SOLAR RADIATION FORECASTING BY USING DEEP NEURAL NETWORKS IN ESKIŞEHIR

Mohammed Qasem Mohammed Saleh QASEM

Master's Thesis

Department of Advanced Technologies

Programme in Energy Resources and Management

Supervisor: Assoc. Prof. Dr. Ümmühan BAŞARAN FİLİK

Eskişehir

Anadolu University

Graduate School of Sciences

December 2019

FINAL APPROVAL FOR THESIS

This thesis titled SOLAR RADIATION FORECASTING BY USING DEEP NEURAL NETWORKS IN ESKIŞEHIR has been prepared and submitted by Mohammed Qasem Mohammed Saleh QASEM in partial fulfillment of the requirements in "Anadolu University Directive on Graduate Education and Examination" for the Degree of Master's in Advanced Technologies Department has been examined and approved on 26/12/2019.

Committee Members		Title, Name and Surname	Signature
Member (Supervisor)	:	Assoc. Prof. Dr. Ümmühan BAŞARAN FİLİK	
Member	:	Assist. Prof. Dr. Burak URAZEL	
Member	:	Assist. Prof. Dr. Cahit PERKGÖZ	

Prof. Dr. Murat TANIŞLI

Director of Institute of Graduate Programs

ABSTRACT

SOLAR RADIATION FORECASTING BY USING DEEP NEURAL NETWORKS IN ESKIŞEHIR

Mohammed Qasem Mohammed Saleh QASEM

Department of Advanced Technologies

Programme in Energy Resources and Management

Anadolu University, Graduate School of Sciences, December 2019

Supervisor: Assoc. Prof. Dr. Ümmühan BAŞARAN FİLİK

According to the WEO, the global demand for energy is presumably going to increase, due to growing up the world's population during the upcoming two decades. As a result of that, apprehensions about environmental effects, which appear as a result of greenhouse gases, are grown and cleaner energy technologies are developed. This clearly shows that extended growth of the worldwide market share of solar energy. For this reason, a predictive model that effectively observes solar energy conversion and its performance is considered as the best choice for design solar energy plants.

In this thesis, to find the most accurate model for solar radiation forecasting, many studies have been conducted. In this work, based on three different approaches, the problem was solved by using classic mathematical models, artificial neural network (ANN), and deep neural network (DNN) methods. Besides, four new models have been developed for solar radiation prediction. The data are obtained from the Turkish State Meteorological of Service. The obtained results are compared and evaluated by the root mean square error (RMSE), mean absolute error (MAE), mean bias error (MBE) and test statistics (t-statistic). As a result of the analysis, the best result was given by the deep learning (DL) method.

Keywords: Solar radiation forecasting, ANN, DNN, Artificial intelligence, Hybrid models.

ÖZET

DERIN SINIR AĞLARI KULLANARAK ESKIŞEHIR'DEKİ GÜNEŞ RADYASYONU TAHMINI

Mohammed Qasem Mohammed Saleh QASEM

İleri Teknolojiler Anabilim Dalı

Enerji Kaynakları ve Yönetimi Bilim Dalı

Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Aralık 2019

Danışman: Doç. Dr. Ümmühan BAŞARAN FİLİK

WEO'ya göre, küresel enerji talebinin önümüzdeki yirmi yılda dünya nüfusunun artması nedeniyle 2016'da% 30'dan 2040'ta% 40'a çıkacağı tahmin ediliyor. Bunun bir sonucu olarak, sera gazlarının neden olduğu çevre kirliliği ile ilgili endişeler artmakta ve daha temiz enerji teknolojileri geliştirilmektedir. Bu dünya çapında güneş enerjisi pazar payının genişlemiş büyümesini açıkça göstermektedir. Bu nedenle, güneş enerjisi dönüşümünü etkili bir şekilde gözlemleyen ve performansını gözlemleyen neredeyse gerçek zamanlı bir öngörü modeli, güneş enerjisi santrallerinin tasarımı için en iyi seçenek olarak kabul edilir.

Bu çalışmada, güneş ışınımı tahmini için en uygun modeli bulma üzerinde çalışmalar yurütülmüştür. Bu kapsamda üç farklı yaklaşm temel alnarak problem matematiksel modeller, yapay sinir ağları ve derin öğrenme kullanılarak çözülmüştür. Ayırca, güneş ışınımı tahmini için dört yeni model oluşturulmuştur. Bütün bu yöntemlerin uygulamasında Eskişehir'deki meteroloji istasyondan alınan veriler kullanılmıştır. Elde edilen sonuçlar Kare Ortalamalarının Karekökü (RMSE), ortalama mutlak hata (MAE), ortalama önyargı hatası (MBE) ve test istatistiği (t-istatistiği) değerlerine kullanılarak detaylı bir şekilde karşılaştırılmıştır. Yapılan analiz sonucunda ele alan bölge için en iyi sonucu derin öğrenme yöntemi vermiştir.

Anahtar Sözcükler: Güneş radyasyonu tahmini, ANN'ler, DNN'ler, Yapay zeka, Hibrit modeller.

ACKNOWLEDGEMENTS

I would like to thank Assoc. Prof. Dr. Ümmühan BAŞARAN FİLİK for her expert advice and encouragement, as well as my parents for their help and support for all my life. I am grateful to all of those whom I have had pleasure to work with during my master program. This work would not be possible if I had not had their support and inspiration.

Mohammed Qasem Mohammed Saleh QASEM

26/12/2019

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with “scientific plagiarism detection program” used by Anadolu University, and that “it does not have any plagiarism” whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Mohammed Qasem Mohammed Saleh QASEM

CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

φ	:	AR parameter
Θ	:	MA parameter
$I(t)$	:	Solar radiation in any time
ε_t	:	White noise
δ	:	Solar declination angle
Γ	:	Day angle
ϕ	:	Latitude (degree).
D	:	day number starting from the first of January
I	:	Hourly global solar radiation on horizontal
H	:	DGSR on horizontal surface
I/H	:	Ratio of hourly to DGSR on horizontal surface
k_t	:	Hourly clearness index
K_t	:	Daily clearness index
S_0	:	Day length
t_s	:	Solar time
W	:	Solar hour angle
W_s	:	Sunrise hour angle
ACF	:	Auto-Correlation Function
AI	:	Artificial Intelligence
AR	:	Autoregressive
ARIMA	:	Autoregressive and Integrated Moving Average
ARMA	:	Autoregressive and Moving Average
ANN	:	Artificial Neural Network
BP	:	Backpropagation
BPTT	:	Backpropagation Through Time
CNN	:	Convolutional Neural Network
DGSR	:	Daily Global Solar Radiation
DL		Deep Learning
DNN	:	Deep Neural Network
DRNN	:	Deep Recurrent Neural Network

ENN	:	Elman Neural Network
FL	:	Fuzzy Logic
FNN	:	Fuzzy Neural Network
FFNN		Feedforward Neural Network
GAs	:	Genetic Algorithms
GFS	:	Global Forecast System
IFS	:	Integrated Forecast System
IoT	:	Internet of Things
IT	:	Information Technology
LSTM	:	Long-Term and Short-Term Memory
MA	:	Moving Average
MAE	:	Mean Absolute Error
MBE	:	Mean Bias Error
ML	:	Machine Learning
MLPs	:	Multilayer Perceptrons
MLR		Multiple Linear Regression
NRMSE	:	Normalized Root Mean Square Error
NWP	:	Numerical Weather Prediction
RBFN	:	Radial Basis Function Network
ReLU	:	Rectified Linear Unit
RMSE	:	Root Mean Square Error
RNN	:	Recurrent Neural Network
R&D		Research and Development
SOMs	:	Self-Organizing Maps
SPSS	:	Statistical Package for Social Science
SVM	:	Support Vector Machine
SVR	:	Support Vector Regression
TDNN	:	Time Delay Neural Network
TSI	:	Total Sky Imager
WNN	:	Wavelet Neural Network
WSI	:	Whole Sky Imager
WT	:	Wavelet Transform

1. INTRODUCTION

As a result of the arising on traditional fuel prices and the increase in energy production from alternative sources, the integration of renewable energy plants into the electric grid becomes urgent and must be encouraged. Besides that, alternative energy sources have many advantages, they also have the lowest effects on the environment 'environment-friendly', and uninterrupted (Alzahrani et al., 2017). The output power of renewable energy resources is variable and change in non-stationary time series because of that renewable energy resources are interrupted (Alzahrani et al., 2017). As it is known, solar energy is considered as one of the important alternative energy resources that can convert sunlight to electricity. The energy that can be generated by the solar panel is strongly depended on many factors like solar radiation, weather temperature, and other weather variables.

Keeping global energy supply safe with high integration of renewable energy sources is considered one of the serious challenges that can be faced soon (Alzahrani et al., 2017; Benali et al., 2019). The supply and demand of energy play an important role in the global development in everywhere of human activity. Adequate supplies of clean energy are strongly connected with global stability, economic prosperity, and quality of life. The integration of solar energy into electric grids has come as a result of the increase in demand for energy. To make using of solar energy more efficient, forecasting information must be trusted. The exact prediction of solar radiation variety can increase the quality of solar energy using and management (Chaturvedi, 2016). The effective forecasting is considered as one of the important factors that must be taken into consideration in solar power plant design (Sun, 2018). Prediction can be helpful for the development of different applications for power systems (Alzahrani et al., 2017; Benali et al., 2019).

1.1. Problem Statement

Nowadays, to make solar radiation forecasting more accurate, there are many methods are used. In the type of classic mathematical methods, there are autoregressive (AR) models, autoregressive and moving average (ARMA) model, …, etc. In the type of artificial intelligence (AI) techniques, there are different types of ANNs methods. Also, there are some hybrid models that are considered as a combination of two different models to get better results. Some of these methods may not be recommended because of their high rate error. For instance, the accuracy of solar radiation forecasting ranges between 85 - 91% by using classical mathematical methods. For this reason, we need to create new models by using new methods in order to get results with high accuracy. One of the ANNs methods which are called DNNs can be used to get results with high accuracy. For this reason, the main purpose of writing this thesis is to create an ideal model, that can be used to make solar radiation prediction simpler and within a reasonable accuracy.

1.2. The Purpose of This Study

Solar radiation can be affected by many weather factors, such as cloud cover, precipitation, temperature, wind speed, pressure, humidity. Nevertheless, the precise correlation between each weather feature and the received solar radiation can be difficult to be found. The main goal of this research is to create a direct relationship between the weather forecasting data and solar energy by using DNNs. DNNs can be used to find linear as well as the non-linear relationship between features and targets without much of feature engineering or domain knowledge. We are planning to use a fully connected DNN to find the direct relationship between daily global solar radiation (DGSR) with other weather variables.

1.3. Definition of Terms

Solar radiation

It is the radiant energy released by the sun in all directions, from this radiation all planets and moons located in the solar system derive their heat. This energy is so huge, it is estimated $384.6 \cdot 10^{24}$ watts/m^2, but approximately only 1,360 watts/m^2 enter the earth’s atmosphere. This, a tiny amount of energy is responsible for all the thermal energy of the earth's surface and atmosphere, which usually meant when talking on solar radiation as an important variable of climate elements, called the insulation (http-5).

ANNs

In the information technology (IT) field, ANN is a system of interconnected processing units that are designed to simulate the neurons and neural network of the human brain. The neural networks typically contain a large number of processors running in parallel and arranged in levels, in other words, ANNs generally consist of three main layers; input, hidden and output layers. The first layer received data works in the same way that the optic nerves in the human visual system. Each successive layer receives the outputs from the previous layer, and the last layer gives system processing outputs. Each layer consists of a number of nodes, each node applies a function that processes the inputs with their weights, and the layers are highly interconnected, which means that each node in layer n will be connected to many nodes in layer n-1. Neural networks are known to be adaptable, which means that they can adjust themselves as they learned from initial training, and later provide more accurate information (Chau, 2006).

There are three moments in the limelight of ANNs. The first one is at the end of the fifth decades of the twentieth century 1950s. The second moment has started in the 1990s. In this period ANNs received a large interest that led to a well understanding of neural networks. In this period ANNs is started to be used for solving classification and regression problems. The third important moment for ANNs has been becoming with DNNs. While DNNs can be defined as ANNs with three or more hidden layers. For that reason, DNNs are also known as a simple enlargement of ANNs (Díaz et al., 2017).

DNNs

DNNs are known as ANNs with a number of hidden layers (at least three). DNNs are also considered as a simple extension of ANNs. DNNs give a perfect result in image identification, speech recognition competitions, translations, and classification. They are considered among the building blocks that aim to achieve AI which is still elusive (Grekousis, 2019).

1.4. Research Framework

This work aims to improve the solar radiation forecasting by using DNNs methods, to make forecasting data more accurate and has high efficiency. Solar radiation forecasting result obtained by using the DNNs method is more accurate than the result obtained by using classical and ANNs methods.

This work consists of five main chapters. In the first chapter, an introduction to the components and the main parts of this work is given. In the second chapter, the most common methods and models that are used for solar radiation forecasting have been reviewed from more than twenty different resources. In the third chapter of this thesis, the content is divided into four main parts. In the first part, the forecasting of solar radiation is realized by using classic mathematical models, such as CPR, CPRG, Liu and Jordan model, Jain Model 1 and Jain model 2.

The second part of this chapter focuses on solar radiation prediction by using one method of ANN's methods. The ANN model, which is used in this work, is created by Visual Gene Developer Software which is used for solving ANN Multiple Linear Regression (MLR) problems. In the third part of this chapter, the solar radiation prediction model by using the DNN method is achieved. In the last part of the third chapter, four new models are created by one of the best software packages used for statistical analysis; the name of this software comes from the first letters of the Statistical Package for the Social Sciences (SPSS). In the fourth chapter of this work, the results are shown and discussed.

2. LITERATURE REVIEW

In this section, the importance of solar radiation prediction is explained. Also, the most common methods and models that are used for solar radiation forecasting are reviewed comprehensively.

2.1. The Importance of Solar Radiation Prediction

Solar radiation prediction has the biggest role in modeling solar systems with a higher percentage of accuracy. Figure 2.1 demonstrates some applications which use prediction results to upgrade works and future plans of the electric grid with the conforming required time-resolution of the forecasting (Voyant et al., 2017). Furthermore, solar radiation prediction about the upcoming second is important to operate network stability and grid voltage regulation optimally. Power reserve management, dispatching and load flowing need information about the next hours of solar irradiance. Similarly, transmission scheduling and unit commitment require data about the upcoming days of solar irradiation. And also, optimization, production, and consumption electrical energy need predicted data about the upcoming days, weeks or months of solar irradiation. And finally, Capacity and global management need solar radiation predicted data of the upcoming years. Solar radiation forecasting is necessary for many applications. In Figure 2.1, the importance of solar radiation forecasting with different time intervals is shown.

Figure 2.1. *Different applications use forecasting data with different time intervals (Ssekulima et al., 2016)*

Because of the discontinuous and unpredictable character of renewable energy, the integration of them into the electrical grid increases the network management complexity and the balance continuity between production and consumption. To avoid damage in the electrical loads they must be powered by constant balanced electricity all the time. However, the electrical generator and operator has some difficulties to keep the balance constant with traditional and energy production systems that are able to be controlled. The reliability of the system depends on how can the system is being adaptive to expected and unforeseen changes in production, disturbances, and consumption of electrical energy. In addition, to keep the quality of service at a good level, the energy provider should manage the system with different temporary horizons.

Grid voltage, power quality, and grid stability problems can be related to uncontrollable solar energy production. Therefore, day-ahead forecasting the output power of the solar systems is paramount for:

- grid operators to execute tasks such as scheduling, optimum management of the energy generated from solar panels;
- unit commitment;
- load flowing;
- optimum management of energy storage;
- trading renewable energies in the energy markets and to keep energy costs down, assessment the amount of the reserves, scheduling the electrical grid;
- overcrowding management (Fouilloy et al., 2018; Ssekulima et al., 2016).

As a result of increasing energy production from solar panels, forecasting solar radiation becomes necessary, to avoid large variations and differences in grid voltage. Also, it is important for whole the system including storage solutions. The storage systems are developed to solve the excess produced energy problem to reuse the stored energy during the peak demand (Voyant et al., 2017). The predictions help energy providers in preparing alternative plans in case of high energy demand and some cases of energy interruptions, to keep the services safe all the time (Ssekulima et al., 2016). The different storage categories and their technical specifications are clearly shown in Table 2.1.

Table 2.1. *The different storage categories and their technical specification (Yahyaoui, 2018)*

Category	Discharge power	Stored energy	Discharge time	Representative application
Bulk energy	10 – 1000 MW	10 – 8000 MWh	1-8 h	Load levelling, generation capacity
Distributed generation	0,1 – 2 MW	50 – 8000 kWh	0.5-4 h	Peak shaving, ransmission deferral
Power quality	0,1 – 2 MW	0.03 – 16.7 kWh	1-30 s	End-use power quality / reliability

As can be seen from Table 2.1, energy storage devices operate at different time intervals. Therefore, their management requires knowledge about the produced energy by renewable energy systems in advance to store sufficient energy in difficult situations. Likewise, to prepare the generation systems and for designing their start-up, the energy that will be produced must be known at different time intervals from one day to three days (Yahyaoui, 2018).

2.2. Solar Forecasting Methodologies

Solar radiation forecasting techniques can be classified into three main different classes. These classes are given in Figure 2.2.

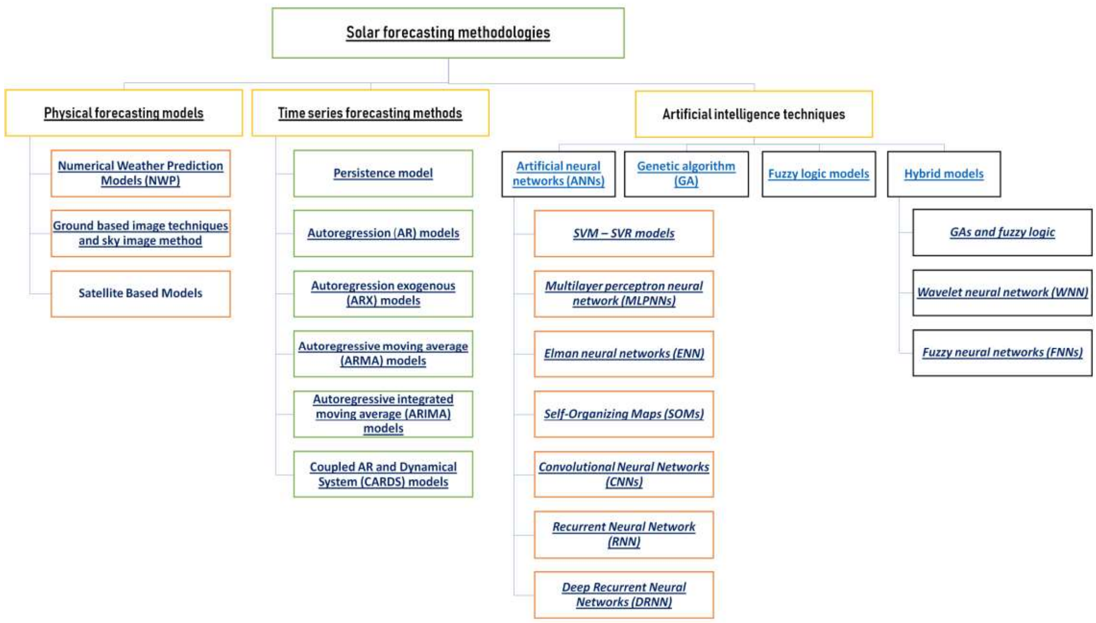


Figure 2.2. *Solar irradiance prediction methods (Alzahrani et al., 2017)*

As can be seen in Figure 2.2 (Voyant et al., 2017), these classes are:

- Physical structural forecasting models, which based on the meteorological and geographical parameters;

- Time-series models, which are based on the historical observed data of the solar radiation as input values;

- AI and hybrid models are based on the historical observed data of solar radiation. A hybrid model is considered as a combination of two or more different models of solar radiation forecasting.

