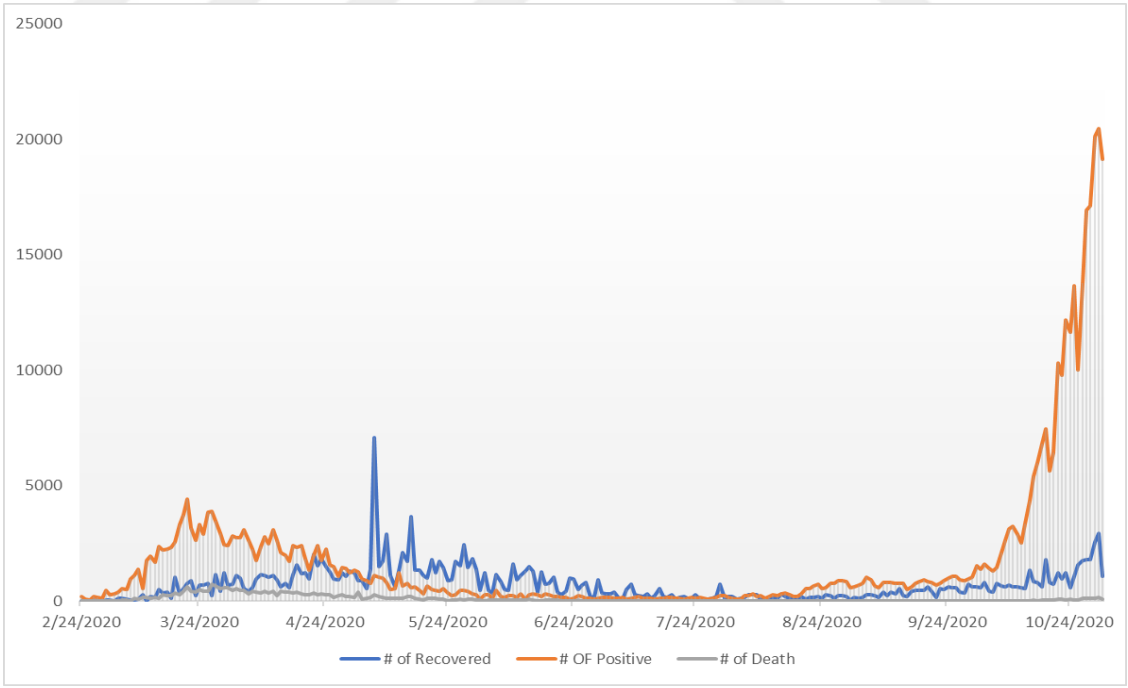


Figure 4.10: Graph over total data set.

RMSE results for each part of the data set are given in Table 4.5. When the results are examined, a much lower prediction success is observed in case and death estimation for the first part according to all data. This happens in the opposite direction with improving numbers. In the second part, closer RMSE values were obtained with all data in case and death values. The values of the first and second part in the number of improvements are close to each other. For the last section, there is an increase in all cases, albeit a little.

Table 4.5 RMSE results of the data set.

	# OF CASE	# OF DEATHS	# OF RECOVERED
RMSE	107.86	2.01	279.42
1st Section	459.25	7.66	112.94
2nd Section	379.51	7.31	137.97
3th Section	105.49	2.89	415.39
4th Section	393.25	4.31	497.97

When the results are examined, a much lower prediction success is observed in the case and death estimation for the first section (March and April) compared to the whole data.

In the numbers that have recovery, this situation occurs in the opposite direction. In fact, named as third section (July- Aug), closer RMSE values are obtained with all data in case and death values.

In the healing numbers, the values of one and second section are close to each other. Looking at the last section, an increase is seen for all cases, albeit less. It is possible to say that the LSTM network gives more stable results depending on the amount of data with the experiments performed on the partial data. In addition, the LSTM network estimates are adversely affected in the sections where the fluctuations in the data are high.

For example, the case and death rates in the first section peak very rapidly and fluctuating, and this situation negatively affects the results. Similarly, fluctuations are very high in the last section in the recovering graph, and in this case, the prediction success of the LSTM network decreases.

In order to increase the success here, determining the external factors mentioned in the conclusion section and teaching them to the network may contribute to getting positive results.

4.2.2 Experimental Study Results (Using Seasonal Data)

Analysis of the effect of seasonal data on forecasts is handled in two stages in the study. In the first stage, the effect rates of seasonal parameters on the result are determined and the parameters are selected. In the second stage, the effect of the data of the past days of the selected parameters on the cases was interpreted by experimental studies.

