Arab American University

Faculty of Graduate Studies

Classification and Prediction Model of Renewable Energy Depending on Weather Factors Using Artificial Intelligence Techniques

By

Sumoud Mahmoud Saeed Jaradat

Supervisor

Prof. Dr. Mohammed Awad

CO- Supervisor

Dr. Osama Omari

This thesis was submitted in partial fulfillment of the requirements for the Master's degree in Computer Science.

October 2020

Classification and Prediction Model of Renewable Energy Depending on Weather Factors Using Artificial Intelligence Techniques

By

Sumoud Mahmoud Saeed Jaradat

This thesis was defended successfully on **25/11/2020** and approved by:

Committee Members	Signature
1. **Supervisor:** Prof. Dr. Mohammed Awad	....
2. **Co-Supervisor:** Dr. Osama Omari	....
3. **Internal Examiner**: Dr. Amjad Rutrot	Dr Amjad RATTROUT
4. **External Examiner**: Dr. Emad Natsheh	.....

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

The work was done under the guidance of Professor Mohammad Awad and Doctor Osama Omari, at the Arab American University, Palestine.

Sumoud Mahmoud Jaradat

22.6.2021

Sumoud

Dedication

This thesis is wholeheartedly dedicated to my beloved mother, who has been my constant source of inspiration and gave me support to strength when I thought of giving up. To my husband and sincere friends who shared their words of advice and encouragement to finish this work.

Acknowledgments

I would like to express the deepest appreciation to my supervisors, Professor Mohammed Awad and Doctor Osama Omari for their valuable guidance and continuous encouragement to accomplish this thesis.

Also, I would like to thank the Palestinian metrological department and Tubas municipality for their efforts in providing the datasets.

Last but not least, I extremely grateful to my family for their love, caring, and supporting me spiritually throughout my life.

Abstract

For many decades, renewable energy resources are used as a sustainable and environmentally friendly resource to compensate for the shortage of conventional energy resources. Solar and wind energy systems are the most popular renewable energy resources that have been used to produce electricity. These systems have fluctuating nature due to the dependency on weather conditions. Accurate prediction of the potentially available power in a specific location is economically feasible for designing and installing renewable power systems. For this reason, an accurate prediction and classification model is necessary for the operation of this sector in Palestine and other countries.

This thesis proposes a hybrid model of Neural networks (NNs) and Optimization Algorithms (OAs) that uses the historical data of solar cells to predict the potential production of different solar energy farms installed in the main cities of Palestine. Also, meteorological data is used to calculate and predict the potential wind energy that can be obtained from wind fields in Palestine. The data sets of historical power production were collected from different locations in Palestine. these locations include Hebron in the south, Bethlehem and Salfit in the middle, and Tubas and Jenin in the north of the West Bank. The metrological data are collected using the Palestinian Meteorological Department for the cities of Jenin, Nablus, Bethlehem, and Hebron. The proposed model combines different types of NNs and OAs to enhance prediction results with the aim of training and testing the model to enhance the prediction result of the electrical energy production; using historical data of solar cells and weather data, especially solar radiation, temperature, and wind speed. The performance of the proposed model shows that the hybrid model system

that combines GAs with Multi-Layer Perception neural network (MLPNNs) outperforms the Particle Swarm Optimization (PSO) algorithm with MLPNNs model and Radial basis function Neural Network (RBFNNs) with GAs of predicting power production. GAs-MLPNNs achieved the lowest MSE among the other models. For instance, with 40 neurons, the MSE using the proposed model was 0.0048, where the MSE for GAs-RBFNNs was 0.0094 using a complex dataset of four years; while the PSO-MLPNNs model failed to handle the dataset. The GAs-MLPNNs model also demonstrate that better results can be obtained using the metrological dataset rather than the use of the historical dataset, where the MSE using the historical dataset was double that for results of the metrological dataset. The results of the model used for classifying the geographical regions depending on their potential power production. We employed a general classification algorithm; the self-organizing map (SOM) using the Kohonen neural network to classify the regions based on the predicted annual power production. The classification results demonstrate that Jenin, Tubas, and Bethlehem have similar potential solar power production, while Nablus, Salfit, and Hebron have diverse capacities of electricity production from the solar resource. The results also show a variety of wind power production for each city of Jenin, Nablus, Bethlehem, and Hebron.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AC	Alternating Current
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
DC	Direct Current
EAs	Evolutionary algorithms
EL-GAs	Elman- Genetic Algorithms
EL-PSO	Elman- Particle swarm optimization
FIS	Fuzzy Inference System
GAs	Genetic Algorithms
GAs-MLPNNs	Genetic Algorithms Multi-Layer Perceptron Neural Networks
GD	Gradient Descent
GSO	Genetic Algorithms- Particle Swarm Optimization
IOTs	Internet Of Things
KWH	Kilo Watt Hour
MAF	Moving Average Filter
MLPNNs	Multi-Layer Perceptron Neural Networks
MPH	Mile Per Hour
MSE	Mean Square Error
NNs	Neural Networks
OAs	Optimization Algorithms
PSO	Particle Swarm Optimization
PV	Photovoltaic
RBFNNs	Radial Basis Function Neural Networks
RE	Renewable Energy
SL	Supervised Learning
SOM	Self-Organizing Mapping

Chapter One: Introduction

Introduction

Conventional energy resources such as oil and gas about to be exhaustible with the increasing electricity demand. Furthermore, many problems are emerged due to these polluting resources as they are contributing to global environmental disasters, global warming and the greenhouse gas emission. In Palestine, the electrical energy resources face more difficulties because of the unstable political condition and the limited conventional energy resources in the Palestinian lands. With the high density of population and fast-growing electricity demand, it became very challenging to provide adequate energy for many areas. Renewable energy, which is available and permanent resources, are the perfect alternative to overcome these limitations of conventional resources. Renewable energy can be used to keep the earth clean by reducing the potential environmental problems since it is considered as clean and environment-friendly energy. Also, it can compensate for the shortage of electricity demand because it is reliable, affordable, and environmentally sustainable energy. This energy is also cost-effective and it can contribute to the financial growth of any developed and developing country [1]. Solar energy is the most famous one of renewable energy resources since it was used previously in water heating and cooking. Later, the world is turning to solar energy for electricity generation without harming the environment because it does not produce any damaged remnants. The other major resource of clean energy is wind turbines. The implementation of this system became very popular in the last decade as an alternative to conventional energy in areas with low solar radiation and enough wind speed. Another form to use wind energy is the hybrid solar-wind renewable system that depends on solar radiation and wind speed to produce energy in areas with a diverse climate [2]. The combination of these two resources can be considered as a better solution to solve the electricity problems and to produce an adequate amount of electrical energy whether in standalone mode or grid-connected system; by

avoiding the oscillating change of weather conditions from semester to another. The amount of the produced power mainly depends on the weather conditions which vary from region to another [3]. So, choosing the installation location will affect the efficiency of the farm. This requires classifying the geographical areas into regions depending on the weather conditions to choose the fittest location for installing the system. It also requires predicting the potential amount of power production in a certain region depending on historical weather data to design an appropriate system. Because of the variable nature of the renewable energy generation, predicting the potential amount of power production provides a tool for managing and organizing the production of the systems which increases their effectiveness and reliably. Accurate prediction of the output lets the operator best optimize the size of the plants which can reduce the cost of the system. Also, it helps in scheduling the output of the grid connected power systems for better load distribution and balancing. There are two types of power prediction: short-term power prediction which provides predictions up to 7 days ahead, this type manly used for output scheduling. The second type is long-term prediction which refer to predicting the annual or monthly output of the system. There are four approaches to make power prediction, a statistical approach which depends on statistics to predict power from previous observations of power production, physical ways that use satellite images and numerical weather prediction to predict power production, artificial intelligence (AI) and machine learning (ML) methods that use time series to predict power production, and the hybrid approach which combines two of the previous approaches [4]. Statistical and physical methods have a limited capability on predicting the complex behavior of the weather conditions as cloudiness, temperature and wind speed. So, the use of AI techniques with multiple inputs models as ANN can increase the prediction accuracy. AI techniques characterized by their ability to predict complex patterns of data by self-learning property which provides high output accuracy

and reliability. AI and ML methods are widely adopted in nonlinear prediction. In [5], ANN used to extract the complex characteristics of the input data and perform well in nonlinear prediction. Support Vector Machine (SVM) is used in [6] for classification and regression prediction of power production. In [7] power production is also predicted by RBFNNs. The hybrid prediction approach is the most common approach for the renewable energy output prediction that combines the prediction results and avoids the limitations of the single models. In [8] genetic algorithm is used to optimize three different types of neural networks for power prediction. Also, the neuro-fuzzy logic method is used to predict the performance of power systems in [9]. In this thesis, we used a hybrid model that combines ANNs with the OAs for long-term prediction and classification of renewable energy systems production. The GAs used to optimize the parameters of MLPNNs to improve the performance of the network for annual power prediction. Then, the SOM algorithm is used to classify the West Bank into regions depending on the annual power production. A historical dataset of solar farms in Palestine used for power prediction. Moreover, a metrological dataset for several cities in Palestine was collected to calculate and predict the potential power production based on weather conditions. The collected datasets preprocessed using a sequence of processes before applying to the model. The datasets were used for both training and testing the proposed model, also they applied to other models to validate the performance of our model.

1.1 Objectives

Power prediction can be beneficial for improving the electricity system and help in planning the distribution of the electricity network. ANNs are self-learning models that can learn the relationship between inputs and outputs. Thus, it is helpful to use ANNs to obtain high accuracy and reliability of the power prediction system from the weather conditions which have a complex nature. In this thesis, we aim to predict the potential output of solar and wind energy systems in Palestine based

on historical and metrological data. Because real data often consist of linear and nonlinear parameters, we propose an GAs-MLPNNs hybrid model that combines ANNs with OAs to enhance the efficiency and the accuracy of predicting the renewable energy output using methods of ANNs. The GAs-MLPNNs model employs GAs to optimize the MLPNNs parameters then the resulting network can be used for power prediction. We also aim to classify the geographical area of West Bank into regions based on the potential renewable energy. Through the thesis, we aim to accomplish the following objectives that lead to the main goals of this work.

- Collecting historical solar and wind data from all cities of West Bank.
- Collecting metrological data for all cities of West Bank.
- Preprocessing the collected data and selecting the best datasets for training and testing the proposed model.
- Examining the proposed model by comparing its performance with single and hybrid models.
- Employing the hybrid GAs-MLPNNs to predict power production depending on historical data of RE systems.
- Employing the hybrid GAs-MLPNNs to predict the potential power depending on metrological data and theoretical calculations of the selected locations.
- Compare the prediction results of historical data and the prediction result of a theoretical calculation, to perform the robustness of the applied GAs-MLPNNs.
- Apply SOM using Kohonen network for classifying the geographical regions depending on prediction results; to decide the best installation locations in Palestine (West Bank) based on the prediction of the power production for these locations.