2.2.1. Physical forecasting models

Physical forecasting models contain Ground-based image techniques, sky image method, Satellite-based models and Numerical Weather Prediction (NWP) Models. These models are discussed in detailed as in the following.

2.2.1.1. Ground based image techniques and sky image method

Regardless of the continuous changing of solar radiation throughout the year, cloudiness and cloud thickness has the highest effect on the solar radiation values at the earth's surface level. For this reason, the estimation of the cloud amount at a specific time for a specified place is appropriate for solar radiation forecasting and modeling (Ssekulima et al., 2016). Total sky imager (TSI) is used in ground monitoring method, to show a clear image of the cloud shadows which can be used for forecasting. Figure 2.3 shows the typical structure of solar radiation forecasting based on sky imaging.

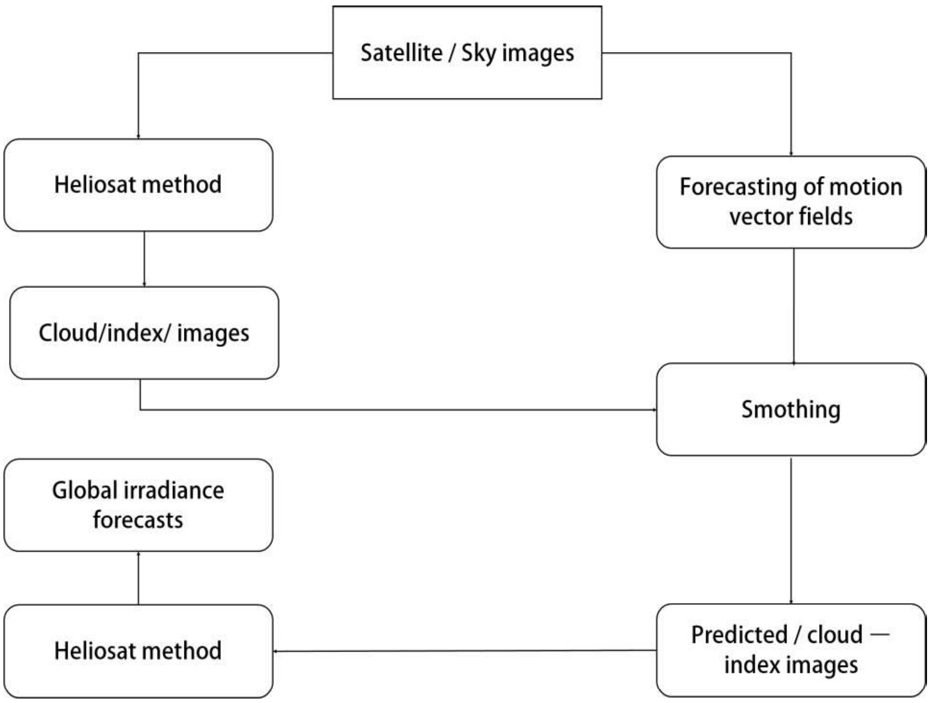


Figure 2.3. *General structure of solar radiation prediction model founded on sky imaging (Ssekulima et al., 2016)*

Compared with NWP and satellite forecasts, which are inadequate until now, ground monitoring method is considered as a good method for spatial and temporal resolution ((Diagne et al., 2013b; Arya et al, 2016). This method has the possibility of catching unexpected and sudden variations in solar radiation (Diagne et al., 2013b). The larger spatial resolution of the cloud field can be projected forward over longer periods (Luukkonen et al., 2013).

Cloud imagery method based on the cloud structure appreciation in the earlier time steps and drawing the movement of the clouds. In order to be easier to predict their next position and size and therefore solar radiation (Ssekulima et al., 2016).

2.2.1.2. Satellite based models

The satellite and cloud imagery-based model is considered as a kind of physical forecasting models (Chaturvedi, 2016). The satellite imagery method considered a developed procedure that is used significantly in solar resource mapping. This method is conceptually similar to the cloud imagery method (Arya et al., 2016; Luukkonen et al., 2013). Unlike Whole Sky Imager (WSI), a much larger spatial scale of cloud patterns can be determined by satellite images method. However, the spatial scale which is determined by ground-based sky images method is larger. The satellite has the potential to determine the required information on cloud motion. The position of the clouds can be located by the cloud motion. Also, solar radiation can be forecasted by using the structure and positions of the clouds (Chaturvedi, 2016).

Clouds in satellite based model reflect light into the satellite which leads to calculating the cloud index accurately (Arya et al., 2016). From consecutive satellite images, the cloud motion vectors can be determined, which are used to optimize obtained results from numerical models. Cloud motion speed is determined by finding the same features in a subsequence image. As assumed in (Luukkonen et al., 2013) the features of the clouds do not change between the subsequence images. To make morning forecasting more accurate, infrared channels must be integrated into the satellite cloud motion forecasts. For short-term forecasting, it is found that satellite forecasts outperform NWP models (Luukkonen et al., 2013).

2.2.1.3. NWP Models

NWP completely depends on atmospheric physics (Chaturvedi, 2016). It can be used in solar radiation forecasting, for scheduling of solar power plants and for predicting the transient variations in clouds, because it has the ability for prediction in different time intervals from one day to many days ahead horizons (Arya et al., 2016).

Approximately 12 international NWP models around the world are used for solar radiation forecasting. However, the most commonly used two model types are the Global Forecast System (GFS) and the Integrated Forecast System (IFS). These models can be used for solar radiation forecasting up to 15 days ahead (Ssekulima et al., 2016). NWP models can be classified into global models, where worldwide simulation of the atmosphere movement is achieved, and regional models, where atmosphere movement is achieved on a smaller scale (continent or country). The classification of NWP models depends on the input parameters, spatial resolution, and the underlying physical models. Determine the forecasting area, data collection development and selection of the NWP systems that uses the fittest physical models, are considered as the important tasks that must be done to start any solar radiation forecasting. The general structure diagram of NWP-based prediction model is shown in the Figure 2.4.

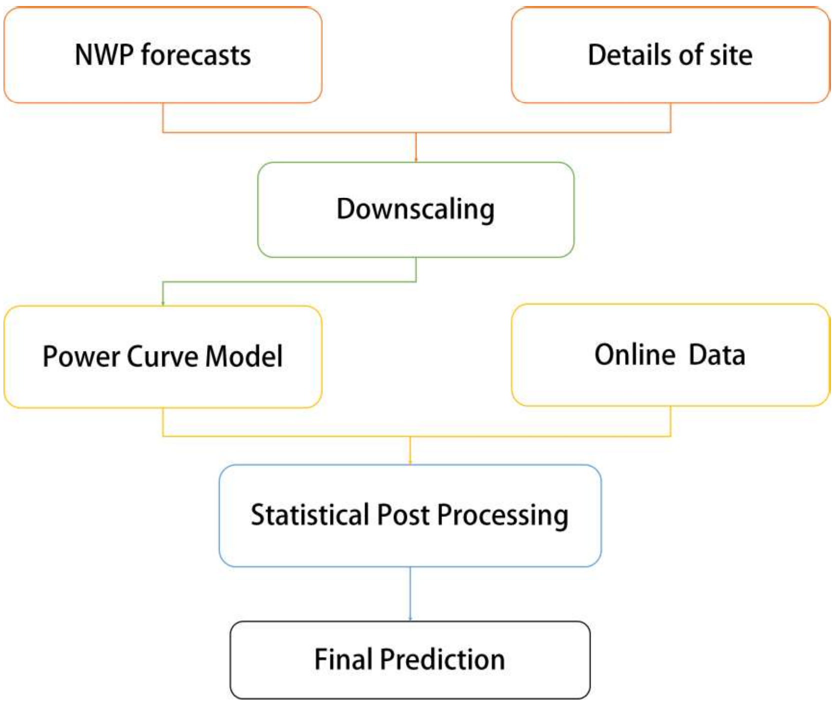


Figure 2.4. *General structure diagram of NWP-based predictions (Ssekulima et al., 2016)*

As can be seen in Figure 2.4, NWP processes can be summarized in three steps: firstly, the primary conditions of the atmosphere are collected by satellites and ground observations. NWP error in the first step is caused by data-assimilation. Secondly, with the integration and calculation of the equations of the atmosphere, the output of the NWP models, which includes hundreds of parameters including solar radiation, is run. In the last step which called statistical postprocessing, NWP output is manipulated by using a trial and error after simulation. In order to find the statistical relation between the measured and output data, we compared them to each other, and therefore correct the error (Chaturvedi, 2016).

2.2.2. Time series forecasting methods

Linear statistical models are a type of time-series prediction method. Linear statistical models are considered as the best choice instead of the physical methods. Time-series models depend on using historical data and focus on the patterns of the data, during the solar radiation forecasting process to develop a relationship between different explanatory variables to predict the future values (Luukkonen et al., 2013). Observation by using this method also known as stochastic process. Conventional statistical methods involve AR model, MA model, ARMA model and other types of the similar models. The mathematical structure of this type of forecasting method is described by:

$$x_t = \sum_{j=1}^{p} \varphi_j x_{t-1} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-1} + k + \varepsilon_t \tag{2.1}$$

where; x_t is the predicted value at the time t, j represents the AR parameter, k is a constant, φ_j is the AR parameter, θ is the MA parameter, ε_t is the white noise, p and q are AR and MA models orders, respectively. Sample and partial auto-correlation function (ACF) of the time series can be used to determine p and q values, respectively. The general structure of ARMA model is illustrated in Figure 2.5 (Ssekulima et al., 2016).

Different types of statistical models are explained in detail in this section: AR model, autoregressive integrated moving average (ARIMA) model, coupled autoregressive and dynamical system (CARDS), and Seasonal ARIMA (SARIMA). All of these models are very useful techniques to calculate, analyze and optimize the stationary and nonstationary processes. For these reasons, these techniques can be used for solar radiations forecasting (Arya et al., 2016).

2.2.2.1. Persistence model

The persistence model is known as one of the simplest models in solar radiation forecasting, at the same time as a naive predictor. The future values are predicted depending on the previous values. Prediction by using this method suppose that solar radiation at any time I_t is equal to solar radiation at the time I_{t-1} with considering the condition that the sky is clear (Ssekulima et al., 2016). For this reason, this method is not recommended for forecasting for more than one hour ahead of when the variations in the weather patterns are high (Chaturvedi, 2016). For one-hour forecasting, the results which are obtained by the persistence method and those which are obtained by satellite-derived cloud motion are approximately the same as those are obtained by linear statistical method. The reason for the approximate results can be the navigation and parallax of the satellite, which tend to mitigate for longer times (Luukkonen et al., 2013). The persistence model can be expressed mathematically as follows:

$$I_t = I_{t-1} \tag{2.2}$$

where; I_t is the irradiance at any time, I_{t-1} is the irradiance at the previous time step.

2.2.2.2. AR models

In the AR models, the results are predicted by using a linear combination of the previous values of the variable. However, in the multiple regression models, the results are predicted by using a linear combination of predictors (Ssekulima et al., 2016). AR model can be defined as a regression of the variable against itself. Consequently, the AR model can be calculated as follows:

$$x_t = k + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + ... + \varphi_p x_{t-p} + \varepsilon_t \tag{2.3}$$

where; x_t is the forecasting parameter, k is a constant, φ_i is the AR parameter, x_{t-1} is the forecasting parameter at the previous time step, ε_t is the white noise, p is the order of AR model. AR models are noticeably flexible at dealing with a wide range of different time series problems.

The general structure diagram of ARMA models is shown in the Figure 2.5 below.

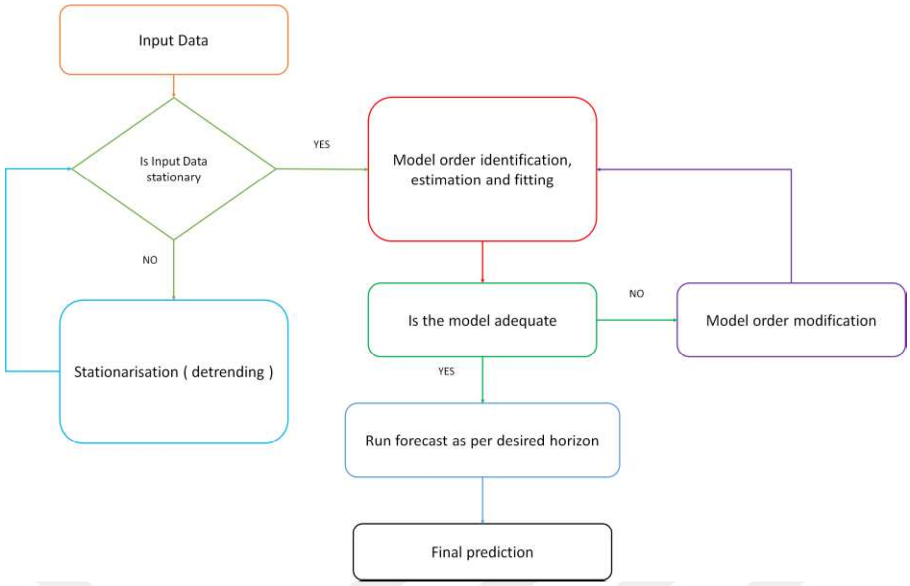


Figure 2.5. *The general structure diagram of ARMA model*

AR models are used for hourly solar radiation prediction up to 36 h. AR model is operative at mean reversion but inoperative for peaks modeling of the resulting residual series which are obtained by subtracting the Fourier series from the original series (Diagne et al., 2013b).

2.2.2.3. ARX models

ARX models are divided into linear and nonlinear ARX models that integrate each other. By using flexible nonlinear functions, the structure of ARX models makes it possible to model complex nonlinear behavior, for example, wavelet networks and sigmoid networks. Linear ARX models' mathematical structure is defined as:

$$\begin{aligned} &y(t)+a_1y(t-1)+a_2y(t-2)+\ldots+a_{na}y(t-na)= \\ &=b_1u(t)+b_2u(t-1)+\ldots+b_{nb}u(t-nb+1)+e(t) \end{aligned} \tag{2.4}$$

where; *u(t)* is the input function*, y(t)* is the output function, *e(t)* is the noise function, *na* is the number of past output terms, *nb* is the number of past input terms used to predict the current output. To make the notation simple the input delay *nk* is set to zero. So, the equation (2.5) below expressed the ARX model and is given by:

$$\begin{aligned} &y_p(t)=[-a_1,-a_2,\ldots,-a_{na},b_1,b_2,\ldots,b_{nb}]\cdot \\ &\cdot[y(t-1),y(t-2),\ldots,y(t-na),u(t),u(t-1),\ldots,u(t-nb-1)]^T \end{aligned} \tag{2.5}$$

where; $y(t-1), y(t-2),..., y(t-na)$ are delayed inputs, $u(t), u(t-1),..., u(t-nb-1)$ are output variables, $[-a_1,-a_2,...,-a_{na}, b_1, b_2,..., b_{nb}]$ are coefficients vector represent the weight which applied to the regressors, y_p is the current output. Nonlinear ARX models have in their structures a more flexible nonlinear mapping function *f,* instead of the weighted sum of the regressors (2.6). Nonlinear ARX regressors can be both delayed input-output variables and more complicated (http-4).

$$y(t) = f(y(t-1), y(t-2), y(t-3),..., u(t-1), u(t-2), u(t-3),...) + \varepsilon_t \quad (2.6)$$

where; $y(t)$ is the nonlinear ARX, $y(t-1)$ is the delayed input, $u(t-1)$are output variables, ε_t is white noise. Model regressors are the inputs to *F*, in the nonlinear ARX model structure different nonlinear functions chosen (http-4).

ARX models are used for hourly solar radiation prediction for horizons up to 36 h (Diagne et al., 2013b).

2.2.2.4. ARMA models

An ARMA model is a combination of two elementary models, MA and AR model. This model can be presented mathematically as it is shown below:

$$x_t = \sum_{i=1}^{p} \varphi_i x_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + k + \varepsilon_t \quad (2.7)$$

where; x_t is the predicted value at time *t*, *i* represents the AR parameter, *j* represents the MA parameter, φ_i is the AR parameter, θ_j is the MA parameter, *k* is a constant, ε_t is a random white noise, *p* and *q* are AR and MA models orders, respectively (Box et al, 2015).

By removing the annual periodicity and differentiating of the solar radiation in different seasons, ARMA models create an immediate series of global solar radiation. ARMA model is considered as a flexible model because of its ability to extract useful statistical properties and to represent several different types of time series by using different orders (Diagne et al., 2013b). The forecasting results are obtained by using this model are closing to the measured values and show a good performance (Diagne et al., 2013b).

2.2.2.5. ARIMA models

ARIMA models are used more than other models because the uploaded input data to these models are nonstationary yet (Ssekulima et al., 2016). The mathematical structure of ARIMA models is described as:

$$y_t = \alpha + \sum_{i=1}^{p} \varphi_i y_{t-i} - \sum_{j=1}^{q} \theta_j u_{t-j} + e_t \tag{2.8}$$

where; y_t is the predicted data, α is a constant, $\varphi_i y_{t-i}$ is the linear combination lags of y, $\theta_i u_{t-i}$ are the linear combinations of lagged forecast errors (http-1).

i represents the AR parameter, j represents the MA parameter, φ_i is the AR parameter, θ_j is the MA parameter, k is a constant, e_t is the prediction error. random white noise, p and q are AR and MA models orders, respectively

The ARIMA models are considered as the reference estimators in the global radiation forecasting. The sharp transitions in the radiation are captured by these models more accurate than other models (Diagne et al., 2013b). Seasonal ARIMA models “SARIMA” are developed by (Craggs et al., 1999), which have the same structure with ARIMA models including additional seasonal terms, which can be shown as follows.

$$ARIMA(p,d,q)(P,D,Q)^S \tag{2.9}$$

where; d is the differencing order, P, D *and* Q are seasonal AR, differencing, and MA models orders respectively, S time span by which seasonal data is repeated.

2.2.2.6. CARDS models

CARDS model is published by (Boland, 1995) and (Mellit et al., 2009) for solar radiation forecasting in time series for one-step-ahead at different time scales. The results that can be given by using this model have a lower normalized root mean square error (NRMSE) value for all days (Chau, 2006). This model is described as:

$$\dot{R} = z \tag{2.10}$$

$$\in \dot{Z} = k(z+R) - \lambda(3R^2 z + R^3) - \in z - \gamma R - b + \zeta \tag{2.11}$$

where; κ, λ, ϵ, γ and b are the parameters that can be adjusted, ζ is a noise term, $\dot{R}$ denotes the derivative of R with respect to time, and hence z stands for the second derivative of R

with respect to time. The following discretized version of the model for our deseasoned solar radiation time series R_t shall be used as:

$$R_{t+1} = R_t + z_t \Delta_t + \omega_t \tag{2.12}$$

$$Z_{t+1} = Z_t + \left[k(z_t + R_t) - \lambda\left(3R_t^2 z_t + R_t^3\right) - \in z_t - \gamma R_t - b \right] \frac{\Delta t}{\in} + a_t \tag{2.13}$$

where; ω_t and a_t are noise terms, Δ_t is the time interval, k, λ, ϵ, γ and b are parameters that can be calculated by using the method of ordinary least squares (Diagne et al., 2014).