4.2.2.1 The Effect of Seasonal Parameters on the Prediction

Seasonal parameters in Table 4.6 are applied as input data to the LSTM network, which is used in the estimation studies made with the number of daily cases, deaths and recovering patients. Determination of the effect rates of seasonal parameters on the result and parameter selection are made.

Also, experimental studies interpret the effects of the data of the past days regarding the season's parameters on cases.

Seasonal parameters belonging to 5 cities, which are mentioned in the used data sets section and constitute over 70% of the cases, are respectively given to the LSTM network, and their effects are analyzed independently with experimental studies.

Table 4.6: Meteorological parameters table used in the study.

Parameter	Description	Measurement unit
Maximum air temperature	Average temperature value during the day.	°C
Minimum air temperature	The lowest temperature value recorded during the day.	°C
Wind speed	Average daily wind speed.	m / sn

When the results given for the situation estimation in Table 4.7 are examined, it is seen that only the maximum temperature values among the 3 different parameters have a positive effect on the result.

Other seasonal data negatively affect the result and increase the forecast error. Especially the wind speed parameter increases the obtained RMSE errors approximately 3-4 times. Although the minimum temperature increases the error rate, its effect is lower than the other parameters.

Table 4.7 The effect of seasonal parameters on the estimation of the number ofpositive

Parameter	RMSE
Number of Positive	107.86
Maximum Air Temperature	70.54
Minimum Air Temperature	128.06
Wind speed	400.16

RMSE values of these 3 parameters are given in Table 4.8 for the estimation of death numbers. Seasonal data do not appear to have as obvious an impact on death prediction as on case estimation.

Again, the maximum temperature value makes a contribution to improving the results, albeit a little. It is observed that other parameters increase the error rates.

Table 4.8: The effect of seasonal parameters on the estimation of the number of deaths.

Parameter	RMSE
Number of Deaths	2.10
Maximum Air Temperature	2.01
Minimum Air Temperature	3.12
Wind speed	4.52

When the results given in Table 4.9 are examined, it is seen that seasonal data do not have a positive effect on the estimation of the number of patients recovering, and all parameters increase the error rate.

Table 4.9: The effect of seasonal parameters on the estimation of the number of recovered.

Parameter	RMSE
Number OF Recovered	279.42
Maximum air temperature	489.88
Minimum air temperature	527.06
Wind speed	313.56

Given the results, it has been experimentally proven that those affected and killed by the COVID-19 outbreak are associated with the maximum temperature during the day. However, this situation has the opposite effect on the recovery of patients.

4.2.2.2 Effects Of Past Seasonal Data On Prediction

One of the most important factors in the COVID-19 outbreak is the incubation period. This period varies between 2 and 14 days depending on the person.

Based on this, it is understood that historical data are important when estimating the spread of the disease. In this part of our study, the effect of the maximum temperature parameter detected on the prediction of the values of the past 14 days is analyzed. In Table 4.10, RMSE values of the estimates of the number of cases obtained for these days are given.

Table 4.10 Maximum temperature effect for the past 14 days in the positiveprediction.

Past Day	1	2	3	4	5	6	7
RMSE	**60.93**	70.54	106.33	78.45	110.54	81.62	**58.22**
Past Day	8	9	10	11	12	13	14
RMSE	78.32	73.18	65.29	**59.54**	69.75	80.21	132.68

The results are examined, it is seen that the temperature values 3, 5 and 14 days ago have a negative effect on the case prediction. When the data of 7 and 11 days ago are examined, it is seen that an almost 50% reduction in case prediction error has been achieved.

In addition, experimental studies show that the temperature values 1 day before, which are not included in the incubation periods, achieve the highest prediction success. On other days, a positive effect is seen again. However, this effect does not change the result sharply compared to day 1, 7 and 11 data.

Table 4.11 Maximum temperature effect for the past 14 days for the deaths.

Past Day	1	2	3	4	5	6	7
RMSE	**1.05**	1.36	1.52	1.41	1.31	1.58	**1.24**
Past Day	8	9	10	11	12	13	14
RMSE	1.61	1.56	1.5	**1.04**	1.82	2.32	3.26

In Table 4.11, the effect of past temperature data in the estimation of the number of deaths is given. When the RMSE values are examined, it is seen that the temperature data of 6, 8, 12, 13 and 14 days ago have a negative effect on the results. In particular, the lowest RMSE values belong to 1, 7 and 11 days ago. Unlike the case prediction analysis, the effect of temperature values 11 days ago stands out with the RMSE value of 1.04.