1.2 Contribution

In this thesis, artificial intelligence techniques that include ANNs and evolutionary algorithms have been applied to improve the predictability of potential energy production from renewable sources. An GAs-MLPNNs hybrid model was proposed and applied to several data, which includes historical data for solar farms production in various areas of the Palestinian territories. It has also been applied to metrological data to predict the amount of energy that can be produced in the selected locations. The model indicated the preference of using meteorological data over historical data. The results helped in depicting a visualization of the renewable energy distribution in cities of West Bank, that can be used to solve electrical energy problems in Palestine. The model was able to classify the geographical regions in the West Bank according to the productive capacity of this region from renewable energy using SOM NN.

1.3 Matlab Environment

To perform the prediction using our proposed model we used the Matlab environment. Matlab is a programming and data analysis environment that can be used for models simulating and building. Moreover, it provides add-on toolboxes to predict the future behavior of a certain data using neural networks. Our proposed model has been simulated using Matlab 8.3.0 under windows 10 with I3 core processor. Also, we used the neural network and network fitting toolboxes to test and evaluate the performance of our model.

Summary

In this chapter of the thesis, we illustrate the problem statement that is the renewable energy prediction from metrological data and historical power production data. It also presents our objectives and contribution. The rest of the thesis is organized as follows. Chapter two introduces the solar and wind energy systems and the effects of the weather conditions on the performance of

the systems. Also, it illustrates the common methods that are used for energy prediction and classification. Chapter three is used to explain the AI and NNs methods that are used to enhance the accuracy and reliability of the prediction system. Chapter four of the thesis consists of the proposed hybrid ANNs model that we used for prediction and classification of the amount of energy in several areas of the Palestinian territories. It also contains a detailed description of the dataset and the experiment's procedures. Chapter five is dedicated for discussing the results of using the proposed model and comparing the results with other models results. Chapter six includes the thesis conclusion and the most important obstacles, in addition to the future aspirations for the development of the model.

Chapter Two: Background

Introduction

Renewable energy (RE) resources have to play important role in the world's future since it fulfills the defined social, economic, and environmental aspects by producing energy again and again, with low cost, free of air pollution and greenhouse gases which can provide electricity demands and reduce the environmental problems. For many decades, people were used various types of RE resources to suit their needs due to their environmental and economic benefits. Hydropower was used to power machines, in what was called a watermill, by utilizing the flow of rivers or the flow of water from above. One of its most important uses is the grinding of seeds, especially wheat. They also were used the energy of water falling; hydroelectric energy, for producing electricity to operating the factories and lighting the residences. Moreover, solar energy has been used for warming homes, water heating, drying crops, or for cooking through solar collectors which can convert the solar energy to heat energy. Currently, heating water by sunlight is common in many schools, restaurants, homes, and hospitals. It is also widely used in greenhouses on farms. Furthermore, solar energy has been used extensively to generate electricity at a low cost by using the photoelectric (PV) system, especially after the increase of oil prices. It converts solar radiation energy directly into electrical energy, and the resulting electrical energy can be stored in huge batteries for use in the absence of the sun [10]. Countries that have high annual solar radiation should be interested in this technology to obtain energy due to the presence of sunlight on most days of the year. On the other hand, wind energy is one of the renewable energy resources that has been used as a substitute for fossil fuels and nuclear power. The kinetic energy of the wind can be exploited by using wind turbines to produce electrical energy, and it is considered as a type of electromechanical energy [11]. Other types of RE are emerging to meet the increasing demand for electricity like geothermal, marine, biomass, biofuel, and more. Solar and wind contribute to the

highest potential RE resources. The overall global renewable capacity totaled some 2,378 GW by the end of 2018. Solar PV systems are the frontrunner for installed renewable power capacity. Additions for solar PV accounted for 55% of new renewable capacity followed by 28% for wind power and 11% for hydropower [12]. Additions for renewable power generation capacity outpaced net installations of fossil fuel and nuclear power capacity combined.

2.1 Renewable Energy in Palestine

The energy sector in Palestine faces many difficulties due to the dependency on other countries to import fossil fuel, which increases the cost of energy to the highest in the middle east with the lowest consumption per person [13]. Also, the instability of the political conditions affects directly the development of the energy sector since the energy sector infrastructures are controlled by the occupation to a far extent. Mainly, RE can obtain the main advantage of green energy; reducing the impact on the environment by reducing greenhouse gas emissions. Moreover, RE can play a key role in the transition to sustainable, affordable, and reliable energy development because of favorable weather conditions. Palestine features promising capacities in the potential use of solar and wind energy due to the high solar radiation and moderate wind speed in Palestine which encourages the exploitation of these resources for different applications and reduce the dependency on other countries [14],[15]. Figure 1 illustrate the monthly averages of solar radiation in different cities in West Bank in year 2010. The almost whole country has high sunshine hours throughout the year with average solar energy ranges of 5.4 KWH/M/day to 6 KWH/M/day [14]. On other hand, wind speed varies depending on the regions and its topographic features. 5 m/s at least is needed to operate the wind turbine. In Palestine, the highest wind speeds were observed in the regions with elevations around 1000 m above sea level; Nablus, Ramallah, Jerusalem, and Hebron, with an average wind speed of 4 to 8 m/s [15]. This value encourages to use of small wind turbines

to feed the rural areas that are located far from the grid or to operate the pumps on the farms. So, RE can be considered as an ideal solution to many energy problems in Palestine. It can substitute the increasing energy demand because of the rapid growth of the population. Also, it can be used as an alternative to the conventional energy resources on the isolated areas in Palestine, and reduce the dependency on foreign resources.

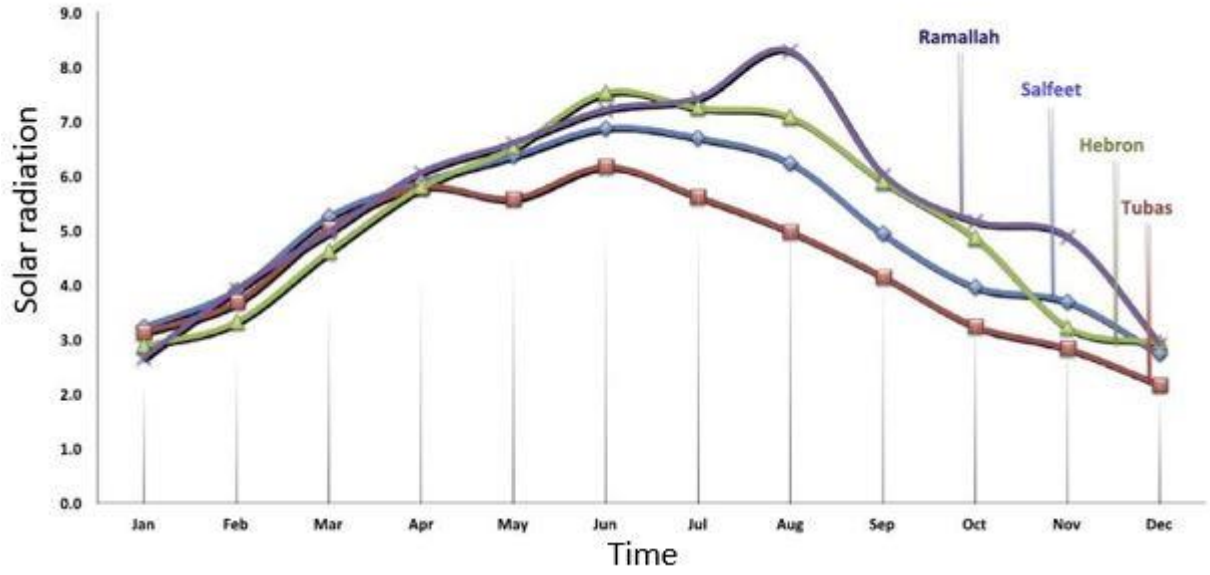

Figure 1 monthly averages of solar radiation in West Bank [14]

2.2 Solar Energy

Solar energy is the light spectrum and heat emits from the sun and hits the earth after a long journey from space. Actually, the heat energy of the sun does not reach the earth due to the large distance between them; and the source of the heat is the light energy itself [16]. This energy has been exploited for many purposes a long time ago. Humans have used solar energy using a range of constantly evolving technologies that are meeting their needs. These technologies include the use of solar energy directly for water heating, cooking, and houses warming, or indirectly for generating electricity. The electrical energy can be produced from solar radiation using two

methods: thermal solar energy system and PV system. The thermal system depends on heating fluids in a pipe network with running turbines to produce electricity, where the PV system uses solar cells that utilize the PV phenomenon to produce electricity directly from solar radiation [17]. Solar energy is considered an essential source of RE due to the large availability of solar energy which makes it a highly attractive source of electricity. The following subsections discuss the PV system and the process of generating electrical energy using the PV phenomenon.

2.2.1 Solar Cells

Solar cells are among the recent achievements that contribute to the reliance on RE by generating electricity from the sun. A solar cell is a device made up of semiconductor materials such as silicon [18]. It can be defined as PV cells that absorb photons (light particles carrying electromagnetic energy) that are released from the sun and convert it directly into an electrical current. The solar cell consists of the following components to produce electricity [19].

- Layer of non-reflective glass to pass the sunlight through with minimum reflection and dispersion.
- Front contact: a cathode layer of copper wire conducting electricity to receive electrons that enter the positive layer and merge with the holes.
- The absorber layer absorbs the photons from the sunlight that strikes the cell. This layer includes three sub-layers:
    - Negative-type semiconductor added with phosphor.
    - Positive-type semiconductor added with boron.
    - P-N junction layer.
- Back contact: an anode layer of copper wire conducting electricity. Figure 2 illustrates these layers of the solar cell.