Statistical models can provide moderately exact performance for the forecasting that take a short time and have the ability to correct the local directions in the data. Nonetheless, statistical procedures need considerable quantity over historical day sequence data. In statistical models, there is a disadvantage that their performance in medium and long-term prediction processes cannot be more accurate as in a short-time prediction process (Ssekulima et al., 2016).

2.2.3. AI techniques

According to the researches results for the last two decades, AI technologies have been showing a great noticeable difference in the forecasting processes, when they are compared with the traditional forecasting approaches (managerial and quantitative) (http-3). There are three main methods commonly use AI systems for forecasting:

ANNs simulate the human cognitive process and try to solve problems by learning and analyzing from the already found data. ANNs are also considered as a good choice for modeling complex behaviors and patterns (Pijanowski et al., 2002). For instance, the ability to recognize and learn patterns. The structure of ANNs is made up of a lot of nodes that work as mini-computers to process input data and send the results to the next nodes in the network. One of the advantages of these networks that there is no need for prior assumptions about possible relationships.

Expert systems are considered one of the most commonly used AI forecasting methods. The totality of available knowledge, which stored in a set of “if-then” rules and obtained by interviewing experts or integrating sets of data, and rules are epitomized by Expert systems. For instance, weather conditions prediction depends on many factors like weather temperature, humidity levels, and the location where the prediction is achieved (Pijanowski et al., 2002).

Belief networks use a tree format for describing the database structure, where the nodes represent variables and the branches the conditional correlations between them. Conditional probabilities for the diversity of the upcoming future data are generated by Belief Networks (Pijanowski et al., 2002).

These AI systems can be used for classification and prediction problems. AI techniques are considered as ones of a non-linear statistical model. In the following subheadings, most of the AI techniques and methods that are used for solar radiation forecasting will be explained in detail as follows:

2.2.3.1. ANNs

ANN is considered as a processing system which connects small processing units. ANN in shape and work simulates the shape and the work of the biological neural network in the human body. Generally, ANN is made up of three main layers, an input layer, an output layer, and one or more hidden layers. In each layer, several nodes called neurons are connected through connections. Each connection has a weight (Grekousis, 2019). These nodes are grouped in layers with connections between them. The arrangement and connections of the nodes, between the different layers, are exemplarily referred to as ANN architecture (Grekousis, 2019). The more complex data, is the higher number of hidden layers, connections, and neurons in the ANN.

ANNs can automatically learn to realize manners used in data from the previous data that can be added to the network (Kalogirou, 2001). The biggest advantage of ANNs that is not found in the other forecasting methods and models, is their ability to dealing with complex and non-linear processes. Shortly, an ANN receives data through input neurons and processes and analyses them to give results as output.

The method of changing the weights of the connections between the layers is achieved through the options of the Machine Learning (ML) method. The main classes of ML methods are; supervised ML, unsupervised ML, semi-supervised ML, and reinforcement ML. Choosing a suitable and right learning method depends on the problem that is to be solved, how it will be conceptualized (predictive model, clustering, classification) and the available data.

ML can also be classified into two main classes: shallow learning and DL. The difference between these classes is summarized in the depth of the analysis and the complexity of the network's architecture (Deng & Yu, 2014).

Shallow ANNs were used for a long time until 2006 the accurate results of the DNNs are noticed. Interests in DL have been starting in progress since 2006, with many experiments with a large number of data that support using DNNs rather than shallow ANNs (Grekousis, 2019). A shallow learning method contains one or two hidden layers; however, DL contains more than two hidden layers and more developed architecture than shallow learning. More hidden layers in the ANNs easily discover structures in a large number of data and extract worthy knowledge. As the DNNs have more than two hidden layers, appreciably more parameters (weights or biases) need to be adjusted. For this reason, DNNs need more data to get trained than a typical shallow ANNs need (Deng & Yu, 2014; Grekousis, 2019).

Shallow ANNs have few numbers of hidden layers, for this reason, the back-propagation (BP) learning algorithm is mainly used. However, with a large number of hidden layers, it can suffer from the vanishing gradient problem. However, in deeper ANNs which contain more sophisticated architectures, new activation functions, and better regularization techniques; the BP learning algorithm can perform more efficiently.

The difference between ANNs and DL cannot be reduced only in the number of hidden layers but the processing functions, the entire architecture, and regularization techniques which completely change the ANN shape.

ANNs can be in different forms - multilayer perceptrons (MLPs), recurrent neural network (RNN) and radial basis function network (RBFN). However, the most common form of ANNs is MLP structure, where, three main layers (input, hidden and output layer) are considered as the main component of this structure. They are connected in interconnected processing elements called nodes, by an interfacing "connection" link (Praynlin & Jensona, 2017). Each interfacing link generally contains transmitted incoming signal multiplied by its weight. Through an input layer, the data can be presented to the neural network, the hidden layer can be one or more than one layer which is characterized by many neurons and the output layer gives the results. The activation function can be calculated by equation (2.14) which is called the tangent hyperbolic function *f(x)*. Hyperbolic function considered as a non-linear function (Diagne et al., 2013b). Learning in ANNs includes adjusting the weights of the links (V. Bhatnagar, 2014).

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.14}$$

ANNs can automatically learn to realize the manners that can be presented suddenly in data from real systems, physical models or other sources. The result of the output layer *y* is given by the equation (2.15):

$$y = y(x;w) = \sum_{j=0}^{h} \left[w_j f\left(\sum_{i=0}^{h} w_{ji} x_i \right) \right] \tag{2.15}$$

All network layers, except the input layer, process the input data by multiplying them by their weights, then process the sum by using a non-linear function to obtain satisfactory results (Chau, 2006; Praynlin & Jensona, 2017). In the following some of the most used ANNs type models will be discussed in detail:

SVM – SVR models

In 1986, Vladimir Vapnik (Diagne et al., 2013b) has introduced a new ML technique which is based on the kernel method and used in classification and regression problems. But the method used in time series forecasting problems is called SVR which based on the application of SVM. The results calculated by an SVM can be given by:

$$\hat{y} = \sum_{i=0}^{n} \alpha_i K(x_i, x_*) + b \tag{2.16}$$

$$K(x_p, x_q) = e^{\left[-\gamma \left| x_p - x_q \right|\right]} \tag{2.17}$$

where; x_* is the input test case, x_i is the input training case, b is the bias parameter is derived from the preceding equation and some specific conditions, γ is the hyperparameter of the covariance function, α_i is the coefficient is related to the difference of two Lagrange multipliers, which are the solutions of a quadratic programming (QP) problem which has a unique solution. The optimal selection of the cost function in the QP problem gives permission to get a scattered solution. SVs are defined as the examples which accompanies the nonvanishing coefficients (Soubdhan et al., 2015).

MLPs

MLPs are considered as one kind of ANNs. Typically, it is made up of three main layers type; the input layer, several different hidden layers, and one output layer (Atkinson & Tatnall, 1997). MLP neural networks have a feed-forward architecture, and they are made up of perceptrons which are arranged in layers.

Rosenblatt (F. Rosenblatt, 1958) has built the prototype of the first ANN which was a form of a single perceptron receiving several inputs and one or more outputs exited from it; that is either zero or one, can be seen in Figure 2.6. Data in the Feedforward ANNs directs in one way and one direction, from the input nodes to the output nodes. The output results depend on the input and the values of the weights (F. Rosenblatt, 1958; Grekousis, 2019). The general structure of these networks is shown in Figure 2.6.

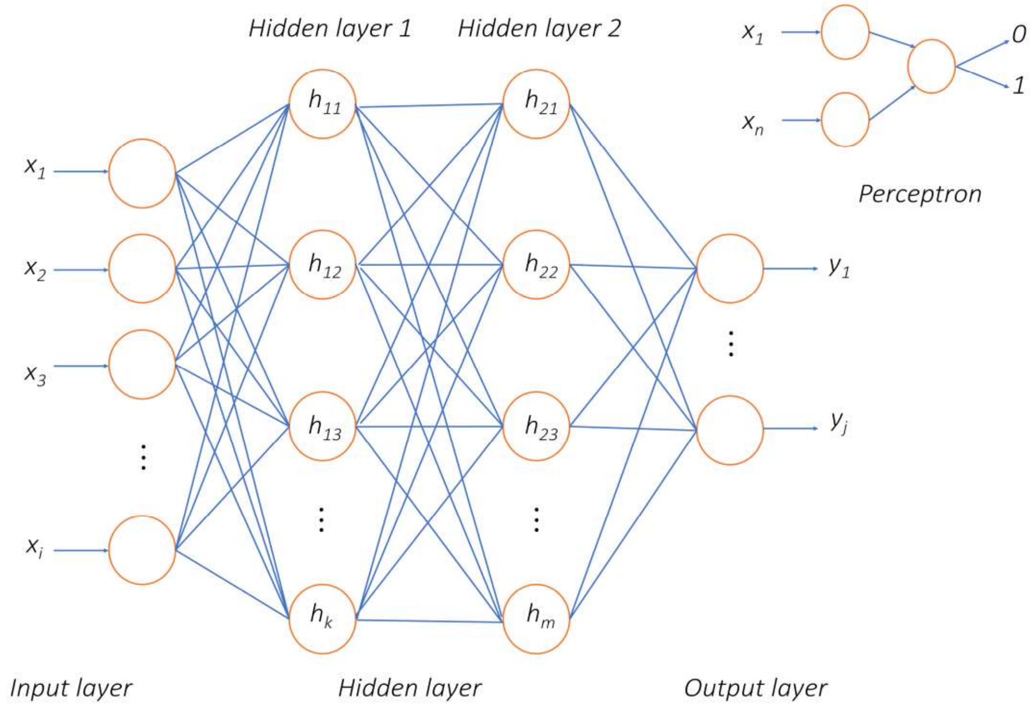


Figure 2.6. *The general structure of MLPs (Atkinson & Tatnall, 1997; Grekousis, 2019)*

The layers neurons are connected to the neurons of the previous and next layer. However, the neurons that are located in the same layer are not connected. The mathematical model of the MLPs is shown below:

$$y_j = f\left(\sum_{i=0}^{m} w_{ij} x_i + b\right), \qquad i = 1,2,\ldots,l \tag{2.18}$$

where; j is the number of the neurons used in the output layer, i is the number of the neurons used in the input layer, w_{ij} are the weights of links connecting the neurons of the input, hidden and output layers (Ko & Lee, 2013).

As well known, the output results are affected by the value of weights, due to this BP learning is used to find these values. But the BP method has a disadvantage that is, it needs more time for the iterative training process (Elsheikh et al., 2019; Ko & Lee, 2013).

Elman neural networks (ENN)

Generally, the ENN is considered as one type of the RNNs which are a type of ANNs, in which the directed cycle between the neurons is created by connections leading to dynamic temporal behavior. The difference between RNNs and feedforward ANNs can be summarized in that RNNs use the internal memory to treat the sequences of the inputs, however feedforward ANNs are not. For this reason, RNNs are commonly used for handwriting or speech recognition.

Figure 2.7 represents the general structure of the ENNs' that consists of four main computing layers (input, hidden, recurrent, and output layer).

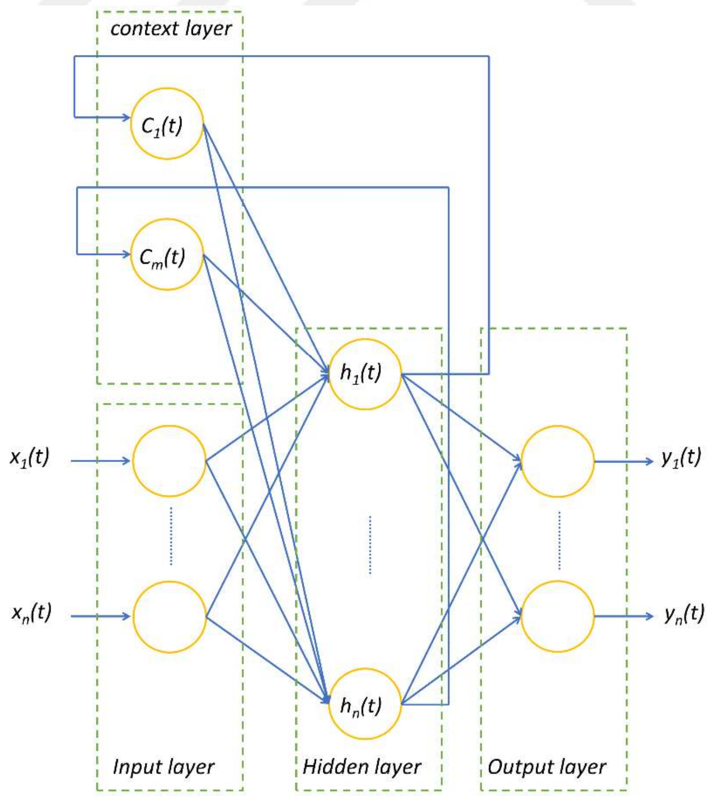


Figure 2.7. *The general structure of the ENN (Ren et al., 2018)*

The ENNs are developed by Elman in 1990 (Elman, 1990). In the ENNs, as can be seen in Figure 2.7, the output can feedback to itself by using a recurrent layer, in which the recurrent neurons number is equal to the number of the hidden neurons. The feedback character of these networks gives it a high capability to learn and build Time–Location patterns, in addition to a dynamic function, that makes the system capable to be usable with the time-varying properties (Elsheikh et al., 2019). Every node is joined to a single recurrent node using a constant weight value.

The input of the neural network is represented by *u(k-1)*, however recurrent layer is represented by $x_c(k)$ and the output of the hidden layer by *x(k)*. Equation (2.19) represent the mathematical structure of the ENNs:

$$x(k) = f\left(w_2 x_2(k) + w_1 u(k-1)\right), x_c(k) = x(k-1) \tag{2.19}$$

where; w_1 is the connection weight between the input layer nodes and the hidden layer nodes, w_2 is the weight of the connection between the recurrent layer and the hidden layer and *f* is the transfer function used in the hidden layer and it can be given by the sigmoid function. Also, the hidden layers' output which is calculated as the output of the neural network *y(k)* is computed as:

$$x(k) = g\left(w_3 x(k)\right) \tag{2.20}$$

$$E = \sum_{k=1}^{K} \left(d(k) - y(k)\right)^2 \tag{2.21}$$

where; w_3 is the connection weight between the hidden layer nodes and an output layer node, *g* is the output layers' transfer function, *d(k)* is the desired output of the input *u(k)*. For updating the weights of the this neural network BP is used; where the error of the ENN is calculated by equation 2.21 (Jia et al., 2014).

Self-Organizing Maps (SOMs)

SOMs are considered as one of the most common ANNs models which use the unsupervised learning method to process a large number of data by representing similarities of the input data founded on topological connections on charts (Grekousis, 2019; Kohonen et al., 2001). Figure 2.8 illustrates the common design of the SOMs.

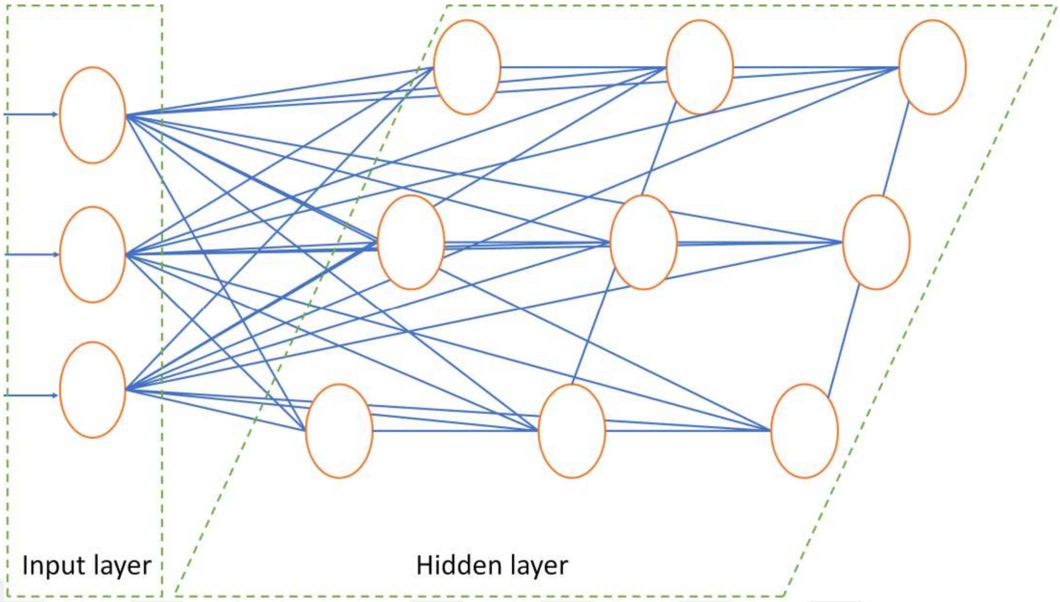


Figure 2.8. *The general structure of SOMs (Grekousis, 2019)*

As can be seen in Figure 2.8 each node is connected with a weight vector of the same aspect as the input vector and its place on the map. There is no need for an output vector (external supervision) because the training is unsupervised. There is a similarity between the concepts of the SOMs' topological structures and geographical concepts (Grekousis, 2019).

Convolutional Neural Networks (CNNs)

The first research where CNNs were introduced, for data analysis, as a process of recognizing patterns and image recognition, with the use of multiple arrays called tensors was published by (Le et al., 1998). The structure of a CNN consists of different operational blocks, where the BP algorithm is used, the first group of layers is convolutional layers which have a significant effect on image recognition processes. Non-linearity and pooling layers are considered as optional layers and can be ignored. The other layers are similar to the fully connected layers of MLPs. Fully connected layers are also considered as ones of additional convolutions. The most commonly used activation function in CNNs is the Rectified Linear Unit (ReLU) activation function.

From the input image, features are received by convolution blocks. By learning features using filters, which make a specific activation map; the features gain the ability to maintain the spatial arrangements of pixels. Figure 2.9 illustrates the common design of the CNNs.

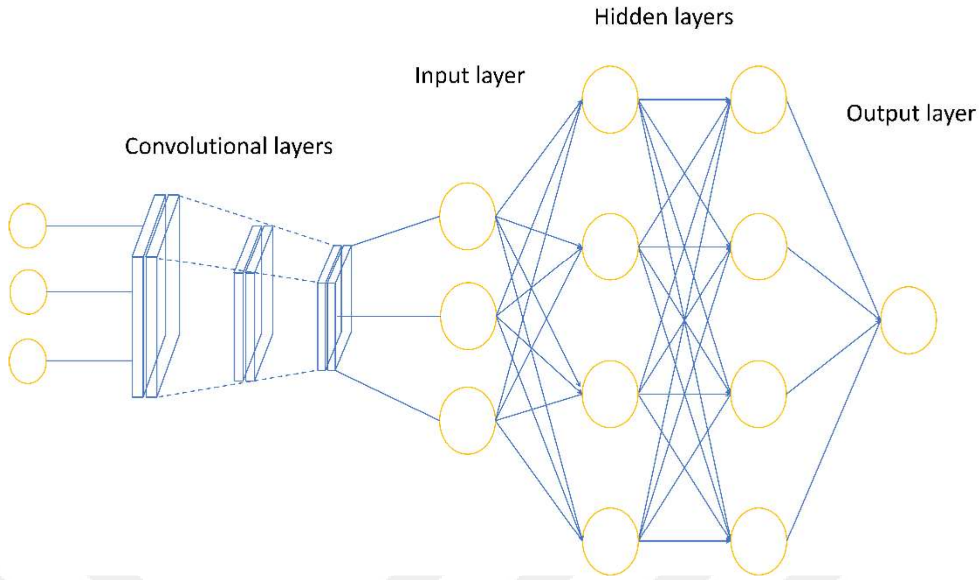


Figure 2.9. *The general structure of the CNNs (Grekousis, 2019)*

CNN aims to find the values of the filters during the training period. The more filters used; the more objects identified in an image (Grekousis, 2019).