The impact of historical temperature data on COVID-19 case and death rates for Italy is evident in this study. Especially the seasonal maximum temperature data of 1, 7 and 11 days before minimizes the error rates for the LSTM method. The case prediction graphics for these three days are given in Figure 4.11. When the zoomed graphs are examined, the prediction success interpreted with RMSE values is also presented here visually. The graphs of the LSTM network, which was trained with the temperature data of 1 day ago, present a much closer forecast graph compared to the others.

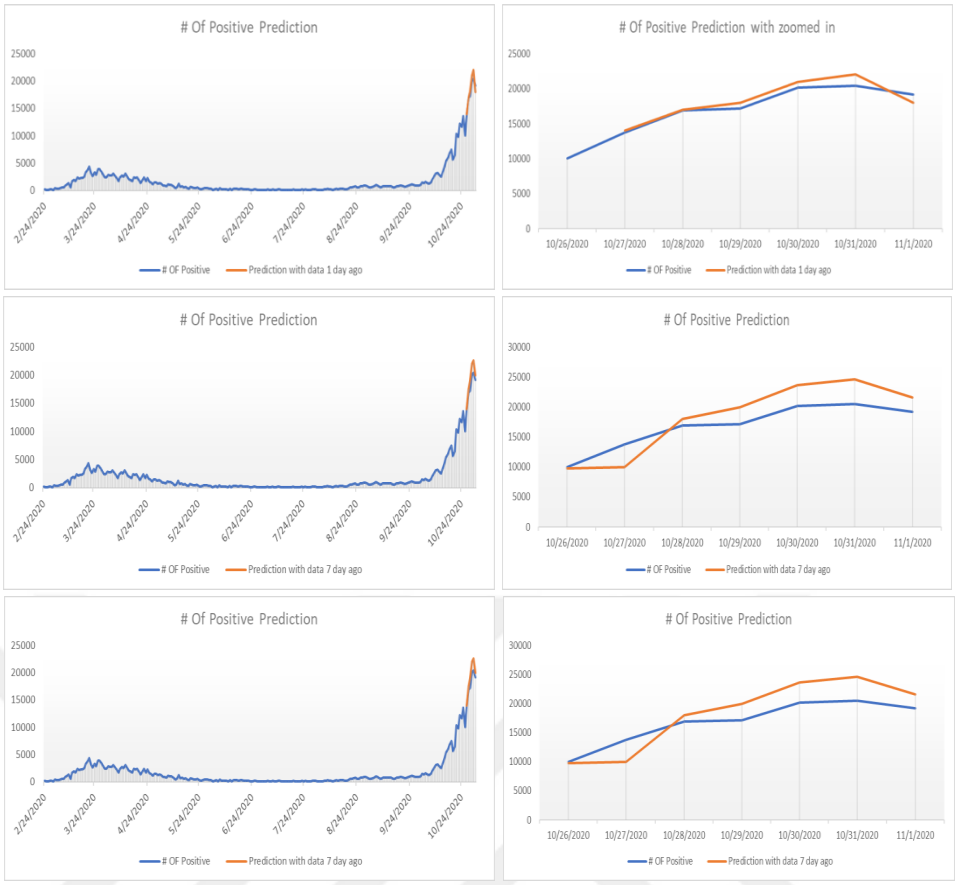


Figure 4.11: COVID-19 case prediction graphs for 1, 7 and 11 days of maximum temperature data and with zoomed in.

Figure 4.12 shows the measurement estimation graphs obtained by using the temperature data of 1, 7 and 14 days ago. When visual graphics are examined, closer graphics draw attention compared to the case estimate. On the other hand, since seasonal data do not have a positive effect on the number of patients recovering, the graphics of patients recovering are not shown here.