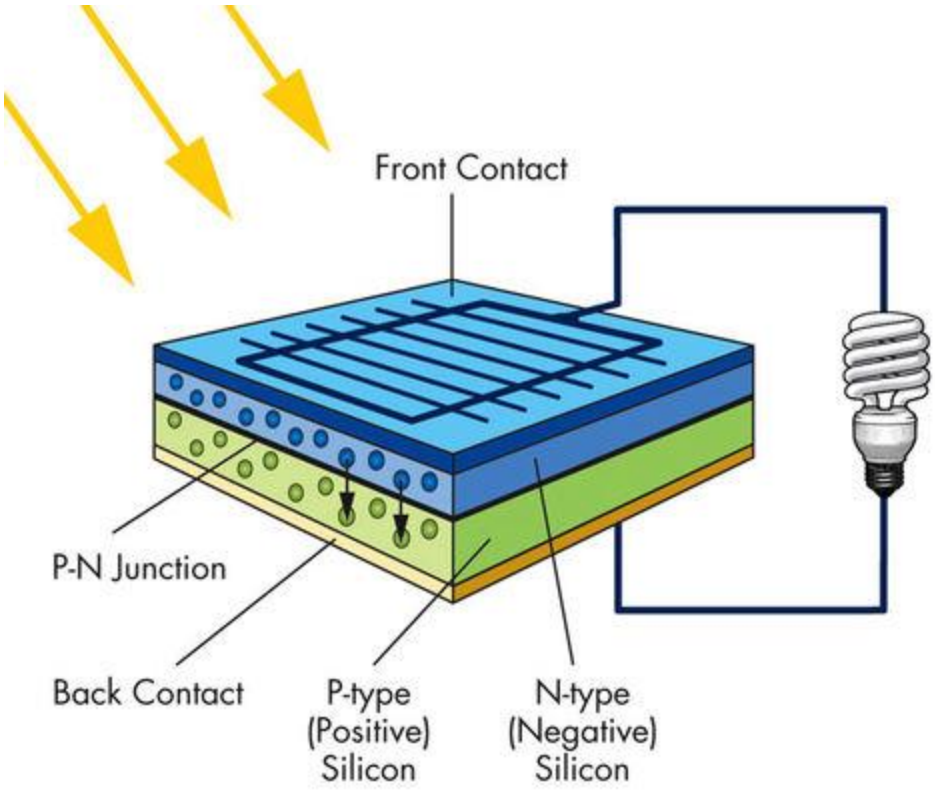

Figure 2 Solar Cell Components [20].

Solar cell efficiency is measured by the percentage of the energy that is converted into electrical energy in the cell. In general, three factors that play important role in the efficiency of the manufacturing material; sensitivity to the sunlight, thickness of the material, and surface reflectivity of the cell [21]. The cell is sensitive to light with a wavelength above the bandgap of the material and it is not sensitive to light with a wavelength less than the bandgap as it cannot convert light into energy through this part of the light. Also, the thickness of the semiconductor material in the cell affects the efficiency of the cell by absorbing the photons energy as the thickness increases. The nature of the cell surface also affects cell efficiency. Whereas, when a portion of the light is reflected and not absorbed into the cell, it reduces the energy that can be produced. So the solar cells are covered with non-reflective materials during manufacture.

The first production of a solar cell was in 1954 with low efficiency of about 6%, high cost, and fragility of used materials [22]. The development of the solar cells continued to 30% efficiency for space applications [23]. The most popular material used for solar cell production is silicon, which has some properties as; availability in nature, environmentally friendly, and unpolluted material, and it can be easily formed into a monocrystalline silicon form. Three major types of silicon are used to manufacture the solar cells [24], [25]:

- Crystalline Silicon: Two types of crystalline silicon are used to produce solar cells; monocrystalline silicon which has single crystalline silicon, and polycrystalline silicon that has multi-crystalline silicon. The polycrystalline silicon cell has a lower efficiency than monocrystalline crystalline silicon cell but both of them have high efficiencies that average about 10-12%.
- Amorphous Silicon: A thin-film silicon cell that absorbs light more effectively than crystalline silicon cell, so it can be made thinner. this type is suitable for applications that do not require high efficiency and low cost is important. The typical efficiency of amorphous silicon cells is around 6%.
- Hybrid Silicon: A combination of single crystalline silicon surrounded by thin layers of amorphous silicon provides excellent sensitivity to lower light levels or indirect light. The Hybrid silicon cell has the highest level of conversion efficiency about 17%.

Other types of materials have suggested manufacturing solar cells to reduce the cost of manufacturing. One of them is the copper Phthalocyanine which is organic material. It is available and has the required resistance to the environmental conditions but still have lower efficiency than silicon cells [26]. Hybrid material also can be used to enhance the efficiency of the cell by

increasing the bandgap, but this solution is costly. Therefore, an increase in efficiency comes at an increase in cost.

2.2.2 Photovoltaic Power Production

The PV technology is considered a real revolution in the generation of electrical energy because it depends on the generation of energy directly from sunlight in an effective manner without causing environmental damage. The electric energy is produced using the PV cells by converting the light into an electric current using the semiconductors. When sunlight or photons of light enter the surface of the non-reflecting glass into the solar cell, solar cells absorb sunlight by the absorber layer, which generates motion of electrons due to the absorbed energy, depending on the strength of the solar radiation. Then, the electrons transfer to the holes in the negative layer, and the holes transfer to the positive layer. As a result of their transmission between the two layers, an electric current or a continuous voltage difference is generated that passes through the electrical conductor (cathode and anode). If the positive and negative junctions of the solar cell are connected to electrical equipment, the current is delivered to operate the electrical equipment [27].

To conclude the electricity production using PV cells, there are 5 fundamental processes to produce electricity:

- The first step is collecting the solar photons.
- Then, the generation of the holes and electrons pairs due to the power absorption.
- The next process is the diffusion of the holes and electrons pairs in the N-P junction layer.
- The dissociation of the pairs into holes and electrons
- The last process is to transport the holes and electrons toward the P-layers and N-layer which produce a voltage difference between the two layers of conductors that generate current flow.

2.2.3 Photovoltaic System

The solar cells are the units that are responsible for generating electricity; by converting the energy of sunlight to the electrical current due to the PV phenomenon. Other components are required to build an integrated device that can be used as an RE product to supply power to equipment. The PV system that is illustrated in figure 3 contains all the required components to complete the process of generating power for the required equipment [28].

- Solar module: a group of solar cells that are connected to form an array. These cells are connected in different ways depending on the requirements. If the purpose of the solar arrays is to increase the output current, the cells are connected in parallel. And, if the solar arrays are connected in series, the output voltage will increase. The solar modules are also arranged in arrays to raise the throughput of the PV system.
- Inverter: the output of the solar cells is only direct current (DC), therefore they are only used for DC equipment. If alternating current (AC) is needed for AC equipment, a critical component is needed for the PV system to converts DC power output of solar arrays into AC for AC devices.
- Battery: used for energy backup, to store the electricity that is produced from solar arrays for use when sunlight is not available.
- Charge controller: is used to regulating charging the battery, which prevents the battery from overcharging and also performs control over discharges.
- Loads: the electrical devices that are fed by the photovoltaic system.

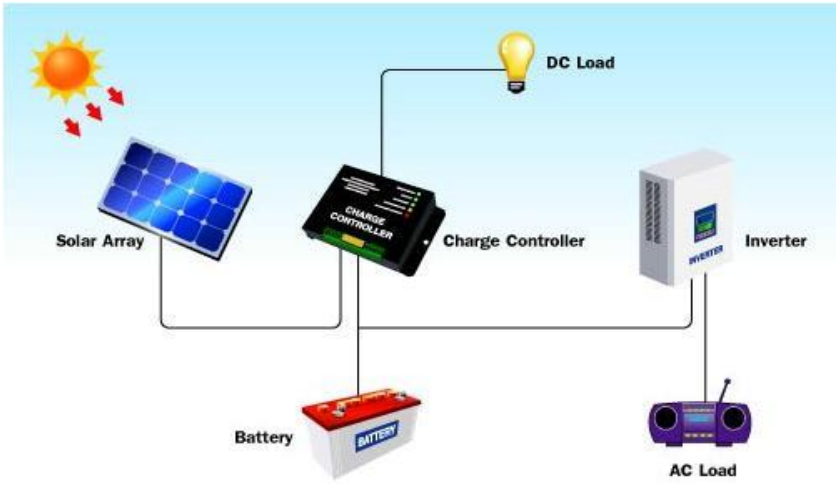

Figure 3 Photovoltaic System [29].

The PV system is affected in terms of the amount of energy that can be produced from solar energy by several factors. Most silicon cell efficiencies are averaged by 15-20% which is a cost-effective ratio [30]. But the manufacturing materials of the solar cells can be affected by other factors that reduce the throughput of the PV system. These factors include the weather conditions of the surrounded region, the position of the sun in the sky, the direction and the angle of the solar panels, and the location of the installed system [31]. In general, the temperature of the surrounded region has a direct influence on power production and system efficiency. High temperatures can reduce the output of the system and harm the materials of the solar cells. Also, the sun provides a different amount of radiation through the days depending on its location in the sky which makes the amount of the system throughput vary. On other hand, different regions of the earth have a different number of sunny days throughout the year according to the distance from the equator. Thus, the number of sunlight hours and sunny days are oscillating depending on the installation location of the system, which affects the efficiency of the system. These factors have approximately the same behavior over the years. So, the prediction of these factors can help in predicting the output of the PV system for making decisions about the installed capacity and the output distribution of the

power system. Moreover, classifying the regions depending on these factors or by the predicted power throughput can help in identifying the best geographical places to install the system. Other factors can affect the performance of the system as dust on the solar panels and cloudiness which directly reduce the amount of sunlight that reaches the cells.

2.3 Wind Energy

Wind energy is considered as a form of solar energy because of the contribution of the sun to producing the wind and controlling its movement and direction. The wind is formed due to the change in the air temperature near the surface of the earth. The sun heats the Earth's surface and the air near it to varying degrees, so the Earth's temperature at the equator is higher than at the north and south poles of the Earth. This leads to a decrease in air pressure in the areas near the equator due to the heat. The hot air rises to the upper atmosphere because of its low pressure, which leads to the transfer of air from the less heated regions (in the north and south of the globe) to the warmer regions (the equator regions) forming the wind [32]. The wind is also formed by the effect of the Earth's rotation around itself, or what is called the Coriolis Effect [33]. Where the rotation of the Earth around itself displaces air from the east to the northern hemisphere and from the west to the southern hemisphere.

As the population is rapidly growing and the electricity demand is continuously increasing, sustainable resources of energy should be exploited to reserve the conventional resources for the future and meet the rising energy demand. Wind energy is considered an important RE resource because it is available, clean, and sustainable energy that can fulfill the growing energy demand effectively and help in controlling the environmental problems by reducing the toxic gas emission due to fossil fuel burning. Wind energy is described as the process of generating electricity by converting the kinetic energy in the wind into mechanical power using wind turbines; Then, the

generator of the wind turbines can convert the mechanical power into electricity [34]. People have used wind energy since ancient times, where they used the wind to run and direct boats in rivers. Wind energy can be utilized by converting the kinetic energy of wind into mechanical energy, like the case with windmills that is used to run water pumps on farms and remote areas. The kinetic energy of the wind can also be converted into electrical energy, as generators convert the mechanical energy into electrical energy that can be used in homes, factories, street lighting in remote places, and for other purposes.