Recurrent Neural Networks (RNNs)

An RNN is considered as one of the most commonly used types of neural networks, which can be used for sequential data prediction. RNNs are found in several applications for solving some problems, such as image classification and processing, analyzing feelings, translation from one language to another, and speech recognition. With using internal memory, that can be used for storing information about previous calculations, the RNN has the ability to predict a random sequence of inputs. The general structure of the different types of RNNs is shown in the Figure 2.10 below.

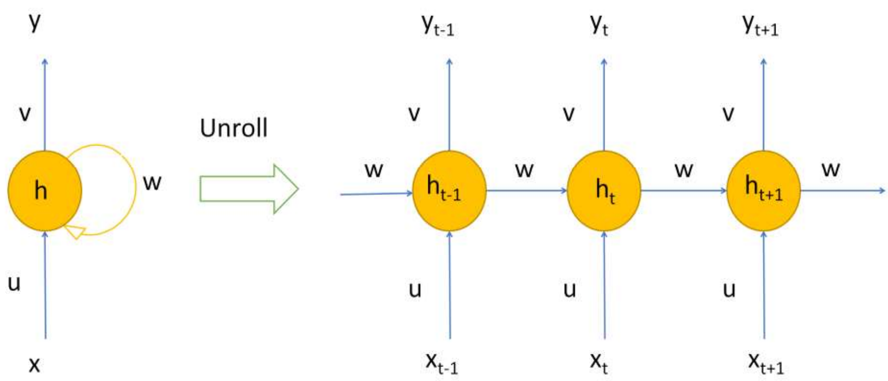


Figure 2.10. *The general structure of the RNNs unfolded RNN (left), and folded RNN (right) (Alzahrani et al., 2017)*

As shown in Figure 2.10, the general structure of the RNN can be in two types; folded and unfolded RNN. In the RNN the hidden connection h owns feedback from the other connections in a previous time interval multiplied by its weight w. The input data of the hidden connection is received from the connections located at the previous time-interval when basic RNN is branched out a full network. The output connection of RNN takes the result of multiplying the input, which can be taken just from the hidden connections, by the output weight v. The mathematical structure of the RNN is expressed as follows:

$$h_t = f_h(u \cdot x_t + \omega \cdot h_{t-1}) \tag{2.22}$$

$$y_t = f_y(v \cdot h_t) \tag{2.23}$$

where; f is the activation function, x_t is the input value, t is an instant time, w is the weight vector, h_t, h_{t+1} are the previous and next hidden neuron respectively, v is the output weight, u is the input weight.

During the training period RNNs similar to ANNs, use the same method for the BP known as back-propagation through time (BPTT). The difference between BP and BPTT is that BP depends on the existing status values, but BPTT bases on the existing and past status values (Alzahrani et al., 2017).

Deep Recurrent Neural Networks (DRNN)

As mentioned before, any ANN consist of two and more hidden layers of neurons is called DNN. According to different researches mentioned in (Alzahrani et al., 2017; Pascanu et al., 2013), RNNs with two or more hidden layers, work more efficiently and give more accurate results than RNNs with less hidden layers.

In Figure 2.11 the possible architectures of different types of RNNs as reported in (Alzahrani et al., 2017; Pascanu et al., 2013) are shown. Figure 2.11 (b) illustrates the common design of DRNN with a deep transition, which is responsible for increasing the number of nonlinear intervals, that can arise the intricacy of the network training. Figure 2.11 (c) illustrates the common design of DRNN with a shortcut connection, that gives a possibility to a shorter path. Stacked RNN which contains a stacked hidden layers u_t, used with various time intervals, plays an important role in reading from sensors that have different sampling rates. DRNN with additional deep outer layer, in between hidden and output layers, is shown in Figure 2.11 (d). General structure of Stacked RNN is illustrated in Figure 2.11 (e).

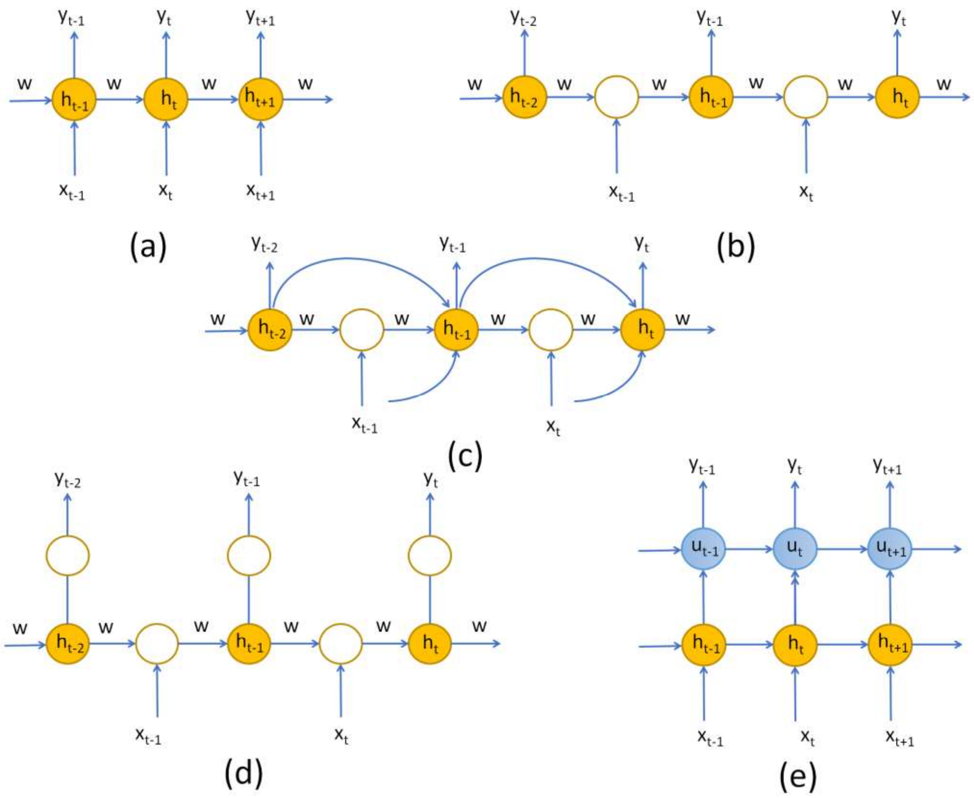


Figure 2.11. *The general structure of the DRNNs with different types a) RNN, b) DRNN, c) DRNN with shortcut connections, d) DRNN with deep output layer, e) Stacked-RNN (Alzahrani et al. 2017; Pascanu et al. 2013)*

The long-term and short-term memory (LSTM) method is introduced to overcome vanishing or exploding gradient, that may happen when the gradient of RNNs use their connection for short-term memory in case of difficulty to track in long-term memorization (Alzahrani et al., 2017; Pascanu et al., 2013).

2.2.3.2. Genetic algorithms (GAs)

GA method developed by Holland (Mellit & Kalogirou, 2008), is considered as a research method and optimization procedure. This method can be categorized as an evolutionary algorithm that depends on the imitation of nature's work from a Darwinian perspective. The GA uses a search technique to find adjusted or approximate solutions for optimization. GA is categorized as global search heuristics, which includes 3 major genetic operators: selection, crossover, and mutation. The operators are used to adjust the selected solutions and choose the most proper race to transfer their characters to the upcoming generations. The basic steps are to modeling possible solutions to the problem in the form of strings of ones and zeros. The method seems promising but has a disadvantage of excessive complexity when it is used to solve problems with a huge amount of data. GA can be used with other intelligent techniques that are mentioned above (Mellit et al., 2009). R&D efforts are focused on:

- The ways that can be used to perform neural networks for easier modeling and design by GAs;
- The ways that can be used to replace GAs for ML algorithms and
- The chances of combining created models from evolutionary computing (Alzahrani et al., 2017).

2.2.3.3. Fuzzy logic (FL) models

FL is considered as a generalization of classical logic which was founded on fuzzy set theory alongside the related methods developed by Lotfi Zadeh (Mellit et al., 2009; Ssekulima et al., 2016). Modeling by FL is based on the 'if-then' principle, where 'if' represents the fuzzy explanatory variables vector, in another meaning 'inputs', and 'then' is a consequence 'output' (Mellit et al., 2009). The variable's membership function assessment is considered as an essential stage in modeling fuzzy models. Membership functions are important for modeling decision-maker processes and can be gained based on practical statistical studies (Mellit et al., 2009).

The aim of FL models is simulating the human cognition side which is known as approximate reasoning (Ssekulima et al., 2016). Besides, this model can be used in control engineering (Mellit et al., 2009). The utility of this model can be reduced in its ability to deal with cases in which the model of the system is inconvenient (Ssekulima et al., 2016). FL predictions procedure is shown in the Figure 2.12 below.

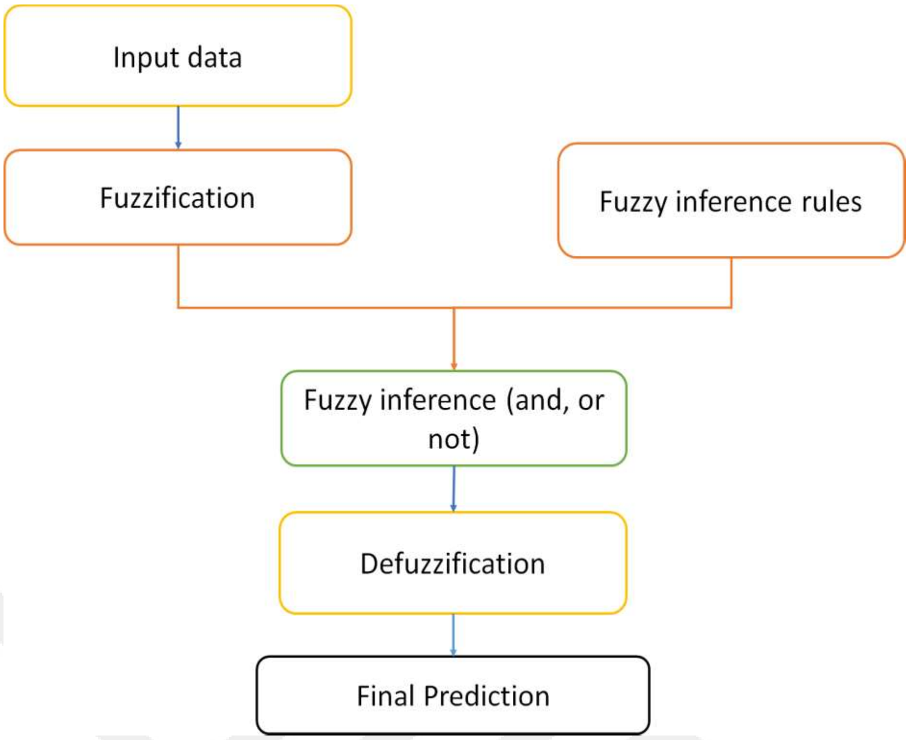


Figure 2.12. *General structure of the FL prediction models*

FL models can be used for determining non-linear mapping for complicated solar radiation. It can be used also for calculating wind speed time series problems even without having a large number of data (Ssekulima et al., 2016).

2.2.3.4. Hybrid models

The hybrid model is a kind of new method which consists of two or more forecasting models to ensure solar radiation forecasting more accurate. For this reason, they are known as combined models. For example, hybrid models can be a combination of linear and nonlinear models, a combination of two linear models or two nonlinear models.

One of the hybrid models which is developed by (Ji & Chee, 2011), it is a combination of ARMA and time delay neural network (TDNN), which is developed to improve the prediction accuracy. In order to fit the linear component and to make the TDNN model able to find the nonlinear pattern found in the remaining part, the daily solar radiation data are merged by linear and nonlinear components and use the ARMA model. This hybrid model owns the ability to utilize the unparalleled features and durability of both models. The results that can be obtained by using this model are more accurate than obtained data by using ARMA or TDNN models in separate (Ko & Lee, 2013).

As shown in the previous studies hybrid forecast methods outperform individual forecast (Chaturvedi, 2016). The increasing popularity of hybrid systems is due to their ability to solve complicated problems. Hybrid systems strongly depend on some intelligent systems such as neural networks, FL, and GAs in solving problems (Mellit & Kalogirou, 2008).

The hybrid model developed by Cao (Cao et al., 2006; S. Cao et al., 2005) for forecasting the sequences of the total daily solar radiation, combines ANN with wavelet analysis (Ko & Lee, 2013). The other model developed by Cao and Lin combines ANN with diagonal recurrent WNN (Box et al., 2015). This model is used for hourly solar radiation prediction. Various methods of meteorological observations can be used as input to the hybrid models; however, the other methods are used for forecasting the global solar radiation and daily cloud movements.

A lot of hybrid models such as FNNs, GAs with neural networks, wavelet transform (WT) with neural networks, and GAs with FL will be discussed in detail as follows.

FL and GAs: FL and GAs systems united in one system (hybrid system) operate well under the same conditions; including non-linearities and the need for high efficiency. For this reason, these technologies can be used well either in the stand-alone or in the transformational models. FL systems performance optimization is facilitated by GAs, which can also be used for setting up membership values, modifying membership functions and obtaining FL rules. The system complexity is simplified by flexibility and abstraction provided with the use of membership functions and associated parameters. In addition, GAs need the operation of genetic systems, to carry out the evaluation function. It can be facilitated by FL control. This type of system is developed to create models with qualified learning abilities (Mellit et al., 2009).

Wavelet neural network (WNN): It is a combination of the WTs and the neural networks. WNN uses WTs as an activation function for the network construction. Unexpected effectiveness can be shown by the WNN in solving the common problems of weak convergence or divergence of some neural networks (Mellit et al., 2009).

WT considered as a new technique used for signal processing is developed from Fourier transform. This technique has essential characteristics, such as the time-localization feature, that is chosen depending on the signal type that has been analyzed (Diagne et al., 2013b; Mellit et al., 2009). Besides, its function is considered as a local function. Recently it has generated enormous attention in the theoretical and applied fields (Diagne et al., 2013b). The general structure of the WNN is shown in the Figure 2.13.

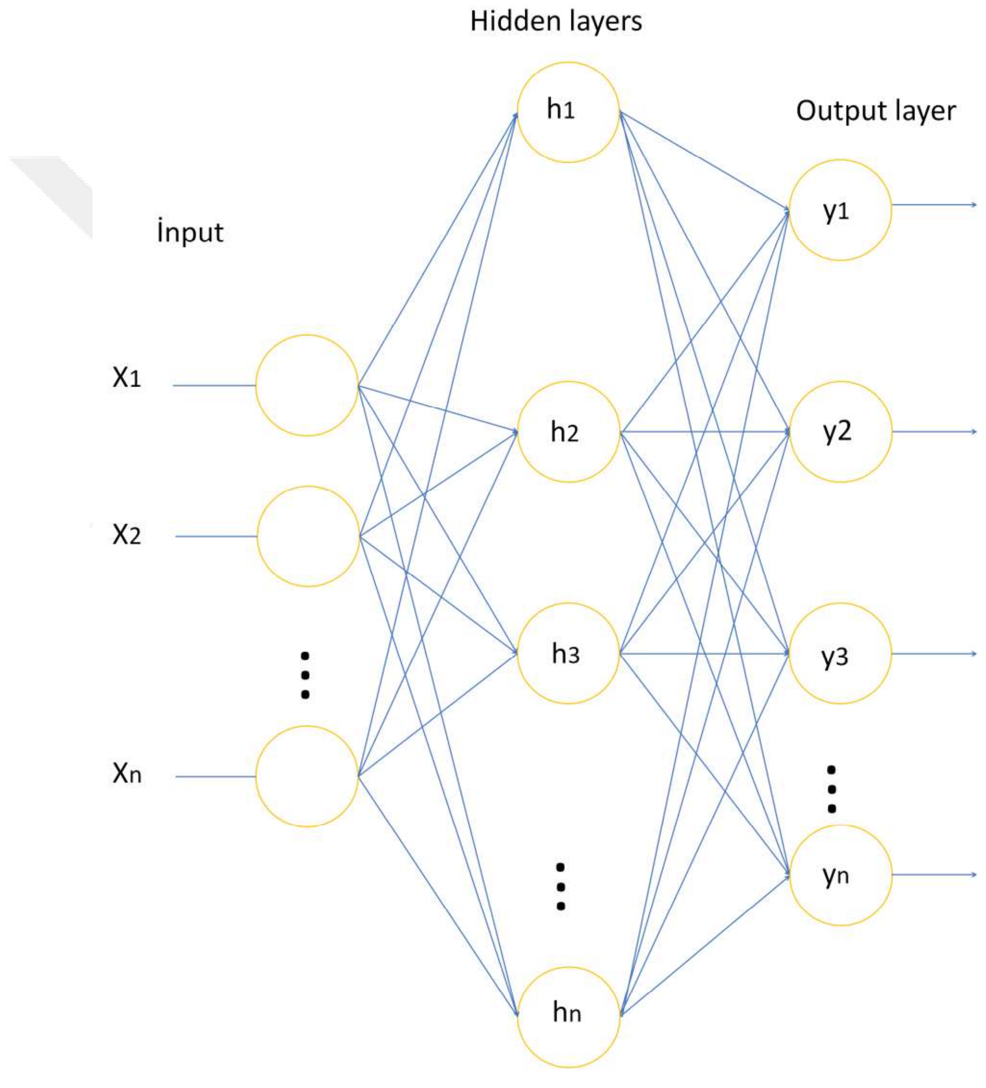


Figure 2.13. *The general structure of WNN (Diagne et al., 2013b)*

As indicated in Figure 2.13, WNNs consist of three layers as ANNs have (Mellit et al., 2009). In the equation below, the mathematical structure of the WNN is shown:

$$y(k) = \sum_{i=1}^{m} \omega_{ik} h(i), \qquad k = 1,2,...,l \tag{2.24}$$

$$h(j) = h_j \left[\frac{\sum_{i=1}^{n} \omega_{ij} x(i) - b_j}{a_j} \right], \qquad j = 1,2,..., m \tag{2.25}$$

where; x is the input vector, h is the wavelet function, w_{ij} is the input-hidden layer connection weight, y is the predicted output, $h(i)$ is the output value for node i in the hidden layer, w_{ik} is the hidden-output layer connection weight, l is the number of output nodes, m is the number of hidden layers.