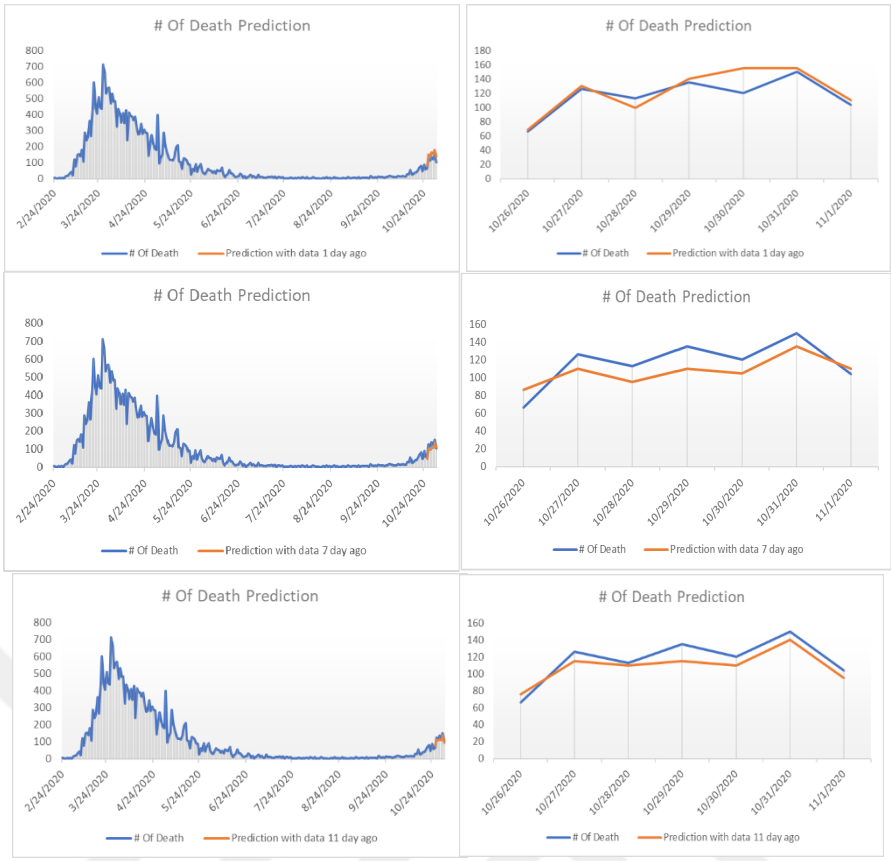


Figure 4.12: COVID-19 death prediction graphs for maximum temperature databefore 1, 7 and 11 days and with zoomed in.

In the finally, the results obtained from simulation studies for the analysis and evaluation of the proposed LSTM model for estimation of values will be discussed. In all models, values belonging to the last 7 days are used as input information.

Table 4.12: Performance comparison of all models.

Models		MAE	RMSE	MAPE (%)
Random Forest	Values	77.3458	89.3098	20.60
SVM	Values	70.5024	80.9322	16.53
KNN	Values	67.7939	75.2485	12,94
LSTM	Values	64.3920	70.5441	10.52

Firstly, it was compared with the prediction errors of other machine learning models to evaluate the prediction errors of the LSTM model. Also, the values change rates of the MAE, RMSE and MAPE (%) criteria used to evaluate the performance of the models are given. In order to compare the performance and accuracy of the results obtained from the proposed LSTM algorithm for the estimation of their values, some machine learning algorithms found in the literature were selected. The machine learning algorithms selected are as follows; K-Nearest Neighbor, Support Vector Machines and Random Forest algorithm.

When the results are examined in general, it has been shown that the LSTM model gives the best results in all performance criteria and generally performs better than the other results. Then, when the other performance criteria of the LSTM approach are examined, the MAE value is 64.3920, the MAPE value is 10.52% and the RMSE value is 70.5441. It was the 2nd KNN (K Nearest Neighbors) with the best results close to the LSTM model. So, It can be said that the accuracy of the number of cases produced for the coming days is more reliable than other models. Other hand, When seasons parameters data are added for the COVID-19 case prediction and results are examined, it is showed that it is not a suitable for the random forest algorithm. As seen in the graphs above Figure 4.13 and Figure 4.14, it gave results in the estimation error criteria MAE, MAPE and RMSE values. Actual values of 251 days of test data and LSTM predicted value were used in the graph. When the predicted values of the LSTM model are compared with the real values, it is seen that there is a consistency. It is also stated that the estimate and actual values show the same result.

CHAPTER V

CONCLUSION

Predicting COVID-19 cases is of paramount importance to the current situation. In this study, appropriate models were applied to real-time prediction of positive cases, deaths, and recovering COVID-19 cases in five different regions in Italy, confirmed daily. During the pandemic process, many studies have been done and continue to be done on modeling the spread, estimating the number of cases and resource planning. In case prediction machine learning studies, it will be inevitable to find different results when comparing various algorithms with each other, because many variables such as the data set and the parameters used by the algorithms can affect the results. In my study, by making case predictions (with seasonal parameter data) with machine learning algorithms, these algorithms were compared and the results were analyzed. One of the problems in the pandemic process is that the uncertainties are high and this process is dynamic.