The first wind turbine to produce electricity was used in 1887 by Professor James Blythe to charge the batteries and light his house, making it the first house in the world to be lit by electricity produced from wind energy [35]. Wind turbines produce electricity by converting the kinetic energy of the wind into mechanical energy by capturing the wind by fan-like blades attached to the rotor, which in turn conveys this torque to an electric generator to convert mechanical energy into electrical energy. Wind turbines mainly consist of three parts: blades, shaft, and generator. Also, there are other components whose function is to control the turbine work and maintain its work effectively [36].

- Blades: The blades act as barriers to the wind; when the wind hits the blades, the wind spins the blades around the axis. The blades are connected by the rotor, which in turn is rotated by the rotation of the blades.
- Shaft: A rotational axis whose functionality is to transfer the mechanical energy from the rotor and convert it into rotational energy which in turn drives the generator.
- Electrical Generator: The electrical generator is usually an induction generator or asynchronous generator.

- Pitch: Its function is to bend the direction of the blades to control the rotational speed of the rotor to reduce the impact of the wind force in case the wind speed is too high, and to avoid the generator operating at very low speeds.
- Tower: It is usually made of steel, and has a suitable height to reach high wind speeds.
- Nacelle: Contains the basic components of the wind turbine. Also, the wind turbine rotor is connected to the nacelle.
- Gearbox: Its functionality is to make the low rotational speed of the rotor increase to a higher speed capable of operating the generator.
- Controller: It has a computer to monitor and control the work of the wind turbine at suitable wind speeds.

Wind turbines are usually installed in the form of interconnected groups called wind farms, the number of which is according to the power required for the system. Wind farms are divided into two main types, depending on the size of the farm: Grid-Connected farms and Stand Alone or Distributed farms. The first type is connected to an electrical network where the turbines supply the power grid with electricity, which in turn distributes electricity to the end-user. The other type is the farms with a small number of turbines feeding one facility such as a house or office. This type is usually used in remote places that are not connected to the power grid. The efficiency of wind farms depends on several factors, which can be summarized as follows:

- Wind speed: The amount of energy produced by the turbine is directly affected by the wind speed cube [37]. That is, if the wind speed has doubled once, the output of electric energy has increased eight times.
- Air Density: The ability of the wind to operate the turbine increases when its density increases. The air density varies with the temperature change: cooler air has a higher

density than hot air at the same height as the Earth's surface. Air density also decreases when it rises above the ground.

- Blade size: The size of the blades affects the swept area of the rotor. The greater the blade size, the greater the area exposed to the wind force.
- Turbine design: Horizontally installed turbines are considered to be more effective in converting the kinetic energy of the wind into mechanical energy because the wind is perpendicular to the blades, making it easier to capture the wind. The vertical turbines are usually less efficient than the first design because the turbines catch wind only at the front of the blades [38].
- Wind direction: Rapid and successive changes in wind direction may affect the efficiency of the turbines and sometimes lead to disruption or damage to the rotor and blades.

2.4 Datasets Description

Typically, the time series method is used to predict energy production from renewable sources with two types of data. The first type is the historical readings of power production from renewable energy fields. As for the second type, is the metrological data on which renewable energy production depends. Through this type of data, potential power production can be predicted by calculating the amount of power from the predicted metrological data in a mathematical way. It is also possible to predict the amount of potential power production by calculating the amount of energy from the historical reading of climatic factors and then predicting the amount of energy that can be produced in the future using statistical prediction methods or using artificial intelligence methods. In the thesis, we used the two methods to predict the amount of solar and wind energy for several regions in Palestine, including the northern, central, and southern West Bank. We have collected historical data for several solar power plants that include previous

observations for several years of electricity production [39], [40]. We also collected meteorological data from the Palestinian Meteorological Department for several cities in the West Bank, including wind speed, daily maximum temperature, and the intensity of solar radiation [41].

2.4.1 Historical Dataset

The historical dataset includes the previous observations of the power production for five solar stations located in Jenin, Tubas, Salfit, Bethlehem, and Hebron cities. The data of Jenin and Tubas sites include the daily power production of solar panels in kwh for four years from January 2015 to December 2018. The data for Bethlehem city represents the historical data of the power production in the period January 2016 to December 2019. For Salfit and Hebron cities, the data represents the power production in the period of January 2017 to December 2019. Table 1 illustrates the maximum and the minimum power production for each site in the mentioned periods. The following figure illustrates the pattern for one sample of the data, power of 1460 days for Bethlehem site, that will be used for the prediction, figure 4. The pattern demonstrates the continuous changes in the production during the years. The peaks refer to the maximum power for each year which occurs in the months of spring, where the bottoms refer to the production at the winters since solar radiation is the lowest during the year.

Table 1 Maximum And Minimum Power Production

Jenin	2015	2016	2017	2018
maximum	487.1	469.1	451.6	454.3
minimum	24.3	6.7	24.5	13.3

Tubas	2015	2016	2017	2018
maximum	36.3	36.2	34.18	34.76
minimum	4.38	4.1	5.57	5.03

Bethlehem	2016	2017	2018	2019
maximum	114.46	109.03	108.77	109.53
minimum	13.26	19.1	11.36	11.09

Hebron	2017	2018	2019
maximum	147.68	139.3	137.92
minimum	6.95	6.92	12.12

Salfit	2017	2018	2019
maximum	262.389	253.78	264.41
minimum	9.5	10.31	18.78

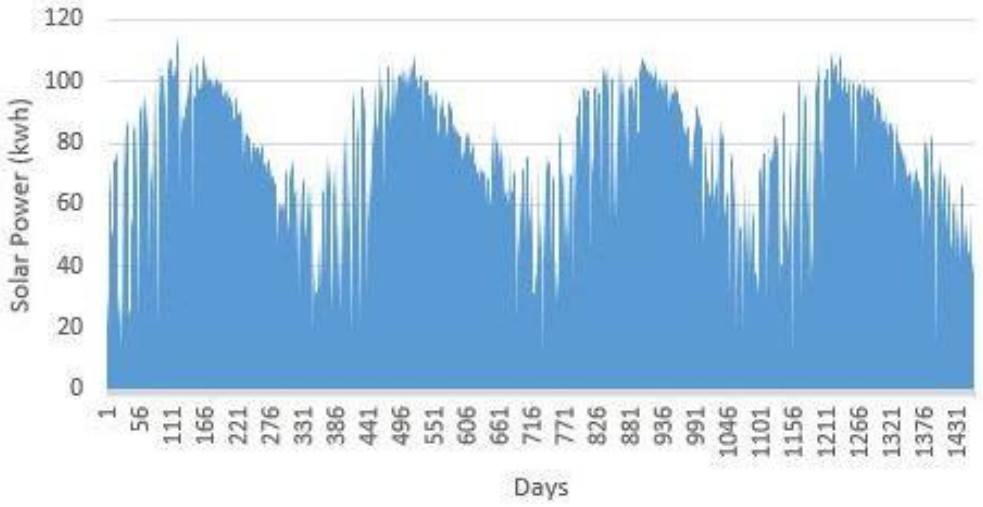

Figure 4 Bethlehem Solar Power Production

2.4.2 Metrological Dataset

The metrological dataset is collected by the Palestinian Meteorological Department for four cities in West Bank that are, Jenin, Nablus, Bethlehem, and Hebron. The dataset contains the maximum and minimum temperature, solar radiation, and wind speed for each city. These parameters were used for calculating the solar and wind power during the observation period. The power predictions relay on these calculations to obtain the production for the future time period, the next year. Table 2 represents the maximum, minimum, and average wind speeds for specific locations in all cities in the period January 2016 to December 2019. This dataset will be used for calculating and predicting wind power in the four cities for the 2020 year.

Table 2 Maximum, Minimum, and Average Wind Speeds

		2016	2017	2018	2019	Average
Jenin	Max	8.8	8.6	7.1	9	8.375
	Min	1	0.3	0.5	0.5	0.575
	Avr	3.23	2.38	2.6	3.2	2.84
Nablus	Max	6.1	5.4	5.8	6.5	5.95
	Min	0.5	0.8	0.9	0.8	0.75
	Avr	2.72	2.51	2.51	2.57	2.58
Bethlehem	Max	11.7	13.2	13.2	13.7	12.95
	Min	0.8	0.3	0.5	0.2	0.45
	Avr	3.15	2.74	3.49	3.47	3.21
Hebron	Max	14.8	11.5	19	14.6	14.975
	Min	1.8	2	1.8	2	1.9
	Avr	3.64	3.04	3.08	3.16	3.23

Figure 5 represents the wind speed pattern of Bethlehem in the interval 2016 to 2019 after ignoring the non-regular values.

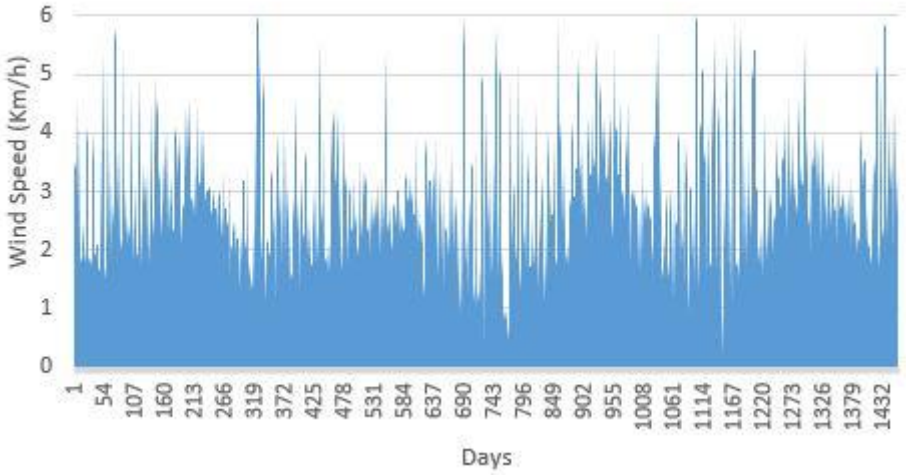

Figure 5 Wind Speed of Bethlehem

The second part of this dataset contains the daily maximum temperature and daily average solar radiation for Jenin, Nablus, and Bethlehem. These parameters are used to calculate and predict

solar power production. The available data for Jenin and Bethlehem was in the interval January 2016 to December 2019, while the data for Nablus was in the interval January 2016 to June 2017. As a sample, figures 6 and 7 illustrate the solar radiation and maximum temperature for Jenin in the four years respectively.