Fuzzy neural networks (FNNs): They are considered as a modified method with improved performance, combining the FL method with a neural network. FNNs are considered as one of the AI techniques and used for hourly global radiation prediction from satellite pictures (Mellit & Pavan, 2010). The fundamental concept used for integrating FL with neural network technologies is the fuzzy neuron (Mellit & Kalogirou, 2008). In FNNs nodes can contain modified neurons in each layer. The input data with their weights are represented by membership functions, consist of a set of fuzzy values (Mellit & Kalogirou, 2008). The membership functions can be found by an adjusted summation process. The adjusted summation process is used for subjoining the obtained membership functions in order to procure another one which illustrates the integration of the weighted fuzzy inputs to the node. According to (Mellit & Kalogirou, 2008; Mellit & Pavan, 2010), the results that can be obtained by using the FNN method seem more accurate than the results that can be obtained by time series forecasting methods.

3. SOLAR RADIATION FORECASTING METHODS

In this section, it is going to be talked about the problem settings that will be solved by using different methods. The models that are going to be used are classic mathematical models (CPR, CPRG, Liu and Jordan, Jain model 1, Jain model 2), ANN and DNN models. The main model that is going to be focused on is the DNN model. Finally, developing new models is going to be discussed.

3.1. Problem Settings

The main aim of this study is to create an ideal model to make solar radiation prediction more accurate and to maintain reasonable simplicity. Solar radiation values can change depending on many weather factors such as, cloud cover, precipitation, temperature, wind speed, pressure, and humidity. Nevertheless, the precise correlation between each weather feature and the received solar radiation can be hard to find. Therefore, this study is aimed at creating a direct relationship between the weather variables and solar radiation by using the DNNs model which is considered one of the best recent forecasting methods. DNNs can be used to find the linear as well as non-linear relationship between features and targets without much feature engineering or domain knowledge. We plan to use a deep fully connected neural network to find the direct correlation by training the network as a MLR problem.

3.2. Dataset

The dataset is collected from the Turkish State Meteorological Service for Eskisehir Province in Turkey. There are six Excel files, each file represents a particular variable from the ensemble weather data. All the used variables are listed as in Table 3.1.

Table 3.1. *Dataset variables from weather model*

Date	SR_time	max_temp	max_Ws	Direction	DA_Humidity	Solar radiation
1/1/2011	0.7	3.5	4.7	97	91.1	82339
1/2/2011	0.8	0.7	7.5	94	91.2	80733.5
1/3/2011	0	4.4	6.6	90	98.2	49605.5
1/4/2011	0	4.3	5.1	283	95.4	36796.5
1/5/2011	0	3.5	5.3	219	93.9	28546.9
1/6/2011	0	2.9	6.5	301	89.4	27612.8
1/7/2011	0.5	4.1	6.5	8	84.1	75256.2
1/8/2011	7.2	6.9	4.3	219	71.9	188280.4
…	…	…	…	…	…	…

12/30/2014	0.0	9.5	7.4	93	76.6	43005.7
12/31/2014	0.0	1.2	5.3	149	82.2	40840.9

The first column represents the total number of days during which the weather data are collected. The next columns represent solar radiation time, maximum temperature, maximum wind speed, wind direction, and solar radiation respectively.

3.3. Initialization

According to the literature, bad weight initialization can be the main reason for vanishing gradients which harms the BP training of the neural networks with more than two hidden layers, such as in the DNNs. Inaccurate initializations appear in the gradients centered at 0 and their difference is decreased as one moved from the output to the input layers. Consequently, the network is attached to the weights which are considered as the main cause of the near-zero gradient demeanor in the first layers and make the network incapable of any further learning (Glorot & Bengio, 2010). To avoid vanishing gradients, (Glorot & Bengio, 2010) proposes a simple way to initialize DNN weights and this involves using a normalized hyperbolic tangent activation functions and designing the initial values of the weights from a steady and regular distribution $U\left[-\frac{\sqrt{3}}{\sqrt{M}},\frac{\sqrt{3}}{\sqrt{M}}\right]$. Therefore, the (linear) activations and (nonlinear) outputs of the neurons are located in the interval between [-1, 1] active range of the hyperbolic tangent. As it is mentioned in (Glorot & Bengio, 2010) that the initial weights w_i verify:

$$M_i Var(\omega_i)=1; \quad M_{i+1} Var(\omega_i)=1 \tag{3.1}$$

where; M_i & M_{i+1} are fan-in and fan-out the units in the i^{th} layer. One can achieve that $Var(z_i)\cong Var(z_j)$ across the successive z_j layers and also that $Var(\partial J/\partial z_i)\cong Var(\partial J/\partial z_k)$ for the preceding z_k layers, where J denotes the MLP cost function.

Particularly, the diversities of backpropagated gradients still stabilized and the appearance of vanishing gradients will not exist. In this thesis, ReLU activation function is used instead of the hyperbolic tangent ones. The different types of activation functions are discussed in detail in Table 3.3.

3.4. Data Standardization

Input data values (solar radiation time, maximum temperature, maximum daily wind speed, wind direction, daily humidity) represent various physical values, and as a result, the scale of these different variables changes. To obtain more accurate results from DNN models, the features data must have a similar or same scale. The standardization process can be calculated as follows:

$$x_{ij}^{norm.} = \frac{x_{ij} - \mu_j}{\sigma_j}, \forall i, j \tag{3.2}$$

where; $x_{i,j}$ is the value of the j^{th} feature in the i^{th} sample, μ_j is the mean value of the j^{th} feature in the training set, σ_j is the standard deviation of the j^{th} feature in the training set.

MinMaxScaler function is used here for data standardization in Python TensorFlow library. Principal function of MinMaxScaler is scaling every feature to a given range. The range can be defined as the variation between the original maximum and original minimum value in the data. The shape of the original distribution is preserved by the range, which does not meaningfully change the information embedded in the original data. Figure 3.1 shows the code of data standardization in the Google Colab environment.

```
[ ]  scaler = MinMaxScaler() # For normalizing dataset
     MinMaxScaler(copy=True, feature_range=(0, 1))

     MinMaxScaler(copy=True, feature_range=(0, 1))

[ ]  X_train = scaler.fit_transform(training_X.as_matrix())
     print(X_train)
     X_test = scaler.fit_transform(validation_X.as_matrix())
     print(X_test)

     [[0.05511811 0.24842105 0.15753425 0.26890756 0.88534107]
      [0.06299213 0.18947368 0.34931507 0.2605042  0.88679245]
      [0.         0.26736842 0.28767123 0.24929972 0.98838897]
      ...
      [0.         0.52842105 0.05479452 0.41456583 0.28447025]
      [0.         0.47578947 0.52054795 0.45098039 0.39767779]
      [0.         0.19157895 0.26712329 0.82352941 0.79970972]]
     [[0.         0.04712042 0.29508197 0.82352941 0.78456592]
      [0.         0.08376963 0.12295082 0.94957983 0.33762058]
      [0.         0.12827225 0.17213115 0.59943978 0.31672026]
      ...
      [0.12598425 0.33246073 0.42622951 0.48739496 0.79581994]
      [0.         0.27486911 0.3852459  0.25770308 0.64630225]
      [0.         0.05759162 0.21311475 0.41456583 0.73633441]]

[ ]  Y_train = scaler.fit_transform(df.drop(['date','SR_time','max_temp','max_Ws','direction','DA_Humidity'],axis=1).head(1097).as_matrix())
     print(Y_train)

     [[0.12655367]
      [0.1235173 ]
      [0.06464712]
      ...
      [0.41899838]
      [0.09868735]
      [0.17615342]]
```

Figure 3.1. *Code of data standardization by using Python TensorFlow library in Google Colab environment*

After training the code, the standardized variables of the weather model are shown in the Table 3.2.

Table 3.2. *Standardized variables of the weather model*

Date	SR_time	max_temp	max_Ws	Wind direction	DA_Humidity	Solar radiation
1/1/2011	0.055	0.248	0.158	0.269	0.885	0.127
1/2/2011	0.063	0.050	0.251	0.261	0.950	0.124

1/3/2011	0.000	0.312	0.221	0.250	1.023	0.076
1/4/2011	0.000	0.305	0.171	0.785	0.994	0.057
1/5/2011	0.000	0.248	0.178	0.607	0.978	0.044
1/6/2011	0.000	0.206	0.218	0.834	0.931	0.042
1/7/2011	0.039	0.291	0.218	0.022	0.876	0.116
1/8/2011	0.567	0.490	0.144	0.607	0.749	0.289
1/9/2011	0.283	0.461	0.215	0.266	0.888	0.200
1/10/2011	0.031	0.291	0.171	0.305	0.952	0.119
1/11/2011	0.000	0.312	0.090	0.421	0.971	0.044
1/12/2011	0.087	0.688	0.137	0.704	0.860	0.146
1/13/2011	0.031	0.433	0.154	0.274	0.926	0.150
…	…	…	…	…	…	…
12/30/2014	0.000	0.674	0.248	0.258	0.798	0.066
12/31/2014	0.000	0.085	0.178	0.413	0.856	0.063

In the Table 3.2 features (input) and targets (output) data are represented after standardization by using MinMaxScaler function. The data values are normalized to the values between 0 and 1.

3.5. Global Solar Radiation Estimation

3.5.1. Solar radiation estimation on horizontal surface by using classic mathematical models

There are a lot of models are used for the hourly global radiation prediction by using DGSR values, which are important for solar energy plants. As it is well known that the generated energy from the PV panel is mainly affected by solar radiation (Akarslan et al., 2014). In this thesis, hourly global solar radiation models included in all groups are carried out to predict the DGSR data on a horizontal surface for Eskişehir city. Measured hourly global solar radiation values of the 2014 year, are recorded. DGSR data is calculated by performing five different classic mathematical models (Ayvazoğluyüksel & Filik, 2018).

3.5.1.1. CPR model

CPR model is considered as one of the best classic mathematical models for solar radiation forecasting. This model is expressed in equation (3.3) as reported in the literature (Collares-Pereira & Rabl, 1979). This model expresses that global solar radiation directly depends on the hour angle with ignoring the atmospheric effect. CPR model is developed by using hourly radiation data collected from four US stations for two years. The model is significantly used for DGSR prediction and clearly described in the following:

$$\frac{I}{H}=\frac{\pi(a+b\cdot\cos W)}{24}\frac{\cos W-\cos W_s}{\sin W_s-\frac{\pi\cdot W_s}{180}\cos W_s} \tag{3. 3}$$

$$a=0.4090+0.5016\sin\left(W_s-60^\circ\right) \tag{3.4}$$

$$b=0.6609+0.4767\sin\left(W_s-60^\circ\right) \tag{3.5}$$

$$W=\frac{360\cdot\left(t_s-12\right)}{24} \tag{3.6}$$

$$W_s=\arccos\left(-\tan\delta\tan\phi\right) \tag{3.7}$$

$$\begin{aligned}\delta=&\frac{180}{\pi}\cdot(0.0069\text{-}0.3999\ \cos(\Gamma)+0.070\sin(\Gamma)-0.0067\cos(2\cdot\Gamma)+\\&+0.0009\sin(2\cdot\Gamma)\text{-}0.0026\cos(3\cdot\Gamma)+0.0015\sin(3\cdot\Gamma)\end{aligned} \tag{3.8}$$

$$\Gamma=\frac{2\pi(d-1)}{365} \tag{3.9}$$

where; W_s is the sunrise hour angle, W is the solar hour angle, a & b are linear functions of (W_s-°60), I is the hourly global solar radiation on horizontal surface (W/m^2), H is the DGSR on horizontal surface (W/m^2), I/H is the ratio of hourly to DGSR on horizontal surface, k_t is the hourly clearness index, K_t is the daily clearness index, h is the solar altitude angle (degree), S_0 is the day length (hour), t_s is the solar time (hour), Γ is the day angle (radians), d is the day number starting from the first of January, δ is the solar declination angle (degree), ϕ is the latitude (degree).

3.5.1.2. CPRG model

The adaptability and high performance of the CPR model are assured. To guarantee uniformity during the renormalization, an insignificant adjustment to the CPR model is suggested in (Gueymard, 1986). The following correlations expressed the simple mathematical form of this model:

$$\frac{I}{H}=\frac{360\cdot(a+b\cdot\cos w)r_0}{f_c} \tag{3.10}$$

$$f_c = a + 0.5 \cdot b \cdot \frac{\frac{\pi W_s}{180} - \sin W_s \cdot \cos W_s}{\sin W_s - \frac{\pi W_s}{180} \cdot \cos W_s} \tag{3.11}$$

$$r_0 = \frac{\pi}{24} \cdot \frac{\cos W - \cdot \cos W_s}{\sin W_s - \frac{\pi W_s}{180} \cdot \cos W_s} \tag{3.12}$$

where; f_s and r_0 are multiplying factor.

3.5.1.3. Liu and Jordan model

In Liu and Jordan's way is proven by confirming that hourly solar radiation can be predicted depending on the hour angle and the average DGSR directly. To use the Liu and Jordan model perfectly, the total amount of solar radiation per hour, including direct and widespread radiations is required to determine the long-term average hourly and daily totals of the widespread radiations (Liu & Jordan, 1960). Subsequently, the proportion of hourly to DGSR function is moderately simplified to be in the following form:

$$\frac{I}{H} = \frac{\pi}{24} \frac{\cos W - \cos W_s}{\sin W_s - \frac{\pi \cdot W_s}{180} \cos W_s} \tag{3.13}$$

3.5.1.4. Jain model 1 & Jain model 2

The average data of eleven-years which reference the proportion of hourly to DGSR, drown for every month relying on the solar time in that time (Jain, 1984). The average of solar radiation is taken at solar noon, and σ data are calculated for every month by convenient the experimental and theoretical data at solar noon. The calculated σ data have a good linear relation with S_0. The σ values are calculated as follows:

$$\sigma = 0.192 \cdot S_0 + 0.461 \tag{3.14}$$

$$S_0 = \frac{2}{15} W_s \tag{3.15}$$

where; S_0 is defined as in equation (3.15). As stated in (Nik et al., 2012), a Gaussian function is suggested to fit the collected data in the model. This provides an easy method to predict DGSR from hourly values. The method makes the prediction of global solar radiation applicable in any short time intervals as well.

The Jain model 2 developed by Jain, P. C. (Jain, 1984) is moderately adjusted to be able to be used to predict hourly radiation values in any place (Jain, 1988). A different σ value is determined to be used in (3.17) as:

$$\sigma = 0.2 \cdot S_0 + 0.378 \tag{3.16}$$

For DGSR forecasting calculation, this model suggests the correlation below to be used as described in (3.17).

$$\frac{I}{H} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{(t_s-12)^2}{2\sigma^2}\right)} \tag{3.17}$$

3.5.2. Solar radiation prediction by using ANN model

ANNs are considered as a processing system that connects small processing units. ANN in shape and work simulates the shape and the work of the biological neural network in the human body. ANNs can automatically learn to recognize patterns in data from the previous data that can be added to the network (Kalogirou, 2001). The ability to model complicated, non-linear processes without the need to suppose the shape of the correlation between input and output variables is considered as the best advantage of ANN that cannot be found in the other forecasting techniques.

For solar radiation prediction by using the ANN model, the Visual Gene Developer software is used. For many reasons, such as this software includes an ANN toolbox and it is easy to be used. Also, users at the time of developing new modules can access to the class to use neural network prediction toolbox; because the software provides a specialized class which is 'NeuralNet'. For instance, any user can use maximum 5 instances of NeuralNet including 'NeuralNet', 'NeuralNet2', 'NeuralNet3', 'NeuralNet4', and 'NeuralNet5'.

In this software, a typical feedforward neural network (FFNN) with a standard BP learning algorithm is used to train networks and supplies several various transfer functions. Without using gene design or optimization, this neural network package works in a perfect manner autonomously even though all menus are still in the software environment (http-2).

Prediction with the use of this software is achieved by using different ANN structures with different parameters. The best model gives accurate results is ANN with

3 hidden layers besides input and output layers. The shape of this model is clearly shown in Figure 3.2.

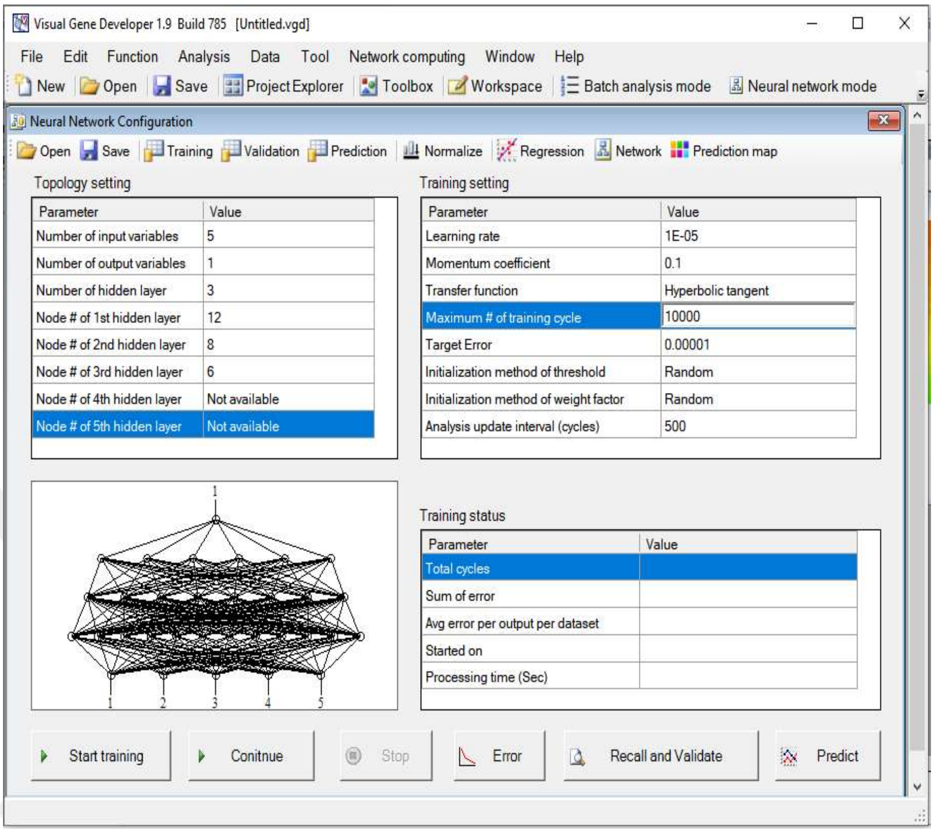


Figure 3.2. *Solar radiation prediction using Visual Gene Developer software ANN window*

3.5.3. Estimation solar radiation by using DNN model

DNNs are used in image and speech recognition competitions, translation and e-marketing (Díaz et al., 2017). DNNs have a fundamental one- or two-dimensional structure that can be captured and exploited by conventional layers. As a result, the forward development of the Internet of Things (IoT) and increasing Big Data capacity, DL models are recently taking large attention to be used in diverse research fields.

In this thesis:

- The DNN forecasting model based on five different weather features, is created;

- The daily solar radiation for Eskisehir city, one year ahead is forecasted;

- The results of DNNs and the results of the different forecasting models that are realized by the classical methods and ANNs method are compared and analyzed.