Therefore, creating scenarios as close to reality as possible is of great importance in terms of public health. The most advanced machine learning algorithms and models in the literature were compared to evaluate and analyze the accuracy and performance of the results of the LSTM model, which is a deep learning approach. As a result of the studies, the predictive performance of the proposed LSTM model gave more effective values in all MAE, MAPE and RMSE performance criteria compared to traditional machine learning models.

Finally, seasonal data were combined with COVID-19 data, and the effect of some meteorological parameters on the epidemic was examined, and it was seen that the daily minimum and maximum temperature value contribute positively to the results.

An improvement is achieved by decreasing the average RMSE value of 107.86 obtained in the estimation of the number of cases to the level of 70.54. Considering

the incubation period of 2 to 14 days for COVID-19, the study also uses the past 14 days of maximum temperature values for outbreak prediction.

As a result of these experiments, it is seen that the temperature values found in the incubation periods affect the course of the outbreak. Especially in experimental studies, it is seen that the temperature data before 1, 7 and 11 days have the highest effect. In Addition, especially the effect of temperature values 1 day before, which are not within the incubation period, draws attention. The experiments conducted clearly reveal the impact of environmental factors in the COVID-19 outbreak.

In future studies, these data can be enriched with different factors such as age, gender, chronic disease status and covid-19 vaccine factor which are known to be medically related to the epidemic, and the prediction success can be increased. For the future, it may be suggested to optimize the parameters of the LSTM model using meta-heuristic algorithms.

REFERENCES

[1] Montanez, J. A. R., Fernandez, M. A. A., Arriaga, S. T., Arreguin, J. M. R., & Calderon, G. A. S. (2019). Evaluation of a Recurrent Neural Network LSTM for the Detection of Exceedances of Particles PM10. In 2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) (pp. 1-6). IEEE.

[2] Dhameliya, A., Deokar, J., Gonsalves, J., & Mathur, A. (2019). Prediction of Dengue using Recurrent Neural Network. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 926- 929). IEEE.

[3] Lofgren, E., Fefferman, N. H., Naumov, Y. N., Gorski, J., & Naumova, E. N. (2007). Influenza seasonality: underlying causes and modeling theories. Journal of virology, 81(11), 5429-5436.

[4] Srimath-Tirumula-Peddinti, R. C. P. K., Neelapu, N. R. R., & Sidagam, N. (2015). Association of climatic variability, vector population and malarial disease in district of Visakhapatnam, India: a modeling and prediction analysis. PLoS One, 10(6), e0128377.

[5] Bandyopadhyay, S. K., & Dutta, S. (2020). Machine learning approach for confirmation of covid-19 cases: Positive, negative, death and release. MedRxiv.

[6] Philemon, M. D., Ismail, Z., & Dare, J. (2019). A review of epidemic forecasting using artificial neural networks. International Journal of Epidemiologic Research, 6(3), 132-143.

[7] Pal, R., Sekh, A. A., Kar, S., & Prasad, D. K. (2020). Neural network-based country wise risk prediction of COVID-19. arXiv preprint arXiv:2004.00959.

[8] Sajadi, M. M., Habibzadeh, P., Vintzileos, A., Shokouhi, S., Miralles-Wilhelm, F., & Amoroso, A. (2020). Temperature and latitude analysis to predict potential spread and seasonality for COVID-19. Available at SSRN 3550308.

[9] Demongeot, J., Flet-Berliac, Y., & Seligmann, H. (2020). Temperature Decreases Spread Parameters of the New Covid-19 Case Dynamics. Biology, 9(5), 94.

[10] D Liang, Y. C., & Cuevas Juarez, J. R. (2016). A novel metaheuristic for continuous optimization problems: Virus optimization algorithm. Engineering Optimization, 48(1), 73-93.

[11] Martínez-Álvarez, F., Asencio-Cortés, G., Torres, J. F., Gutiérrez-Avilés, D., Melgar-García, L., Pérez-Chacón, R., ... & Troncoso, A. (2020). Coronavirus Optimization Algorithm: A bioinspired metaheuristic based on the COVID-19 propagation model. ArXiv preprint arXiv:2003.13633.

[12] Al-Qaness, M. A., Ewees, A. A., Fan, H., & Abd El Aziz, M. (2020). Optimization method for forecasting confirmed cases of COVID-19 in China. Journal of Clinical Medicine, 9(3), 674.

[13] Internet:"https://www.healthline.com/health/what-is-a-pandemic#pandemic-defined".