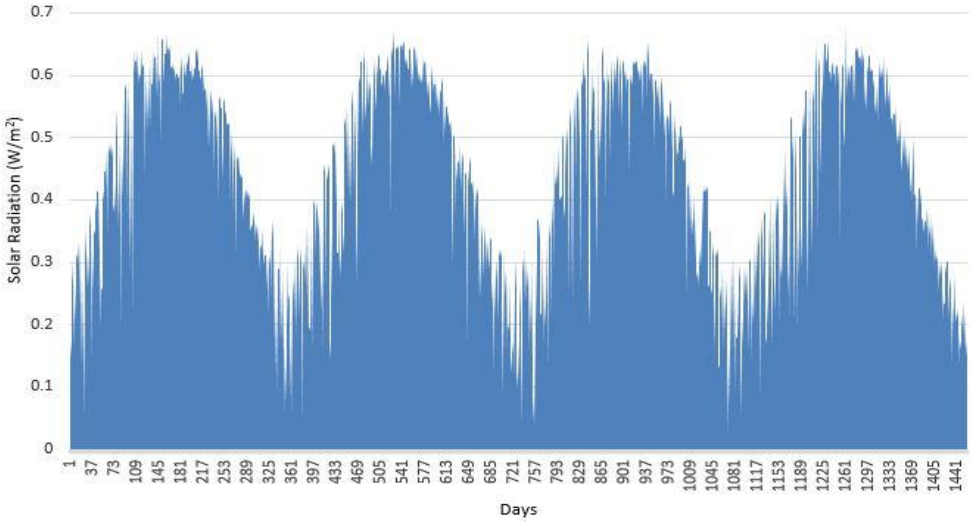

Figure 6 Solar Radiation for Jenin

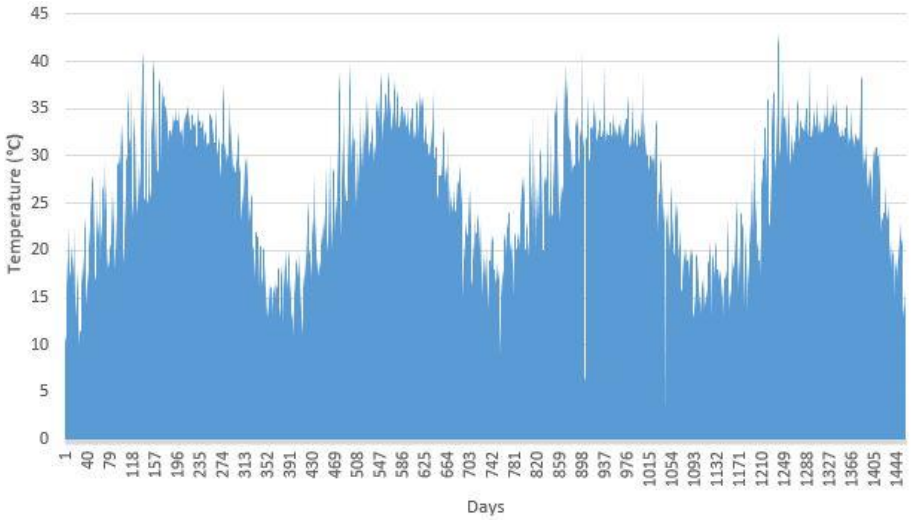

Figure 7 Max Temperature for Jenin

The figures show uniform patterns repeated in every year. The peaks of the patterns refer to values in the summer semesters where the solar radiation and the temperature have the highest values.

2.5 Related Work

The main purpose of the thesis is to predict and classify alternative energy sources in Palestine depending on climatic conditions, which are characterized by sudden changes and their complexity. Artificial intelligence and machine learning methods have been used in many applications in the field of renewable energy including prediction, classification, and fault diagnosis; due to their high ability to learn complex data patterns.

In [42] ANNs were used to diagnose faults in a photovoltaic power system, depending on solar radiation and photovoltaic cell temperature. The model was able to identify eight types of faults according to the expected value of the current and voltage as outputs for the photovoltaic cells. Mellit et al. [43] used ANNs with Fuzzy Inference System (FIS) to calculate the total energy that can be produced from renewable energy sources on the Internet of Things (IoT) system using sensors, body temperature, and photovoltaic cells. The results showed that ANN outperformed FIS in estimating the amount of energy produced. Puri et al. [44] used FIS to determine the price of energy generated from renewable sources based on the change in supply and demand in a proposed system for a smart energy community; in which participants exchange surplus production to reduce storage costs. ANNs were used in [45] to predict the amount of electrical consumption of energy produced from fossil fuels in 2020, in India; to study its replacement with renewable energy resources, taking into account the demand and the reduction of electricity production costs. FIS was used and it turns out that 32% of renewable energy can replace fossil fuels by 2020. In [46] MLPNNs were used to predict the amount of solar radiation using geographic and meteorological data such as temperature, humidity, sunshine duration, and rain from satellites. The model was able to predict the monthly average solar radiation, in which they relied on to calculate the average solar energy that could be produced. The results were used to design maps showing the amount of

solar radiation for major cities in India. Also in [47] ANNs were used to predict the amount of solar radiation in five different locations in the State of Kuwait to help the government and investors determine the best places to produce solar energy. In the field of wind energy, Haupt et al. [48] presented a study that aims to reduce the operating costs of wind farms through an accurate prediction of the amount of energy likely to be produced. PSO algorithm was used to predict wind speed in addition to calculating the amount of energy using numerical methods. the model has achieved satisfactory results, which enabled the management and operation of wind farms efficiently. Said et al. [49] used AI methods along with physical methods for short-term prediction in addition to forecasting for several days the amount of energy that could be produced from dual renewable energy systems that contain wind turbines and photovoltaic panels. As well, in [50], ANNs proved its superiority in predicting the instantaneous results of electrical energy resulting from the photovoltaic panels without relying on technical details of the panels and using a small amount of input data taken from a solar system in Germany; solar radiation, wind speed, and ambient temperature; with obtained error ratio ranged 2.2 to 4.6% RMSE. In [51], the EL-PSO model was used to predict wind power generation. The PSO algorithm was used to optimize the Elman NN parameters, weights, and bias, by defining them as an optimization problem. EL-PSO showed better performance compared with EL-GA for short term prediction. The paper indicates that the higher the data distribution, the weaker performance of NN. Quan et al. [52] presented a hybrid model to forecast PV and wind power by using EAs with NNs to speed up the convergence of the ANNs training phase and reduce forecasting error. It proposed a GSO model that combines the effectiveness and speed of the GAs and PSO algorithm with a large number of neurons and the performance outperforms the conventional algorithms. Results showed accurate forecasting for clear weather conditions and permanently covered days but leak precision when variability

conditions are higher. Lee et al. [53] proposed three particular models to predict PV output power. The first is a single layer ANN-based model to learn PV power in the short term. The second is a multi-layer ANN-based model to estimate more accurate PV power output. The last one is a long-short-term memory-based model to understand both short- and long-term time series patterns in a single model. The model considers the weather conditions and seasonal changes to predict PV power hourly. Results showed that incorporating information from previous hours and days improves the accuracy of the prediction. Also in [54], MLP and Elman NNs used to study the performance of these structures in power forecasting from a historical PV production. The results showed that Elman NN can perform better performance with small window data but no obvious differences between the two structures with large window data. It also indicated that both Elman and MLP NN can be used to forecast PV output Power but more accuracy can be achieved by considering more parameters such as seasons, weather data, and extending the input data to many years.

For classification purposes, ANNs were used in [55] for early detection and classification of faults in wind turbines to reduce maintenance and operating costs. The model was able to detect and distinguish three different types of errors in different operating conditions. Shi et al. [56] used short term PV power prediction model to preprocess the next day weather data and classify it into four types by exploiting the correlation between inputs belonging to the same weather conditions: Foggy, rainy, cloudy, and clear sky day. Then, one of four SVM models is used to forecast the PV power output for one day ahead depending on historical PV power data and weather data. SVM was used to transform the non-linearly separable problem into a linear one by mapping the input space into higher dimensional space. The model was designed for a specific station in China and the results show that classifying weather data enhanced the accuracy of the model especially in a

clear sky and foggy days. SOM algorithm was used in [57] for classifying the faults in renewable energy systems to facilitate the diagnose and handle quickly, reducing damage and maintenance costs. In [58] SOM also was used for the classification of the intensity of solar radiation into different degrees, where the solar radiation per hour is divided into time periods of five minutes in length, and its intensity is determined, accordingly, an appropriate prediction method is determined. In this way, artificial intelligence algorithms were able to achieve greater accuracy in predicting the intensity of solar radiation, on which the photovoltaic output depends. Sheela et al. [59] used supervised and unsupervised ANN techniques to improve the accuracy of wind speed prediction by using SOM and MLP. SOM is used to classify the normalized input data into four clusters depending on the similarity between data due to season changes. Then, four MLP networks were used to produce predicted wind speed depending on wind direction, speed, and temperature which affect the accuracy of wind power in the farm. RMSE criteria were used to evaluate the performance. The results comparing with conventional NN methods like RBF and BP and conventional MLP show enhanced performance with least RMSE. Most previous work in this area is concerned with the prediction of a certain region using metrological data, and does not consider the power prediction of multiple regions; although the sum of the power generation is the key to manage the energy systems. For this reason, through the thesis, we used metrological and historical power data for the prediction of several regions in Palestine to evaluate the situation of the REs in the West Bank. Also, we used SOM to classify the regions according to its potential RE production.

Summary

During this chapter, the most important details related to RE were presented, where RE sources and their advantages, especially solar and wind energy, were discussed. The problems facing traditional energy in Palestine and the prospects open by RE in the Palestinian territories were also

highlighted. The process of producing electric energy from renewable resources has also been clarified, and the factors that affecting the amount of energy produced and the efficiency of RE systems have been clarified. We also represent the dataset that will be used for predicting the potential power production from RE in Palestine. Also, we review the related works in the field of predicting and classifying the RE production using ANNs. In the next chapter, we will discuss the importance of the prediction and classification in improving RE systems (Photovoltaic system and wind system) and its methods.