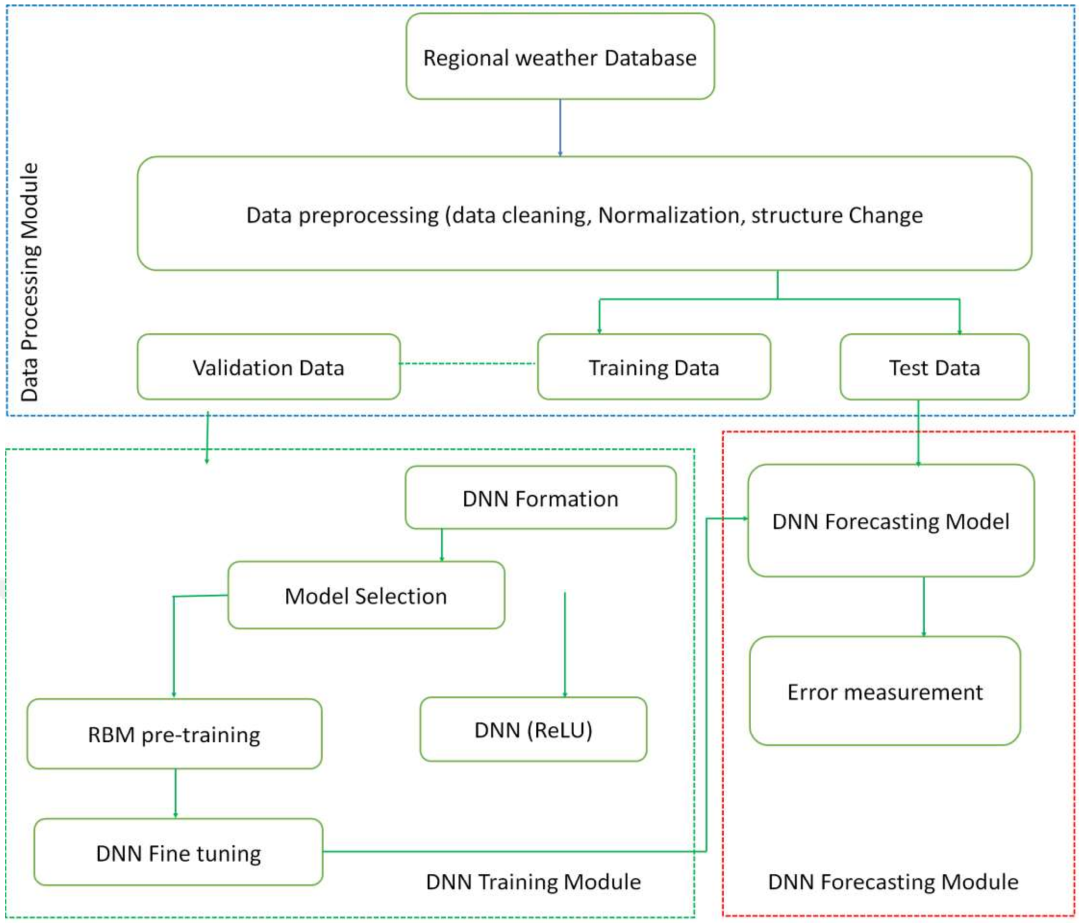


Figure 3.3. *The proposed DNN solar radiation forecasting framework (Ryu et al., 2017)*

DNN model's structure

This part shows the architecture of the DNN solar radiation prediction framework. DNNs have the potential to give us the nonlinear correlation between input data 'features' and the output data 'targets'. It takes on learning exemplifications from data that put a confirmation on learning sequential layers of progressively significant exemplifications. The deeper it goes, the higher-level representation it is able to identify, so that it establishes a proper mapping from input features to their target. The proposed network is a DNN consisting of 1 input layer with 5 input values, 1 output layer with 1 output value, and 5 hidden layers. The designed network is shown below as in Figure 3.4.

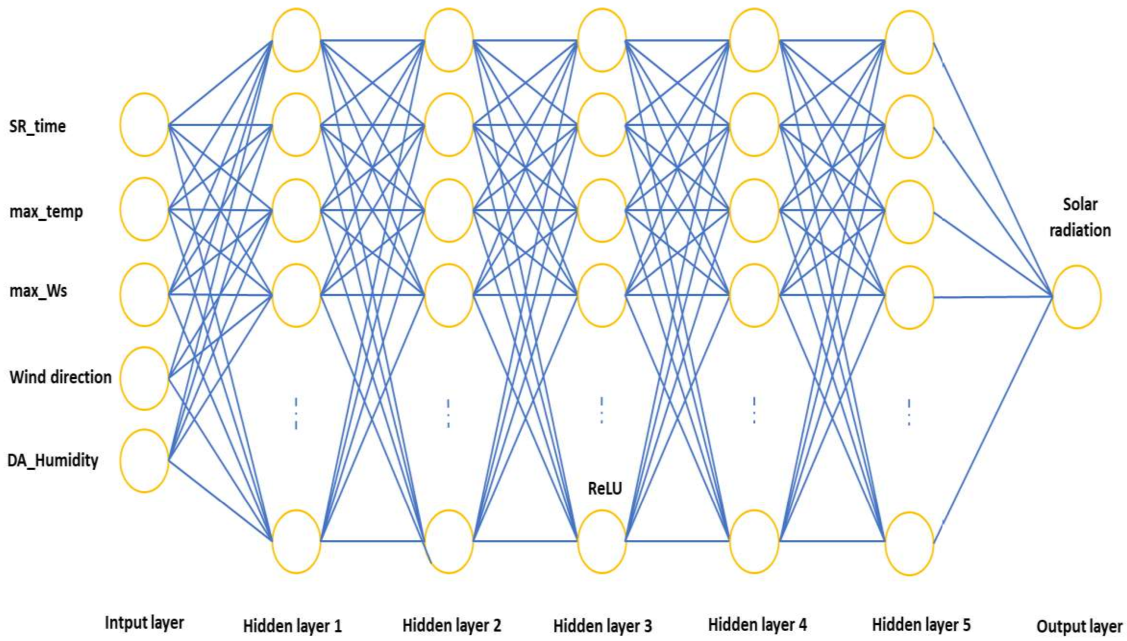


Figure 3.4. *The general structure of the proposed DNN*

The structure of the DNN model in Google Colab environment is shown in the Figure 3.5. In this case, batch size needs to be specified during the training, and the number of features shall be 5, due to the input shape, the input layers shall contain five nodes, each of them represents one variable of the weather parameters. The first hidden layer contains 265 nodes, which is sufficiently large to catch all of the data uploaded to the input layer. For the next hidden layers, the numbers of nodes in them are 265, 265, 185 and 95, respectively. In the output layer, there is only one node since for the regression problem.

```
[ ] #Define the important parameters
    NN_model = Sequential()
    NN_model.add(Dense(5, kernel_initializer='normal',input_dim = X_train.shape[1], activation='relu')) #Input layer
    NN_model.add(Dense(265, kernel_initializer='normal',activation='relu')) #Hidden layers
    NN_model.add(Dense(265, kernel_initializer='normal',activation='relu')) #Hidden layers
    NN_model.add(Dense(265, kernel_initializer='normal',activation='relu')) #Hidden layers
    NN_model.add(Dense(185, kernel_initializer='normal',activation='relu')) #Hidden layers
    NN_model.add(Dense(95, kernel_initializer='normal',activation='relu')) #Hidden layers
    NN_model.add(Dense(1, kernel_initializer='normal',activation='linear')) #Output layer

    #Compile the network
    optimizer = tf.keras.optimizers.RMSprop(0.001)
    NN_model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mean_absolute_error','mean_absolute_percentage_error','cosine_proximity','mean_squared_error'])
    NN_model.summary()


    #Define a checkpoint callback
    checkpoint_name = 'Weights-{epoch:03d}--{val_loss:.5f}.hdf5'
    checkpoint = ModelCheckpoint(checkpoint_name, monitor='val_loss', verbose = 1, save_best_only = True, mode ='auto')
    callbacks_list = [checkpoint]
```

Figure 3.5. *The structure of neural network in Python "colab.research.google.com"*

FFNN

Feedforward where data can only move in one way in one direction layer by layer, is considered as the first step during the training process in the ANNs. The concept of the FFNN is not complex: the network is received data. Then FFNN learns from the input data and gives the wished output results for every data sample. The weights, which can be in a type of numbers, vectors, or matrices, parameterize the input information that are inserted to the layers. The process between the layers involves the addition of the output times weights of each node from the previous layer and sending them to each node of the next layer. This can be mathematically expressed as below:

$$y_j = \sum_{i=1}^{n} \left(w_{ij} x_i + b_j \right) \tag{3.18}$$

where; x_i is the feature of the input layer (also x_i can be the output from the previous layer), n is the number of nodes in the previous layer, w_{ij} is the weight, b_i is the bias term. The output y_j can be considered as the input value for the node in the subsequent layer.

Activation function

The process inside the nodes is called activation process, where the activation function is enforced to the node input. As known, every synapse in any ANN model has its own weight w_{ij}, which is multiplied by its input value x_i before summing all weighted inputs as well as an externally bias b_j which is responsible for lowering or increasing the summation's output v_j, as can be seen in Figure 3.6.

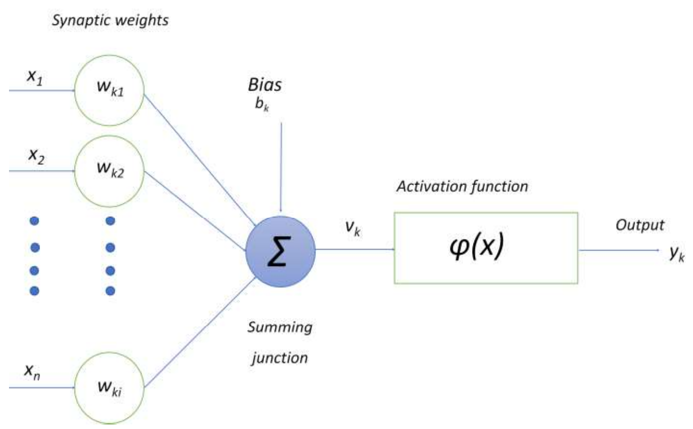
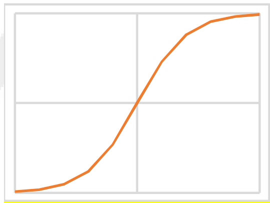
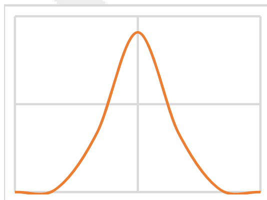
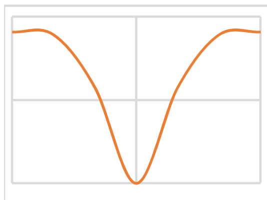
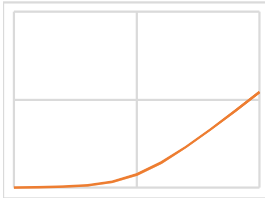
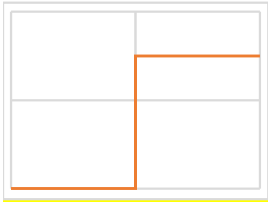


Figure 3.6. *The general structure of the activation function*

Activation function applied to the output layer, has essential role in minimizing and modifying the amplitude zone of the output data y_j into an exact value. There are many types of activation functions, and some of them are mentioned in the Table 3.3.

Table 3.3. *Different types of activation functions (Nwankpa et al., 2018)*

Function type	Description	Equation	Graph
Linear	Considered as one of the simple activation functions in which the output is in a linear relationship with the input	$f(x) = x$	
Sigmoid	Considered as one of S-shaped family functions. The reason of using sigmoid function more is because of its existence in the interval between (0 to 1). So, it is the best choice for prediction the probability problems.	$f(x) = \frac{1}{1+e^{-x}}$	
Gaussian	Gaussian activation function is a function that has a continuous bell-shaped curve.	$f(x) = e^{-x^2}$	
ReLU	Rectified linear function it is an activation function that can be used for each node in hidden layers	$f(x) = \begin{cases} 0, for\ x < 0 \\ x, for\ x \geq 0 \end{cases}$	
Gaussian complement	It is a reflection of the Gaussian function.	$f(x) = 1 - e^{-x^2}$	

Softplus	Softplus AF known as a smooth version of ReLU function. This kind of AF is commonly used in speech recognition systems.	$f(x)=\log_e(1+e^x)$	
Binary step	Binary step activation function is considered as a threshold-based activation function. This kind of activation function cannot be used for classifying problems.	$f(x) = \begin{cases} 0 \; for \; x < 0 \\ 1 \; for \; x \geq 0 \end{cases}$	

To introduce non-linearity into the network, activation functions are used, to make each layer able to take high-level representations out from the input data. ReLU function is one of the activation functions that are used for each node in the hidden layers. The activation function as mentioned before is enforced to the node input in the activation process. The most commonly used activation function in DNN models is the ReLU activation function, which can be written as follows:

$$\mathrm{Re}\,LU = f(x) = \begin{cases} 0 & , for \quad x < 0 \\ x & , for \quad x \geq 0 \end{cases} \tag{3.19}$$

With this choice, only partial nodes in each layer will be activated i.e. not being squashed into a certain range, which is widely used for the regression problems. The derivative of the ReLU function is written as follows:

$$\frac{d}{dx}\mathrm{Re}\,LU = f'(x) = \begin{cases} 0 & , for \quad x < 0 \\ 1 & , for \quad x \geq 0 \end{cases} \tag{3.20}$$

The derivative of ReLU function is continuous and can be easily computed to be suitable for BP process.

Modeling DNN model

The training procedure can be summarized as follows: firstly, the important functions and libraries which are used in this work are determined and imported to the Google Colab environment. The collected data are imported to the Google Colab environment and divided into two main groups; training data set and testing data set. As it is shown in Figure 3.7, the network structure and training parameters are selected after which the model is trained using the training data set. Then, the model is evaluated by using the testing data set. Finally, the last three steps are repeated using various training parameters until the final training parameters are found.

```
def construct_feature_columns(X_data):
  return set([tf.feature_column.numeric_column(my_feature)
              for my_feature in X_data])
```

```
#Define the important parameters
NN_model = Sequential()
NN_model.add(Dense(5, kernel_initializer='normal',input_dim = X_train.shape[1], activation='relu')) #Input layer
NN_model.add(Dense(265, kernel_initializer='normal',activation='relu')) #Hidden layers
NN_model.add(Dense(265, kernel_initializer='normal',activation='relu')) #Hidden layers
NN_model.add(Dense(265, kernel_initializer='normal',activation='relu')) #Hidden layers
NN_model.add(Dense(185, kernel_initializer='normal',activation='relu')) #Hidden layers
NN_model.add(Dense(95, kernel_initializer='normal',activation='relu')) #Hidden layers
NN_model.add(Dense(1, kernel_initializer='normal',activation='linear')) #Output layer

#Compile the network
optimizer = tf.keras.optimizers.RMSprop(0.001)
NN_model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mean_absolute_error','mean_absolute_percentage_error','cosine_proximity','mean_squared_error'])
NN_model.summary()


#Define a checkpoint callback
checkpoint_name = 'Weights-{epoch:03d}--{val_loss:.5f}.hdf5'
checkpoint = ModelCheckpoint(checkpoint_name, monitor='val_loss', verbose = 1, save_best_only = True, mode ='auto')
callbacks_list = [checkpoint]
```

Third : Train the model :

```
history = NN_model.fit(X_train, Y_train, epochs=1000, batch_size=64, validation_split = 0.2, callbacks=callbacks_list)
```

```
# Load wights file of the best model :
wights_file = 'Weights-642--0.09143.hdf5' # choose the best checkpoint
NN_model.load_weights(wights_file) # load it
NN_model.compile(loss='mean_absolute_error', optimizer='adam', metrics=['mean_absolute_error'])
```

Fourth : Test the model

We will submit the predictions on the test data to Kaggle and see how good our model is.

```
def make_submission(prediction, sub_name):
  my_submission = pd.DataFrame({'Id':Y_test.Id,'solar_radiation':prediction})
  my_submission.to_csv('{}.csv'.format(sub_name),index=False)
  print('A submission file has been made')
```

```
predictions = NN_model.predict(X_test)
```

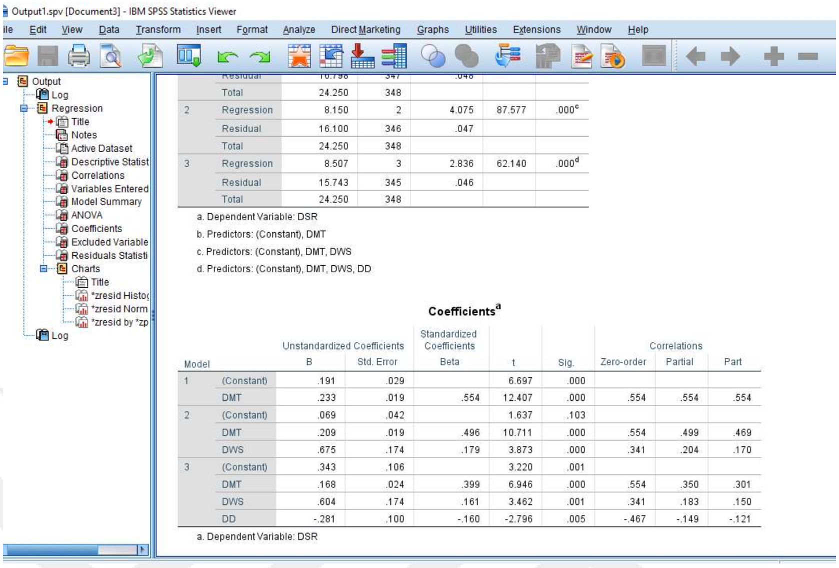
```
print (predictions)
```

Figure 3.7. *Structure of DNN model in Python "colab.research.google.com"*

DNN model is created and trained in Python programming language by using DL framework called Keras library. To train the data, as mentioned earlier, the data normalization process is required. By this way, each feature in the raw data is normalized to have a mean value of 0 and standard deviation of the target values. Daily solar energy attain its original distribution, but is being rescaled to have a reasonable magnitude range. During the training, the input data are divided into four parts. The first three parts are used and trained as a training set while the last fourth part is used and trained as a validation or test set. The network as mentioned above is trained by using the training data set and is tested by using the testing data set. This process is repeated four times as the model has been tested on every single part of the 4 split parts.

3.5.4. Developing new models for solar radiation forecasting

By using IBM SPSS software which is used for interacting and impulsed statistical analysis (Box et al., 2015), four different models are created. These models are developed for solar radiation estimation for Eskisehir city, by using predicated solar radiation data in the period between 01.01.2014 until 31.12.2014. In the Figure 3.8 the results of the first developed 3 models for Eskisehir city by using IBM SPSS software are shown.