[14] Johnson, N. P., & Mueller, J. (2002). Updating the accounts: global mortality of the 1918-1920" Spanish" influenza pandemic. Bulletin of the History of Medicine, 105-115.

[15] Reid, A. H., & Taubenberger, J. K. (2003). The origin of the 1918 pandemic influenza virus: a continuing enigma. Journal of General Virology, 84(9), 2285-2292.

[16] Chen, Y., Liu, Q., & Guo, D. (2020). Emerging coronaviruses: genome structure, replication, and pathogenesis. Journal of medical virology, 92(4), 418-423

[17] Lessmann, N., Sánchez, C. I., Beenen, L., Boulogne, L. H., Brink, M., Calli, E., ... & van Ginneken, B. (2020). Automated assessment of CO-RADS and chest CT severity scores in patients with suspected COVID-19 using artificial intelligence. Radiology.

[18] Mias, G. (2018). Time series Analysis. In Mathematica for Bioinformatics (pp. 329- 373). Springer, Cham.

[19] Mussumeci, E. (2018). A machine learning approach to dengue forecasting: comparing LSTM, Random Forest and Lasso (Doctoral dissertation).

[20] Gurney, K. (1997). An introduction to neural networks. CRC press.

[21] Anderson, J. A. (1995). An introduction to neural networks. MIT press.

[22] Santanu P. (2019). “Intelligent Projects Using Python: 9 real-world AI projects leveraging machine learning and deep learning with TensorFlow”

[23] Internet:” https://medium.com/towards-artificial-intelligence/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6”

[24] Internet (2017):” https://towardsdatascience.com/multi-layer-neural-networks-with- sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f”

[25] Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy blog, 21, 23.

[26] Haykin, S. (2007). Neural networks: a comprehensive foundation. Prentice-Hall, Inc.

[27] Rosa, J. L. G. (Ed.). (2016). Artificial Neural Networks: Models and Applications. BoD–Books on Demand.

[28] Rojas, R. (2013). Neural networks: a systematic introduction. Springer Science & Business Media.

[29] Zaki, Peter W. (2019)."A Novel Sigmoid Function Approximation Suitable for Neural Networks on FPGA." 2019 15th International Computer Engineering Conference (ICENCO).

[30] Iliadis, L., & Jayne, C. (Eds.). (2015). Engineering Applications of Neural Networks: 16th International Conference, EANN 2015, Rhodes, Greece, September 25-28, 2015. Proceedings (Vol. 517). Springer.

[31] Hassoun, M. H. (1995). Fundamentals of artificial neural networks. MIT press.

[32] Suzuki, K. (Ed.). (2013). Artificial neural networks: architectures and applications. BoD–Books on Demand.

[33] Livingstone, D. J. (Ed.). (2008). Artificial neural networks: methods and applications (pp. 185-202). Totowa, NJ, USA: Humana Press

[34] Andersen, J. (2018). Theory, Operation, and Application of Neural Networks.

[35] Braspenning, P. J., Thuijsman, F., & Weijters, A. J. M. M. (1995). Artificial neural networks: an introduction to ANN theory and practice (Vol. 931). Springer Science & Business Media.

[36] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1, No. 2). Cambridge: MIT press.

[37] Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML workshop on unsupervised and transfer learning (pp. 37-49).
[38] Lazy Programmer (2016). Deep Learning: Recurrent Neural Networks in Python: LSTM, GRU, and more RNN machine learning architectures in Python
[39] Brownlee, J. (2017). A Gentle Introduction to Long Short-Term Memory Networks by the Experts. Machine Learning Mastery, 19.
[40] Graves, A. (2012). Supervised sequence labelling. In Supervised sequence labelling with recurrent neural networks (pp. 5-13). Springer, Berlin, Heidelberg.
[41] Brownlee, J. (2017). Long Short-term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning. Machine Learning Mastery.
[42] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780. Olah, Christopher (2015). "Understanding lstm networks."
[43] Olah, Christopher (2015). "Understanding lstm networks."
[44] Internet (2015): http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[45] Internet: The COVID-19 Data Italy Website by Italian Department of Civil Protection, Available: " https://github.com/pcm-dpc/COVID-19"
[46] Internet: "https://darksky.net/dev"
[47] Internet: "https://www.kaggle.com/virosky/italy-covid19"
[48] Internet: "https://power.larc.nasa.gov/data-access-viewer/
[50] Internet: "https://w2.weather.gov/climate/"
[51] Internet: "https://coronavirus.1point3acres.com/data"