Chapter Three: Prediction Methods

Introduction

RE sources are intermittent sources and depend on constantly changing factors such as weather conditions. Therefore, the continuous increase in the use of RE, as an alternative or auxiliary to traditional energy sources, requires the accurate prediction of the potential amount of energy that can be produced through a specific system in a specific geographical area and during a specific time period. The productivity of the PV system is affected by several factors, the most important of which is the intensity of solar radiation, and the production of wind turbines is mainly affected by the speed of the wind. Despite the direct relationship between the electrical energy production on one side and the intensity of solar radiation and wind speed, on the other hand, the complex nature of these two variables makes the process of predicting their value complex. This, in turn, makes it more challenging to calculate the amount of energy that can be produced. Accurate prediction of the amount of these variables helps in improving the effectiveness of RE systems, which increases the reliability and economic viability of these systems. The accurate forecasting of the amount of potential energy produced provides a tool that makes it easier for power farms operators to manage and organize their work effectively and efficiently. It also helps in improving the process of integrating these alternative energy systems within electricity grids and managing the process of loads distribution. This chapter presents the most common methods used to predict the potential energy that can be produced based on the main factors affecting it such as wind, solar radiation, temperature, and others.

Usually, one of four methods is used to predict the basic factors of energy production (wind and solar radiation) or to predict productivity. They are classic statistical methods, theoretical calculation, methods of AI and NNs, and algorithms that combine two types [60].

3.1 Statistical Methods

Statistical methods usually use time-series methods for prediction, which are methods that depend on gathering observations of variables in consecutive and regular time periods [61]. Time is one of the variables in these methods. Time series use mathematical models to predict future values based on previously collected observations. Among the main statistical methods used in predicting solar and wind energy are the following:

- Simple Average (SA): The simplest method of prediction that expresses the expected value as the average of all previously observed points [62]. It is an appropriate solution for some cases despite its low accuracy.
- Moving Average (MA): This method predicts the next value by calculating the average of the values in a sliding window in the time period immediately before the prediction point instead of taking the average of all previous observations [63].
- Weighted MA (WMA): This method uses different weights for the values inside the sliding window depending on their importance in prediction [64].
- Simple Exponential Smoothing (SES): WMA with assigning exponentially decreasing weights to all observations to give the lowest weights to the oldest values [64].
- Autoregressive (AR): This method predicts the next value as a linear combination of past values and noise.
- AR combined with MA (ARMA): The value of the prediction is the sum of the outcomes of the two methods [65].
- Autoregressive MA with exogenous variables (ARMAX): Several exogenous variables can be used to improve the accuracy of the prediction [66].

- Autoregressive integrated MA (ARIMA): This model takes into account the correlations and similarities between the data. It is used in non-stationary time series as data behavior changes after long periods [65].
- Naive approach: The predicted value takes the same previous value of the observations [67].

3.2 Theoretical Calculations Using Weather Conditions

One of the less common methods of predicting the amount of potential energy is the physical method, which depends on calculating the amount of energy mathematically through the main factors affecting it. This requires predicting future values of influencing factors such as wind speed and solar radiation, in addition to the need to know information about each system, such as its capacity, design, and other indirect factors, which were previously mentioned. Also, physical methods can rely on satellite images and observed data for the calculation. In this section, we present some of the mathematical equations used to calculate both solar and wind energy, which we used to verify the effectiveness of our model.

Solar Energy Calculation

Solar radiation is considered one of the most important factors affecting the amount of energy produced from solar panels, in addition to the surface area of the solar panels and the size of the solar system as a whole. However, several factors affecting the system must be taken into consideration such as the effectiveness of solar panels, the degree of tilt and facing of the panels for the sun, the ambient temperature of the panels, and the photovoltaic system, the system life, and maintenance conditions. To calculate the potential energy production from the solar system we can use the global formula that is:

$$E = A \times R \times H \times PR \qquad (1)$$

Where E is the energy in kWh, A is the solar panel area, R is the efficiency of the panel, H is the average solar radiation, PR is the performance ratio of the system [68].

In our model we used a more specified formula that is involved the ambient temperature [69], as follows:

$$P = 0.1 \times G \times [1 - (25 - T) * (-0.0037)] \qquad (2)$$

Where P is the power production, G is the global radiation; T is the ambient temperature in ^{o}C.

Wind Energy Calculations

During the previous chapter, we clarified the most important factors affecting wind energy production; Such as wind speed and direction, air density and temperature, turbine size and design, obstructions from wind flow, and other direct and indirect factors. Many mathematical equations link these elements to reach the amount of potential energy. But the most important points to consider is the following:

- The power output from the air turbine is directly affected by the wind speed cube.
- The power output of the wind turbine is directly related to the region in which the rotator rotates and which is related to the length of the blades.
- The turbine design also affects the efficiency of converting kinetic energy into rotational energy and then electrical energy. Three basic design factors directly affect the amount of power, which is the efficiency of the rooter, generator efficiency, and gearbox efficiency.

Thus, we can derive the power generation formula as follows [70]:

$$E = 0.5 \times M \times V^2 \qquad (3)$$

Where E is the kinetic energy in joules, M is the air mass in kg, V is the wind velocity in m/s. The mass of air hitting a given area is expressed by the following formula:

$$M/sec = V \times A \times \rho \qquad (4)$$

Where M/sec is the mass of air hitting the swept area in each second, in Kg/sec, A is the swept area of the rotor in m2, ρ is the air density in (kg/m3). And whereas power is the amount of energy per second, thus the power formula can be derived from the previous formulas 1 and 2 as follow:

$$Power = 0.5 \times A \times \rho \times V^3 \qquad (5)$$

Other factors that affect the efficiency of the turbine can be added to the formula as follows: efficiency of the router, generator efficiency, and gearbox efficiency.

$$Power = 0.5 \times N_p \times N_g \times N_b \times A \times \rho \times V^3 \qquad (6)$$

Where N_p is the router efficiency that indicates the performance of the rotor. N_g is the generator efficiency, N_b is the gearbox efficiency.

The swept area is the area of the circle created by the blades as they sweep through the air. To calculate the swept area of the rotor for a vertical-axis wind turbine we can use the following formula:

$$A = \pi \times d \times h \qquad (7)$$

Where d is the diameter of the rotor, h is the height of the rotor .Whereas, swept area for a Horizontal-axis wind turbine is:

$$A = \pi \times r^2 \qquad (8)$$

Where r is the radius of the rotor.

Inaccuracies in statistical and calculation methods largely arise when the model cannot track the shape of the input data due to the fluctuations of the weather factors. The use of artificial neural networks to blend multiple prediction model inputs can help increase accuracy of prediction.

3.3 Artificial Intelligence Methods

Statistical and physical methods demonstrate their weakness in tracking the behavior of weather conditions and power generation due to the fluctuations in the weather conditions and the other parameters that affect the power generation like dust, humidity, cloudiness, and temperature. So,

accurate prediction is necessary for RE systems management and scheduling. This requires the use of more accurate methods such as AI and NNs, which are characterized by their ability to predict complex patterns of data through self-learning that provide high accuracy and reliability. Many AI methods are used to obtain the accuracy of prediction. ANN is one of the most common methods of the AI techniques for prediction, as well as Support Vector Machine (SVM). The basic idea of support vector machines (SVM) is to map the data into a high dimensional feature space via a nonlinear mapping and to perform a linear regression in this feature space. Kazem et al. [71] and Mohandes et al. [72] used SVM for daily solar energy prediction and wind speed prediction respectively. Other approaches combine the classical statistical methods with AI techniques to enhance the accuracy of the prediction. In [73], and [74], a combination of the ARMA model and ANN for long- and short-term prediction of the solar system is used, the results show higher accuracy than the individual models. Paoli et al. [75] present a hybrid model of AR and ANN which perform better than the ARMA model. Cadenas et al. [76] presents a hybrid model used to enhance the accuracy of wind speed prediction by combining ARIMA and ANN.

3.3.1 Artificial Neural Networks

ANNs tool is one of the information processing tools for machine learning that is inspired by the biological nervous system's way of data processing and decision making. ANNs consist of a large number of interconnected processing units called neurons that work in harmony to solve a specific problem as in the human nervous system. Neurons work to calculate the outputs from the inputs in parallel processes as the neurons receive several inputs and calculate one output of the network. ANNs are distinguished by their ability to learn from previous experiences and mistakes. The learning process is divided into two types: supervised learning and unsupervised learning. In supervised learning, ANNs learn using training data which is a labeled dataset of input and output

to predict outcomes for unforeseen data. An unsupervised model, in contrast, provides unlabeled data that the neural network tries to extract the features and patterns from them [77]. ANNs are widely used in many sectors like engineering, medical and industrial sectors for several applications including classification, computer vision, time-series processing, feature extraction, and recommendation systems. Also, they are used in predicting the production of electrical energy, either using the previous observations of energy production that are used to train the network and to measure its performance or through the use of weather conditions data affecting the production of RE such as solar radiation, temperature, wind speed, and cloudiness. They are used to forecast production in the short term as a day or even for a few hours, which contributes to the process of distributing and planning the system production to be able to provide the required production and avoid the shortage. They are also used to forecast production in the medium and long term, i.e. for a month, several months, or even years, which helps in determining the capacity of the system before installation and estimating its productivity.

The main goal of the thesis is to build a hybrid system of ANNs to improve its ability to predict the outputs of RE systems over a year based on climate data and historical data for energy systems. It also aims to classify the Palestinian territories according to their production capacity from solar and wind energy; which aims to raise governmental and private sector awareness of the potential offered by these sustainable sources and their ability to create feasible and economic solutions to the electrical energy sector. Additionally, it helps in identifying the best places to install RE systems in Palestine to reduce the risk of project failures as high-cost projects. Throughout the next sections, we address a group of neural networks with their different structures on which the thesis was based. Also, we discuss some algorithms used to improve the structure of artificial neural networks. And, we explain the methodology that we adopted in the thesis to improve the

performance of the neural network in the process of predicting and classifying the production of renewable energy in Palestine.

3.3.2 Artificial Neural Networks Structure

ANNs are superior tools used to reach very complex patterns that the human programmer cannot extract and teach the machine to recognize it. These are brain-inspired systems designed to mimic the way people learn. ANNs consist of the input and output layers, interspersed with a hidden layer consisting of processing units that are connected to configure a network [78]. ANNs contain thousands of synapses, and the neuron can communicate with hundreds of other neurons to receive signals from them. The importance of connections between neurons is determined by weights, with each contact taking a different weight from the other based on the signal produced by the neuron. These weights and how they are attached to neurons are the basis for the synthesis of ANNs. Figure 8 shows the simple mathematical representation of the artificial neuron; the main component of ANNs for all different types.