Model		Sum of Squares	df	Mean Square	F	Sig.
	Total	24.250	348			
2	Regression	8.150	2	4.075	87.577	.000[c]
	Residual	16.100	346	.047		
	Total	24.250	348			
3	Regression	8.507	3	2.836	62.140	.000[d]
	Residual	15.743	345	.046		
	Total	24.250	348			

a. Dependent Variable: DSR
b. Predictors: (Constant), DMT
c. Predictors: (Constant), DMT, DWS
d. Predictors: (Constant), DMT, DWS, DD

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Correlations		
		B	Std. Error	Beta			Zero-order	Partial	Part
1	(Constant)	.191	.029		6.697	.000			
	DMT	.233	.019	.554	12.407	.000	.554	.554	.554
2	(Constant)	.069	.042		1.637	.103			
	DMT	.209	.019	.496	10.711	.000	.554	.499	.469
	DWS	.675	.174	.179	3.873	.000	.341	.204	.170
3	(Constant)	.343	.106		3.220	.001			
	DMT	.168	.024	.399	6.946	.000	.554	.350	.301
	DWS	.604	.174	.161	3.462	.001	.341	.183	.150
	DD	-.281	.100	-.160	-2.796	.005	-.467	-.149	-.121

a. Dependent Variable: DSR

Figure 3.8. *The results of the of creating model (3.21-3.23) in SPSS*

These models are developed by using standard multiple regression method. The results are expressed in the form of SPSS Model 1, SPSS Model 2, SPSS Model 3 as shown in (3.21 - 3.23):

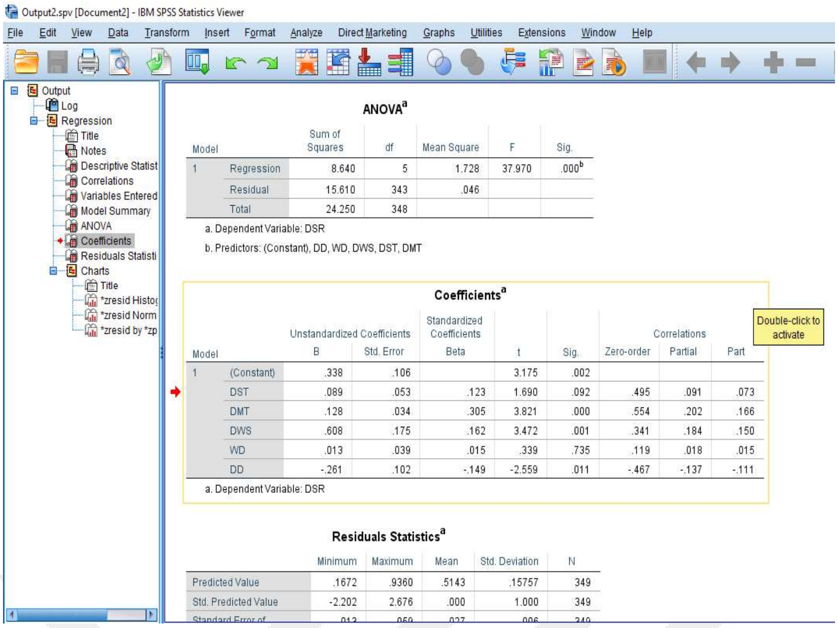
$$SPSS.Model1.(SR) = 0.191 + 0.233 \cdot T_{max} \tag{3.21}$$

$$SPSS.Model2.(SR) = 0.069 + 0.209 \cdot T_{max} + 0.675 \cdot W_{speed} \tag{3.22}$$

$$SPSS.Model3.(SR) = 0.343 + 0.168 \cdot T_{max} + 0.604 \cdot W_{speed} - 0.281 \cdot H \tag{3.23}$$

where; *SPSS.Model.n.(SR)* is the solar radiation function, T_{max} is the daily maximum temperature, W_{speed} is the daily wind speed, H is the humidity, SR_t is the solar radiation time.

In the Figure 3.9, the results of developed forth model is shown.



ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	8.640	5	1.728	37.970	.000[b]
	Residual	15.610	343	.046		
	Total	24.250	348			

a. Dependent Variable: DSR

b. Predictors: (Constant), DD, WD, DWS, DST, DMT

Coefficients[a]

Model		Unstandardized Coefficients B	Std. Error	Standardized Coefficients Beta	t	Sig.	Correlations Zero-order	Partial	Part
1	(Constant)	.338	.106		3.175	.002			
	DST	.089	.053	.123	1.690	.092	.495	.091	.073
	DMT	.128	.034	.305	3.821	.000	.554	.202	.166
	DWS	.608	.175	.162	3.472	.001	.341	.184	.150
	WD	.013	.039	.015	.339	.735	.119	.018	.015
	DD	-.261	.102	-.149	-2.559	.011	-.467	-.137	-.111

a. Dependent Variable: DSR

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	.1672	.9360	.5143	.15757	349
Std. Predicted Value	-2.202	2.676	.000	1.000	349

Figure 3.9. *The results of the of creating model (3.24) in SPSS*

Developing of this model is obtained by stepwise multiple regression method. The results are expressed in the form of SPSS Model 4 as shown in (3.24):

$$SPSS.Model4.(SR) = 0.338 + 0.089 \cdot SR_t + 0.128 \cdot T_{max} + 0.608 \cdot W_{speed} + \\ + 0.013 \cdot W_{direction} - 0.261 \cdot H \qquad (3.24)$$

3.6. Performance Evaluation

By using statistical analysis methods, the predicted solar radiation values are compared with those which are predicted. Consequently, the work of the models used for the solar radiation prediction is evaluated to obtain and to determine the best model. The most commonly used methods are as following: RMSE, MABE, MBE and standardized test statistic (t-statistic), which are described in detailed as following:

RMSE

RMSE is considered as a measure which is frequently used to evaluate predicted values compared to real values. In other meaning, it tells how the predicted points are away from the regression line. RMSE is generally used in climatology, and regression analysis to confirm experimental results. The RMSE can be calculated by the given function:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}{n}} \tag{3.25}$$

where; n is the number of samples, y_i is the real or measured value of any sample i, $\bar{y}_i$ is the predicted value of the sample i.

MABE

The performance of the regression analysis is assessed by MAE, which indicates how much the predicted values deviates from the real values. The MAE can be calculated by given function:

$$MABE = \frac{\sum_{i=1}^{n}|y_i - \bar{y}_i|}{n} \tag{3.26}$$

MBE

Usually MBE is not used as a measure of the model error because of its high individual errors in prediction. The main use of MBE is to estimate the average bias in the model and deciding if any steps need to be taken to correct the model bias. The MBE can be calculated by given function:

$$MBE = \frac{\sum_{i=1}^{n}(y_i - \bar{y}_i)}{n} \tag{3.27}$$

Standardized t-statistic

t-statistic is considered as one of the most important statistic tools that can be used in statistical hypothesis testing. It can be used to compare data with what is expected under the null hypothesis. And to make *p*-values calculating easier, t-statistic sampling distribution must be calculable accurately or approximately, and this is considered as an important feature of the t-statistic. t-statistic can be calculated as:

$$t-statistic = \sqrt{\frac{(n-1)MBE^2}{RMSE^2 - MBE^2}} \tag{3.28}$$

4. RESULTS AND DISCUSSION

A linear regression model has been implemented as a model for DGSR prediction in the study. It is a linear approach for modeling the correlation between a scalar dependent variable *y*, which is the solar radiation in this case, and five independent variables denoted x_i as mentioned before. The linear regression model is shown as the following:

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + b \tag{4.1}$$

where; *y* is the target vector, x_i is the features vector, β_i is the weights matrix and *b* is the bias vector, $i=5$ variables in this case.

In this work solar radiation data for Eskisehir city are predicted by using classic mathematical methods, ANN and DNN models. Classic mathematical models are CPR, CPRG, Liu and Jorden Lian Model 1 and Lain Model 2. According to (Ayvazoğluyüksel & Filik, 2018) study, in this work these 5 models give the best results in the Eskişehir region. For this reason these methods are chosen. The simulation and running these models is achieved by MS excel 2019. ANN model is trained by using Visual Gene Developer software. DNN model is trained by the TenserFlow Python library in the Google Colab environment. DNN model consists of one input layer contains 5 parameters, 5 hidden layers, and one output layer. DNN model is trained and tested using the data collected by the meteorology station in Eskişehir city. After filling the missed data feature, vector of each sample is reshaped into a 5-dimensional vector, and target vector inon one-dimensional vector. Thus, the input layer contains 5 nodes, each node represents one feature of the weather parameters and the output layer contains only 1 node.

In the Table 4.1 the results of 12 different DNN models are given. The ReLU is applied to each node in the hidden layers and the number of epochs is 1000 for all of the 12 models. It can be seen that the best model with the best performance is the DNN model 5 which consists of one input layer, five hidden layers, and one output layer. DNN model 5 is trained with batch size equal 64 and RMSprop equal 0.001. The number of processed samples before updating the model is called batch data. Batch data also known as the number of the finished processes during the training dataset. Batch size value should be integer between one and the number of the samples in the training dataset. However, the number of epochs can be an unlimited value between one and infinity.

DNN model is considered as the best choice for solar energy forecasting in order to contribute to the management of the electric power grid. In Table 4.1 the different DNN models with their results are shown.

Table 4.1. *The results of different DNN models*

DNN models	Batch size	No of hidden layers	RMSE	MABE	MBE	t-statistic	Optimizers . RMSprop
DNN Model 1	32	3	0.075515	0.036831	0.009143	2.327087	0.0001
DNN Model 2	32	6	0.074397	0.036982	0.009748	2.521588	0.001
DNN Model 3	32	10	0.075992	0.036634	0.013348	3.404251	0.0001
DNN Model 4	32	10	0.070635	0.033558	0.011741	3.215976	0.001
DNN Model 5	64	5	0.064527	0.032926	0.007223	2.149198	0.001
DNN Model 6	32	5	0.073544	0.035625	0.007084	1.846348	0.001
DNN Model 7	64	7	0.07498	0.036299	0.008808	2.256841	0.001
DNN Model 8	32	5	0.073994	0.037796	0.012731	3.332268	0.001
DNN Model 9	64	10	0.077638	0.038292	0.013483	3.364422	0.001
DNN Model 10	32	10	0.080724	0.038046	0.021465	5.262578	0.001
DNN Model 11	64	3	0.076221	0.037232	0.01047	2.645715	0.001
DNN Model 12	32	3	0.083152	0.039493	0.00666	1.53296	0.001

The solar radiation also is predicted by using 11 different models' classic mathematical models, ANNs model and four new developed mathematical models. The developed mathematical models are created by using software package named IBM SPSS, which is used for interactive statistical analyses (Box et al., 2015). The first model is in type of a correlation between solar radiation and maximum temperature. This model and the other models can be seen clearly in (3.21-3.23). The last model (SPSS Model 4) is in type of a correlation between solar radiation and (maximum temperature, solar radiation time, maximum wind speed, wind direction, humidity). The last model can be seen clearly in (3.24). In Table 4.2 it is shown that the last created model SPSS Model 4, which depend on maximum temperature, solar radiation time, maximum wind speed, wind direction and humidity, has the best result compared with the results of the other three models.

The results of DNN, ANN, classic mathematical, and four developed mathematical models are compared to each other. The statistical performance is evaluated by RMSE, MABE, MBE and t-statistic. As can clearly be seen in the Table 4.2 the DNN model 5 and ANN model have the closest results to the real values. The statistical performance of the models are shown in the Table 4.2., DNN model 5 has the lowest value of the MABE, when we compared it with the other models.

Table 4.2. *Statistical performance results of the predicted solar radiation data using different models*

Model	RMSE	MABE	MBE	t-statistic
DNN model 5	0.0645	0.0329	0.0072	2.1491
ANN	0.1147	0.0764	-0.0002	0.0279
Lain Model 2	0.1351	0.0891	0.0610	8.3811
Lain Model 1	0.1356	0.0904	0.0606	9.5320
CPR Model	0.1457	0.0976	0.0662	9.7417
CPRG Model	0.1458	0.0977	0.0662	9.7159
Liu Model	0.1712	0.1504	0.0649	7.8186
SPSS Model 4	0.090	0.046	0.026	5.647
SPSS Model 3	0.099	0.056	0.025	4.914
SPSS Model 2	0.097	0.054	0.025	5.026
SPSS Model 1	0.093	0.049	0.025	5.363

In the following graphs it is clearly shown that the DNN model 5 results has the minimum error value. DNN modeling results of the testing and predicted results are shown in the Figure 4.1. The predicted values basically capture the true target value; however, there are some several differences around turning points, and the trends of some middle points are completely toward different directions. From the Figure 4.1 it can be seen how DNN model gives the most accurate results. In the Figure 4.2, the rate of change versus time between hourly and daily radiation is shown. The result shows that the results obtained by DNN model 5 are the closest ones to the real data.

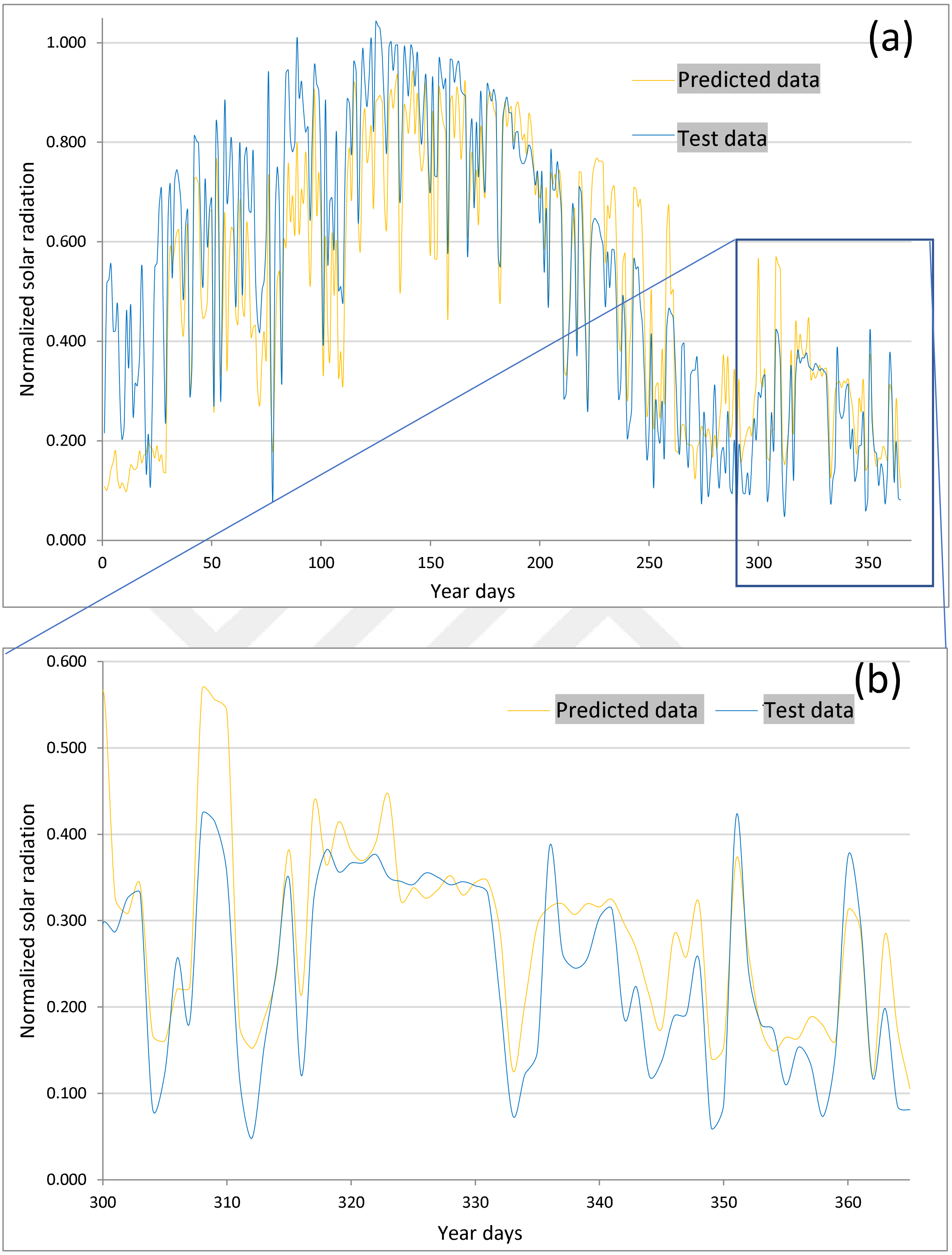


Figure 4.1. *Testing result of multi-linear regression DNN model 5 a) results of one-year forecasting b) zoomed portion of the model results*

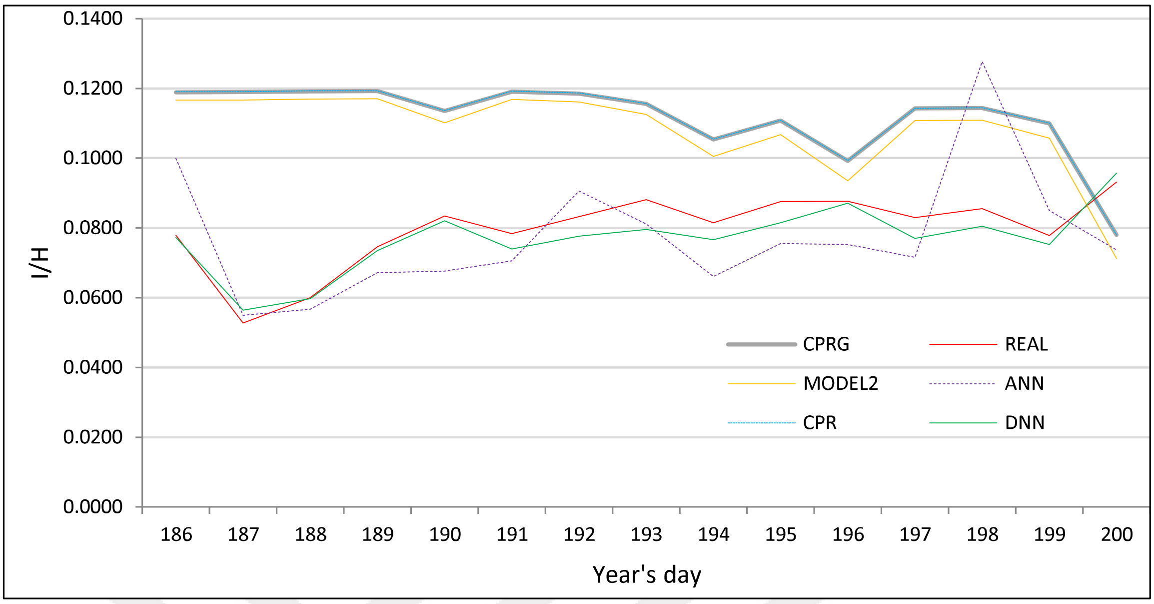


Figure 4.2. *Testing result of linear regression of hourly and daily radiation rate*

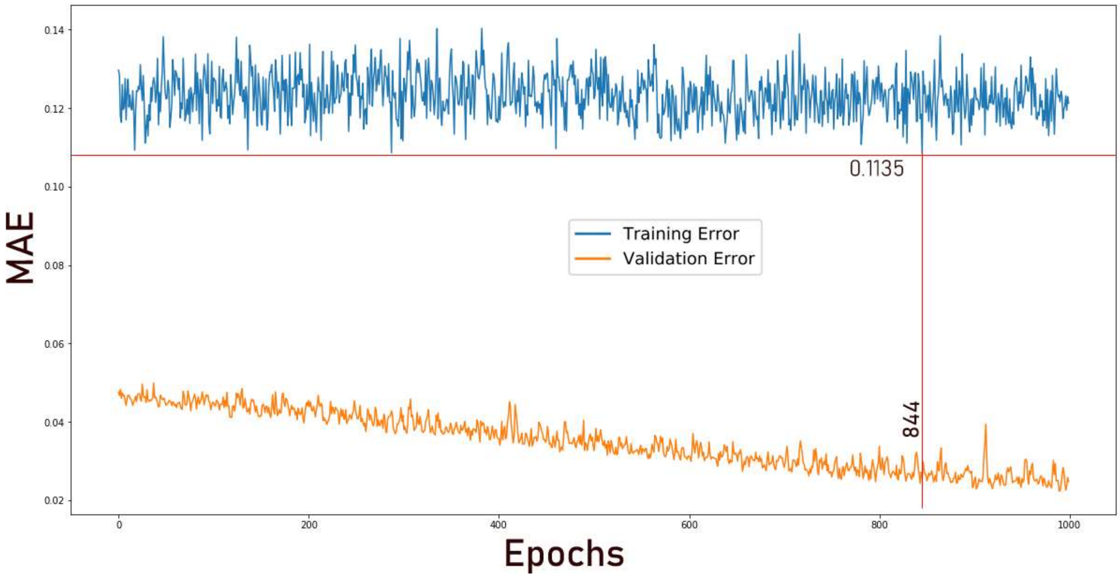


Figure 4.3. *The MAE for training and validation processes of the DNN model 5*

As it is shown in Figure 4.3, the predicted values can capture most of the true values. By using IBM SPSS four different models for solar radiation forecasting have been created. As mentioned before the SPSS Model 4 gives the best results. The prediction results that are obtained by the new developed models can be seen in Figure 4.4.

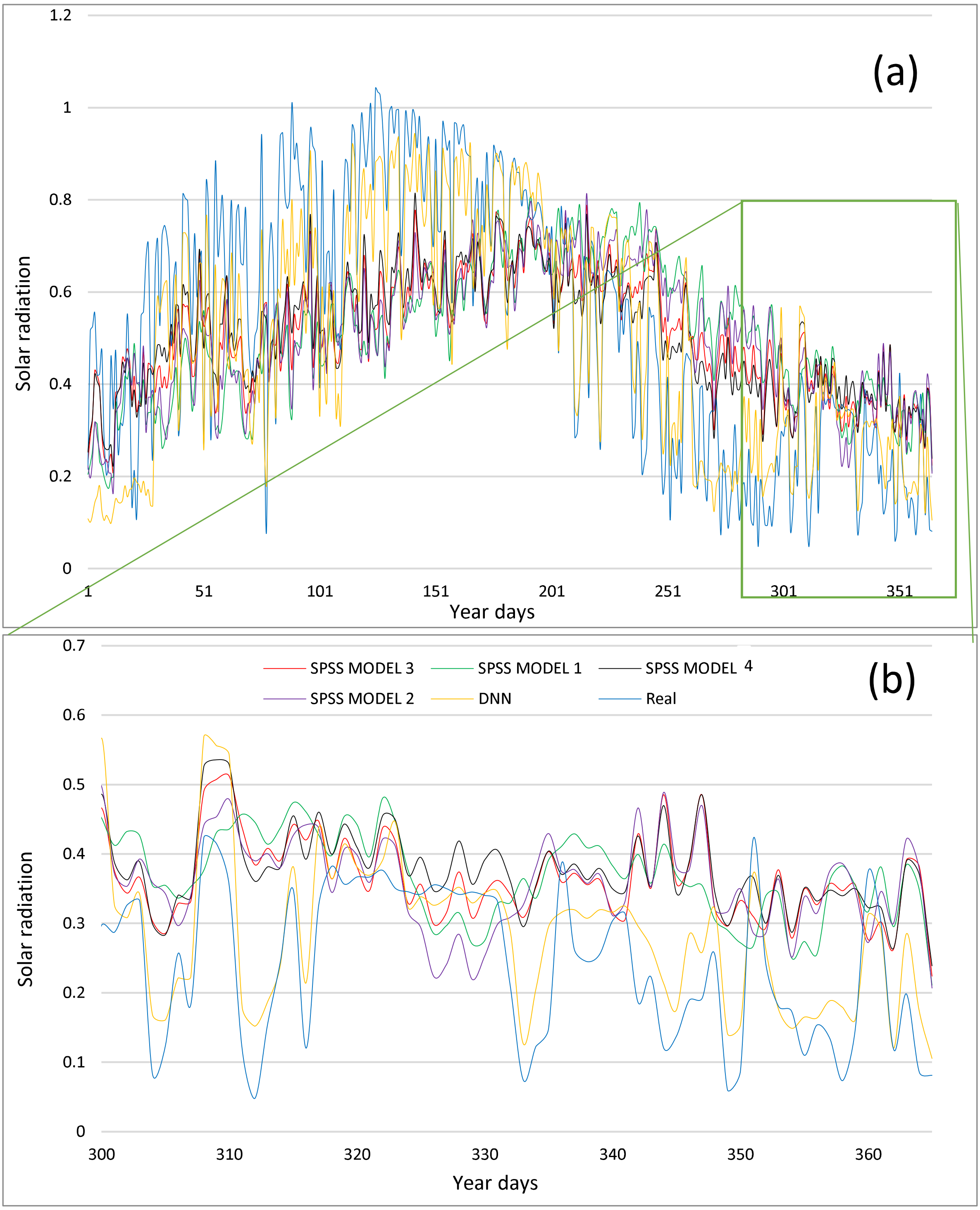


Figure 4.4. *The results of developed SPSS models compared to DNN model 5 results and the real values a) results of one-year forecasting b) zoomed portion of the models result*

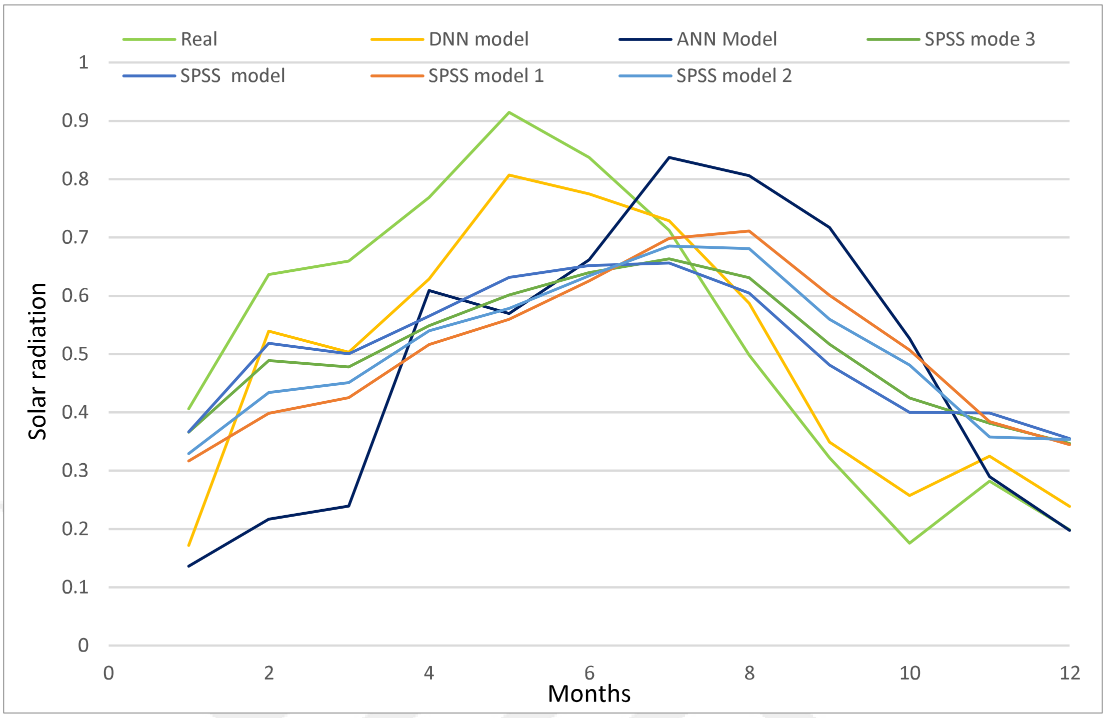


Figure 4.5. *The results of monthly solar radiation of different used models*

In the Figure 4.5 it is shown the average monthly results of solar radiation of different used models. It can be clearly seen that DNN model gives the best results in the models.

5. CONCLUSION

The global energy demand is to increase presumably by 10% from 30% in 2016 to 40% in 2040. This increase in the energy demand is coming from a predictable increase in consumer electricity usage, due to increase in the population around the world in the coming decades. As a result, there is growing concern about environmental pollution which causes greenhouse gases, and cleaner energy technologies are developed. Approximately 65% of the future global investments for creating new power plants will be focused on renewable energy sources by 2040, as projected by the World Economic Outlook (WEO). This clearly shows the extended growth of the worldwide market share of solar energy. Consequently, a near real-time predictive model that observes solar energy conversion effectively and its performance are considered as the best choice for design solar energy plants. The models, in general, give us the statistical relation between solar radiation and the other weather variables.

In this thesis, most of the scientific articles on solar radiation forecasting methods for the last 10 years, are reviewed. Most of the published articles with many experimental results with a large number of data, support using DNN and its superiority are compared with the other methods. Then, the solar radiation for the given location has been predicted by using different models; five classic mathematical models (CPR, CPRG, Lain Model 1, Lain Model 2 and Liu model) and ANN model using Visual Gene Developer software. Also, the same data are predicted by using the previously proposed multilayer DNN. The proposed DNN model contains an input layer, 5 hidden layers, and one output layer. The input layer contains 5 nodes which represent the five weather features on which the solar radiation depends. The output layer has only 1 node and represents predicted solar radiation values. The network is trained and tested using the given data. The statistical performances of all created models are evaluated by RMSE, MABE, MBE and t-statistic. The experiment results show that solar radiation forecasting by using DNN model gives the best result with high accuracy. The proposed model has a testing accuracy of 98 % for daily solar energy forecasting, outperforming the linear regression baseline model of which the testing accuracy is 96.71%.

Furthermore, in order to explore and to create the best model for solar radiation forecasting, based on the DNN model's results, four different models are created by using IBM SPSS software. The model which has the highest accuracy was SPSS model 4 which depends on the solar radiation time, wind maximum speed, wind direction, humidity, and maximum temperature.

Future research could focus on the hybrid models which are considered as complex models. The focus could be on the hybrid models which are combining DNN models with other ANNs models as GAs, FL and WT.

REFERENCES

Akarslan, E., Hocaoğlu, F. O., & R. Edizkan. (2014). A novel MD (multidimensional) linear prediction filter approach for hourly solar radiation forecasting. *Energy*, *73*, 973–986.

Alzahrani, A., Shamsi, P., Dagli, C., & Ferdowsi, M. (2017). ScienceDirect ScienceDirect Solar Irradiance Forecasting Using Deep Neural Networks. *Procedia Computer Science*, *114*, 304–313.

Arya, J. S., Singh, O., & S. K. Aggarwal. (2016). Solar Energy Radiation Forecasting Methods in the Current Scenario.

Atkinson, P. M., & Tatnall, A. R. (1997). Introduction neural networks in remote sensing. *International Journal of Remote Sensing*, *18*(4), 699–709.

Ayvazoğluyüksel, Ö., & Filik, Ü. B. (2018). Estimation methods of global solar radiation, cell temperature and solar power forecasting: A review and case study in Eskişehir. *Renewable and Sustainable Energy Reviews*, *91*(May 2017), 639–653.

Benali, L., Notton, G., Fouilloy, A., Voyant, C., & Dizene, R. (2019). Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. *Renewable Energy*, *132*, 871–884.

Boland, J. (1995). Time-series analysis of climatic variables. *Solar Energy*, *55*(5), 377–388.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. *John Wiley & Sons*.

Cao, J. C., & Cao, S. H. (2006). Study of forecasting solar irradiance using neural networks with preprocessing sample data by wavelet analysis. *Energy*, *31*(15), 3435–3445.

Cao, S., & Cao, J. (2005). Forecast of solar irradiance using recurrent neural networks combined with wavelet analysis. *Applied Thermal Engineering*, *25*(2), 161–172.

Chaturvedi, D. K. (2016). Solar Power Forecasting : A Review, *145*(6), 28–50.

Chau, K. (2006). A review on the integration of artificial intelligence into coastal modeling. *Journal of Environmental Management*, *8*(1), 47–57.

Collares-Pereira, M., & Rabl, A. (1979). The average distribution of solar radiationcorrelations between diffuse and hemispherical and between daily and hourly insolation values. *Solar Energy*, *22*(2), 155–164.

Conti, J., Holtberg, P., Diefenderfer, J., LaRose, A., Turnure, J. T., & Westfall, L. (2016). International energy outlook 2016 with projections to 2040 (No. DOE/EIA0484 (2016)). *USDOE Energy Information Administration (EIA), Washington, DC (United States). Office of Energy Analysis.*

Craggs, C., Conway, E., & Pearsall, N. M. (1999). Stochastic modelling of solar irradiance on horizontal and vertical planes at a northerly location. *Renewable Energy*, *18*(4), 445–463.

Deng, L., & D. Yu. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, *7*(3–4), 197–387.

Diagne, H. M., David, M., Lauret, P., Boland, J., Diagne, H. M., David, M., … Review, N. S. (2014). Review of solar irradiance forecasting methods and a proposition for small-scale insular grids To cite this version : HAL Id : hal-01090087 Review of solar irradiance forecasting methods and a proposition for small-scale insular grids.

Diagne, M., David, M., Lauret, P., Boland, J., & Schmutz, N. (2013b). Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, *27*, 65–76.

Díaz, D., Alberto, V., Barrán, T., Omari, A., & Dorronsoro, J. R. (2017). Prediction. *Neural Processing Letters*, *46*(3), 829–844.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–2011.

Elsheikh, A. H., Sharshir, S. W., Abd Elaziz, M., Kabeel, A. E., Guilan, W., & Haiou, Z. (2019). Modeling of solar energy systems using artificial neural network: A comprehensive review. *Solar Energy*, *180*(January), 622–639.

F. Rosenblatt. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

Fouilloy, A., Voyant, C., Notton, G., Motte, F., Paoli, C., Nivet, M. L., … Duchaud, J. L.

(2018). Solar irradiation prediction with machine learning: Forecasting models selection method depending on weather variability. *Energy*, *165*, 620–629.

Glorot, X., & Y. Bengio. (2010). Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

Grekousis, G. (2019). Artificial neural networks and deep learning in urban geography: A systematic review and meta-analysis. *Computers, Environment and Urban Systems*, *74*(July 2018), 244–256.

Gueymard, C. (1986). Mean daily averages of beam radiation received by tilted surfaces as affected by the atmosphere. *Solar Energy*, *37*(4), 261–267.

http-1: https://www.machinelearningplus.com/time-series/arima-model-time-seriesforecasting-python/ (Date accessed: 22.10.2019)

http-2: http://www.visualgenedeveloper.net/ (Date accessed: 05.07.2019)

http-3: https://gbr.pepperdine.edu/2010/08/artificial-intelligence-techniquesenhance-business-forecasts/business-forecasts/ (Date accessed: 17.10.2019)

http-4: https://www.mathworks.com/help/ident/ug/what-are-nonlinear-arx-models.html (Date accessed: 10.07.2019)

http-5: https://www.ambientweather.com/solarradiation.html (Date accessed: 10.07.2019)

Jain, P. C. (1984). Comparison of techniques for the estimation of daily global irradiation and a new technique for the estimation of hourly global irradiation. *Solar & Wind Technology*, *1*(2), 123–134.

Jain, P. C. (1988). Estimation of monthly average hourly global and diffuse irradiation. *Solar & Wind Technology*, *5*(1), 7–14.

Ji, W., & K. C. Chee. (2011). Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN. *Solar Energy*, *85*(5), 808–817.

Jia, W., Zhao, D., Shen, T., Tang, Y., & Y. Zhao. (2014). Study on optimized Elman neural network classification algorithm based on PLS and CA. *Computational*

Intelligence and Neuroscience, *12*.

K. Chau. (2006). A review on the integration of artificial intelligence into coastal modeling. *Journal of Environmental Management*, *80*(1), 47–57.

Kalogirou, S. A. (2001). Artificial neural networks in renewable energy systems applications: a review. *Renewable and Sustainable Energy Reviews*, *5*(4), 373–401.

Ko, C. N., & Lee, C. M. (2013). Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter. *Energy*, *49*, 413–422.

Kohonen, T., Schroeder, M. R., Huang, T. S., & Maps, S. O. (2001). Springer-Verlag New York. *Inc., Secaucus, NJ, 43*(2).

Le, Y., Bottou, L., & Orr, G. (1998). Efficient BackProp, in Orr, G. and Muller K.(Eds), Neural Networks.

Liu, B. Y., & Jordan, R. C. (1960). The interrelationship and characteristic distribution of direct, diffuse and total solar radiation. *Solar Energy*, *4*(3), 1–19.

Luukkonen, P., Bateman, P., Hiscock, J., Poissant, Y., & Dignard-Bailey, D. H. and L. (2013). International Energy Agency Co-Operative Programme on Photovoltaic Power Systems, 2013. Retrieved from http://www.cansia.ca/sites/default/files/201306_cansia_2012_pvps_country_report_long.pdf

Mellit, A., Kalogirou, S. A., Hontoria, L., & Shaari, S. (2009). Artificial intelligence techniques for sizing photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews*, *13*(2), 406–419.

Mellit, Adel, & Kalogirou, S. A. (2008). Artificial intelligence techniques for photovoltaic applications: A review. *Progress in Energy and Combustion Science*, *34*(5), 574–632.

Mellit, Adel, & Pavan, A. M. (2010). A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy. *Solar Energy*, *84*(5), 807–821.

Nik, W. W., Ibrahim, M. Z., Samo, K. B., & Muzathik, A. M. (2012). Monthly mean hourly global solar radiation estimation. *Solar Energy*, *86*(1), 379–387.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning.

Pascanu, R., Gulcehre, C., Cho, K., & Y. Bengio. (2013). How to construct deep recurrent neural networks.

Pijanowski, B. C., Shellito, B., Pithadia, S., & K. Alexandridis. (2002). Forecasting and assessing the impact of urban sprawl in coastal watersheds along eastern Lake Michigan. *Lakes & Reservoirs: Research & Management*, *7*(3), 271–285.

Praynlin, E., & Jensona, J. I. (2017). Solar radiation forecasting using artificial neural network. In *In 2017 Innovations in Power and Advanced Computing Technologies (i-PACT)* (pp. 1–7).

Ren, G., Cao, Y., Wen, S., Huang, T., & Zeng, Z. (2018). A modified Elman neural network with a new learning rate scheme. *Neurocomputing*, *286*(April), 11–18.

Ryu, S., Noh, J., & Kim, H. (2017). Deep neural network based demand side short term load forecasting. *Energies*, *10*(1), 3.

Soubdhan, T., Voyant, C., & Lauret, P. (2015). Influence of Global Solar Radiation Typical Days on Forecasting Models Error. *The Third Southern African Solar Energy Conference (SASEC2015)*. Retrieved from https://hal.archives-ouvertes.fr/hal-01099487

Ssekulima, E. B., Anwar, M. B., Al Hinai, A., & El Moursi, M. S. (2016). Wind speed and solar irradiance forecasting techniques for enhanced renewable energy integration with the grid: A review. *IET Renewable Power Generation*, *10*(7), 885–898.

Sun, Y. (2018). *Deep Neural Network Regression and Sobol Sensitivity Analysis for Daily Solar Energy Prediction Given Weather Data*. Purdue University.

V. Bhatnagar. (2014). Data mining and analysis in the engineering field. *IGI Global*.

Voyant, C., Notton, G., Kalogirou, S., Nivet, M. L., Paoli, C., Motte, F., & Fouilloy, A. (2017). Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, *105*, 569–582.

Yahyaoui, I. (2018). Advances in Renewable Energies and Power Technologies. *Solar and Wind Energies. Elsevier.*, *1*.