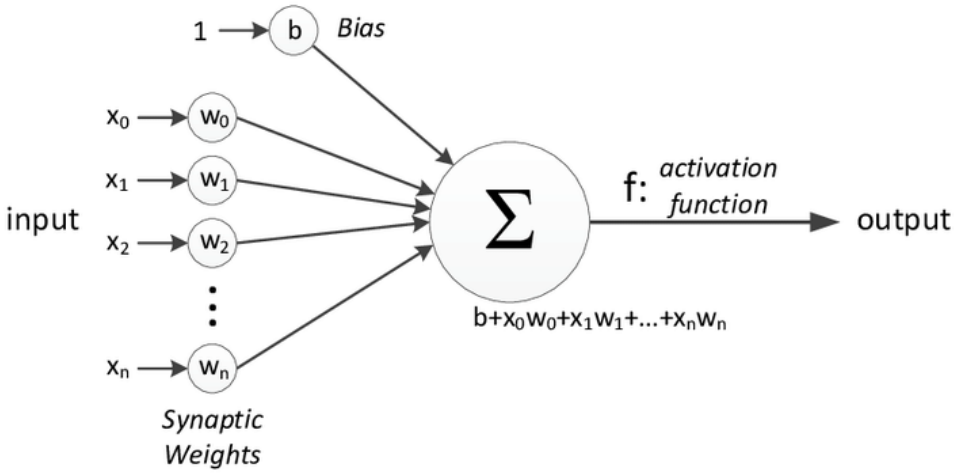

Figure 8 Simple Mathematical Representation of Neuron. [79]

ANN is a network of nonlinear elements connected by adjustable weights. The figure shows a neural network consisting of one neuron, which is the simplest form of the ANN, called Perceptron.

Perceptron consists of input and output layers, with one processing unit (Neuron) connected to the two layers. The perceptron multiplies the first layer inputs by the synaptic weights, where all input refers to one (independent) variable of the dataset, and all weight represents its connection strength. These products are summed, fed to a transfer function (activation function) to obtain a result, and this result is sent as output. The role of the activation function in the network is to convert the input signal to outputs, which in turn are used as inputs for the next layer to activate its neurons. The next equation forms the output of the perceptron:

$$Y = f(\sum xw_{ij}.x_i) + b \qquad (9)$$

Where f is the activation function of the perceptron. And b is the bias value to shift the activation function up or down [80].

The activation process is important for learning the NNs and without it, the processes in neurons become linear, which reduces the performance of the network and weakens the learning process upon which the complex data processing relies. NNs are considered as "Universal Function Approximations" because they can learn any function, no matter how sophisticated it is, which gives strength to neural NNs in solving non-linear and complex problems. The most common activation functions used in NNs are [81]:

- Step or Threshold function: The threshold function activates the neuron if the input is greater or less than a certain value. The same value is sent to the next layer. The problem with this function is that it only takes two values (0.1). This makes it difficult to use it in classification for more than two classes. Figure 9.

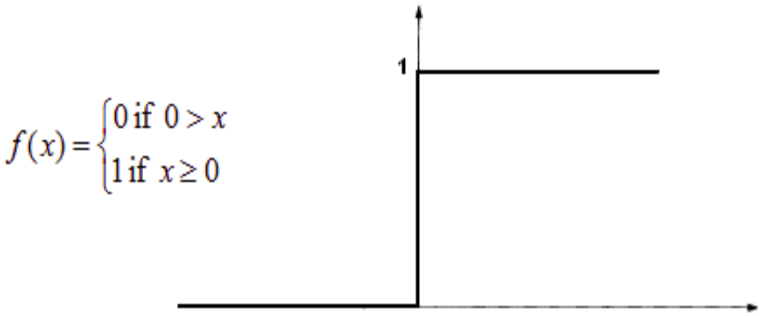

Figure 9 Step Function

- Sigmoid function: It is a function that takes the "s" letter curve, and whose value is between 0 and 1. This function is suitable for tasks that require the prediction of probability as output. The drawback of this function is at the negative values, as it gives a value of 0 for all negatives. Figure 10.

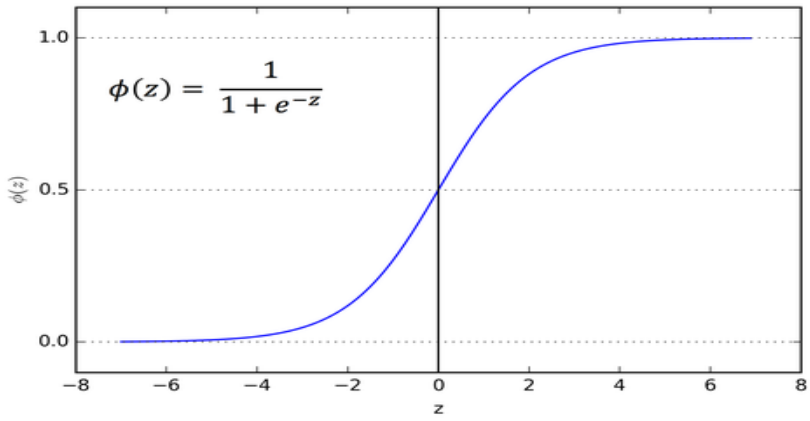

Figure 10 Sigmoid Function

- Hyperbolic Tangent function: This function comes as an enhancement to the performance of the sigmoid function. The values of this function range from 1 to -1 so that it can handle negative inputs by giving negative outputs. Figure 11.

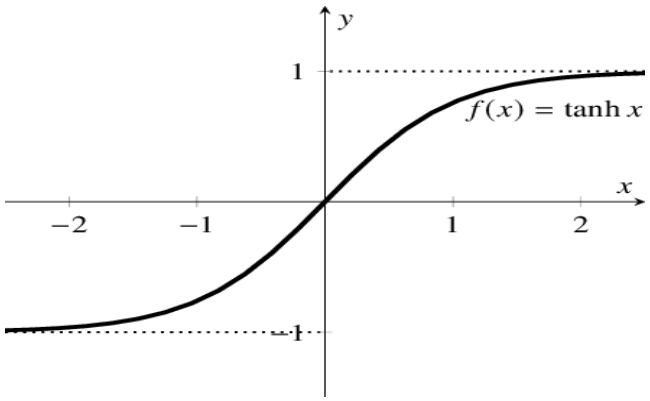

Figure 11 Hyperbolic Tangent Function

NNs learn by comparing the predicted results (output) with the actual values provided by the training data. The difference between the actual and expected value is calculated by the cost function, a function that gives us the value of the error in prediction for each layer in the network. This error is returned to the system to adjust the weights several times called iterations so that the error amount is minimal. In this way, the closer the prediction value to the actual value, the value of the cost function is less and is minimal when the predicted value is close enough to the actual value. This method is called backpropagation learning [82]. The process of adjusting weights depends on the slope of the cost function curve. This curve gives different weights of values during the training process so that the cost function is minimal. By calculating the slope of this function, it is possible to find the appropriate direction for adjusting weights (increase or decrease). This method is called Gradient Descent (GD) learning [83]. The negative slope of the function indicates that the function decreases i.e. the prediction value approaches the required value while the positive slope indicates the opposite.

3.3.3 Types of Neural Networks

ANNs are computational models that work similarly to the work of the human nervous system. As mentioned earlier, NNs are divided according to the learning method into two types: supervised and unsupervised networks. Supervised networks are used in forecasting operations often because

of the dependency on a teacher or target for training. Whereas, supervised networks are used in classification and clustering applications because it involves finding common characteristics and properties between the inputs. There are many types of artificial neural networks that fall under these two groups. As each type differs in its structure and the parameters required to determine the output.

3.3.3.1 Feed Forward Neural Network

It is one of the simplest structures of ANNs. The data is propagated in one direction, through the input layer, to the output layer. This type of network is suitable for classification applications such as computer vision and speech recognition; where the goal of the network is the classification into categories. The network consists of the input layer and the output layer in addition to one or more hidden layers. Feedforward networks contain a large number of neurons connected through weights. Types of neurons are divided into three sections: the input neurons that receive the inputs, the output neurons that show the outputs, and the hidden neurons that are responsible for the processing. Outputs are calculated from the inputs by computing the weighted sum of the inputs and then classifying the results based on the activation function. For example, if the threshold value for the activation function is 0, any value greater than 0 will give an output of a value of 1. If the value of the weighted sum is less than 0 then the output value is 0 or -1 depending on the type of function used. The outputs that take a value of 1 lead to activation of the neuron connected to the next layer if it is a multi-layer perceptron feedforward network (MLPNNs) as in figure 12.

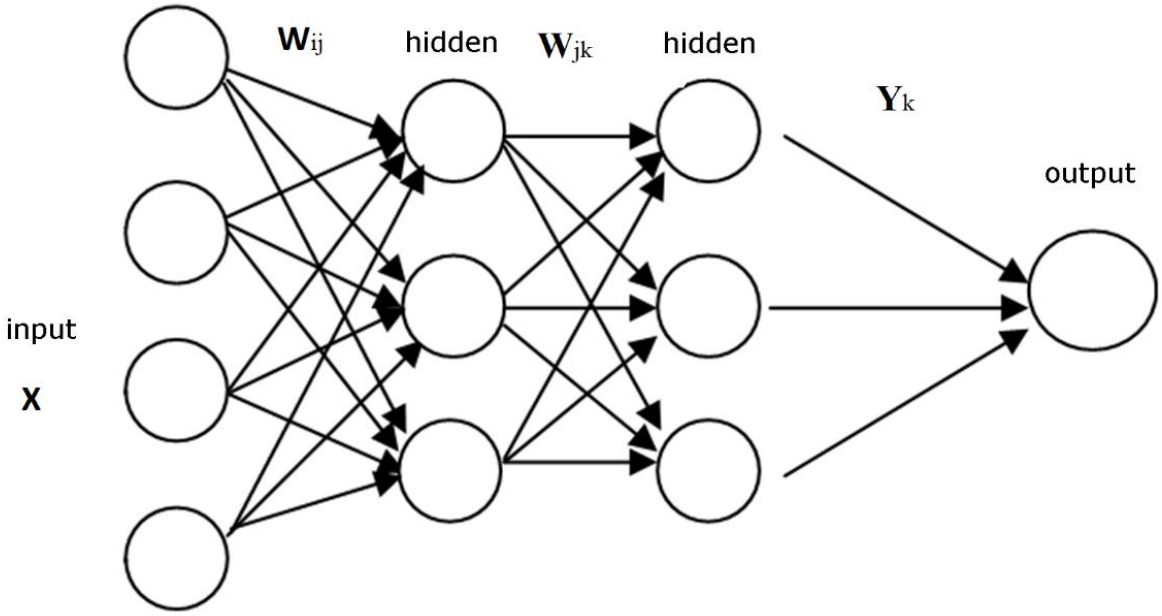

Figure 12 Multi-Layer Feedforward NN

The mathematical representation of the MLP NNs for a network of three layers, feedforward, multi-input multi-output as follows [84]:

- Input layer X_i , i = 1, 2, …, n. Where n is the number of input nodes.
- Hidden layer j: Each node is a processing unit (neuron), connected to the input layer by the weights w_{ij}, where i is the input node and j is the hidden layer node.
- Output layer k: Contains the nodes that produce the output of the network, Y_k.

When the training pattern fed to the input layer, the weighted sum of the input to the j^{th} node in the hidden layer is given by

$$y = \sum W_{ij} X_i + \theta_j \qquad (10)$$

Where θ_j is the bias node that always has a value of 1. MLP NNs use the backpropagation algorithm to calculate the gradients efficiently. This algorithm starts from the output layer and propagates backward to update the weights of the network. It requires the activation function to be differentiable; the sigmoid function is typically used because it makes it easy for the model to

generalize or adapt with variety of data and to differentiate between the output. Thus, the actual output of the j^{th} node is

$$Y_j = X_k = \frac{1}{1+e^{-y}} \qquad (11)$$

X_k is the input to the next layer's node. At the output layer, if the actual value of the node k is Y_k, and the target value is t_k, the difference between them is

$$\Delta_k = t_k - Y_k \qquad (12)$$

The error signal δ_k of the output layer can be calculated by the derivative of the sigmoid function and Δ_k; so,

$$\delta_k = \Delta_k Y_k (1 - Y_k) \qquad (13)$$

Using the delta rule, we can find the change in the weight between node j and node k by multiplying the error at node k by the output of node j,

$$\Delta w_{jk} = l \, \delta_k X_k \qquad (14)$$

where l is the learning rate. So, the weight w_{jk} between node j and k can by updated by the following formula:

$$w_{jk} = w_{jk} + \Delta w_{jk} \qquad (15)$$

To calculate the error signal δ_j for node j in the hidden layer,

$$\delta_j = (t_k - Y_k) Y_k \sum w_{jk} \, \delta_k \qquad (16)$$

As before, to modify the weight, $w_{i,j}$ between the input node, i, and the node, j we use 14 and 15 so

$$\Delta w_{ij} = l \, \delta_j X_j \qquad (17)$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \qquad (18)$$

The backpropagation algorithm repeats until the error on the output node is minimized. The general formula that is used for calculating the error E is the mean square error:

$$E = \frac{1}{N} \sum \sum (t_k - Y_k)^2 \qquad (19)$$

3.3.3.2 Radial Basis Function Neural Network

This network is characterized by its use of radial functions instead of a threshold function. This function takes the bell curve. The factors that must be specified for this function are: center, weight, and spread constantly. The spread constant is proportional to the nature of the data used. A function with a small width is required for rapidly changing data, whose curve contains consecutive peaks and bottoms. But for the data that is characterized by its smooth flow and flatness, we give the function a larger width. The process of building a network is carried out in two stages: The first is the stage of choosing the width and centers of the hidden layer neurons. The second stage is to find the appropriate weights between the hidden layer and the output layer through training. The process of determining the number of neurons in the hidden layer is one of the most challenges facing the design of this neural network [85]. The number of neurons controls the complexity of the network and its ability to generalize. A few neurons number simplify the network's structure, but it cannot fit the training data adequately due to limited flexibility. The very large number of neurons may affect the generalization process so that the network becomes unable to handle new data [86].

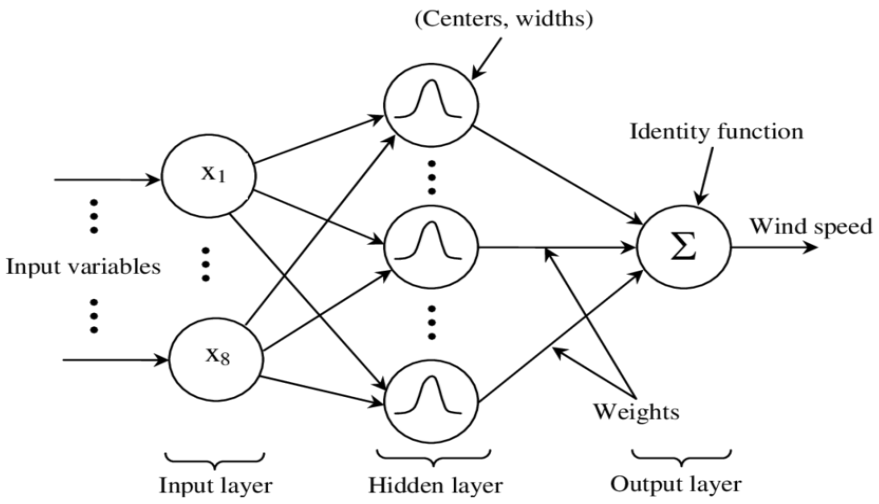

Figure 13 RBFNNs [87].

The structure of RBFNNs similar to the MLPNNs; consists of three layers which are the input layer, the output layer, and one between them is the hidden layer as in figure 13. The hidden layer implements radial activation functions instead of the sigmoid ones which activate by the input data entered by the input layer depending on the value of the activation function. The radial function is usually a Gaussian function Φ.

$$\Phi_j(x) = exp\,\frac{-\|x-\mathcal{M}\|^2}{2{\delta_j}^2} \qquad (20)$$

Where $\|x-\mathcal{M}\|$ is the Euclidean distance between the input vector x and the hidden node j which has the center $\mathcal{M}$ and spread constant δ of its Gaussian function. The network classifies by calculating the Euclidean distance between the point and the center so that the point ultimately belongs to the nearest center, or in other words the center that is within its diameter. It also predicts using a set of functions that take center upon samples of data used in training. Each output neurons implement a weighted sum of hidden neurons output to generate the whole output of the network Y(x) [87].

$$Y_k(x) = \sum_1^N w_{jk}\,\Phi_j(x) \qquad (21)$$

where N is the number of activation functions of the hidden layer, w_{jk} is the weights vector between the hidden and output layers. To adjust the weights of the network depending on the difference between the output $Y_k(x)$ and the target $t_k$using the formulas 22 and 24, [86].

$$E = \sum_x \sum_k (t_k - Y_k(x))^2 \qquad (22)$$

$$E = \sum_x \sum_k (t_k - \sum_1^N w_{jk}\, \Phi_j(x))^2 \qquad (23)$$

Then, the update of the weights depends on the derivative of E as follows

$$\Delta w_{jk} = -\mathcal{M} \frac{dE}{dw_{jk}} \qquad (24)$$

So, the modified values of the weights become

$$w_{jk} = w_{jk} + \Delta w_{jk} \qquad (25)$$

3.3.3.3 Recurrent Neural Network

This network consists of several layers similar to the MLP. But what distinguishes it is the use of memory to save the output and then return it to the inputs to help in calculating the output through feedback loops from the output to the input. One of the most common uses of this network is to convert speech to text and handwriting recognition. The network initially works on finding the outputs of the hidden layer by calculating the weighted sum of the inputs as it happens in the Feedforward network. Neurons retain the resulting value of the layer for the following iterations, which makes neurons appear as they have a memory in the processing. For the input x(t) of the network, the output of the hidden elements at time t is

$$h(t) = f(Ux(t) + Wh(t-1)) \qquad (26)$$

Where *f* is a nonlinear function, as a tangent function, *U* is the input to the hidden layer weights matrix, *W* is hidden to the hidden layer weights matrix, $h(t-1)$ is the output at time t-1.

$$f(t) = tanh(x(t)) \qquad (27)$$

Where $f(t)$ is the activation function of the input $x(t)$. Data processing continues to pass from the hidden layers to the output layer. And the output of the last layer is o(t),

$$o(t) = V(h(t)) \qquad (28)$$

Where V is the weights matrix of hidden to output layers. If the result of the prediction is not correct, that is, there is a big difference between the results and the target, the learning rate is used to make adjustments to reach the required by using the backpropagation [88]. In unsupervised learning RNNs, the fitness function is used instead of the target to evaluate the network performance. Figure 14 is the recurrent network.

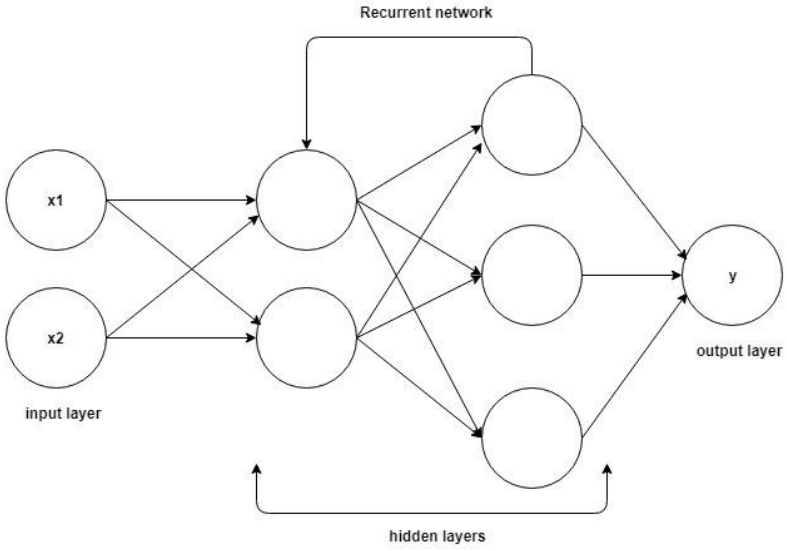

Figure 14 Recurrent NN [89].

3.3.3.4 Kohonen Neural Network: Self-Organize Mapping

Classification of RE systems can help in organizing a geographic region as a country or a continent by its potential for energy production depending on its metrological and weather conditions. This, in turn, contributes to the development and expansion of RE projects to reduce dependence on traditional energy sources. Many efforts have been made to achieve this goal, whether locally or globally. One of the most important means of decision-making is the methods of AI, especially

NNs. It is used to classify data and sort them according to their characteristics, which will facilitate decision-making and problem-solving. SOM is a popular ANNs algorithm used for classification. The main property of the SOM algorithm is unsupervised learning. That allows inputs to be divided into groups without prior knowledge of their similarities. In [59] SOM- Kohonen network was used to classify the data into four clusters depending on the seasonal changes. Kohonen network is an example of unsupervised learning networks used in pattern recognition and data mining. This aims to cluster the data without previous knowledge of it. This network converts outputs from a multi-dimensional plane to a two-dimensional plane, to make a special data arrangement. Figure 15 represents the feature map of Kohonen NNs.

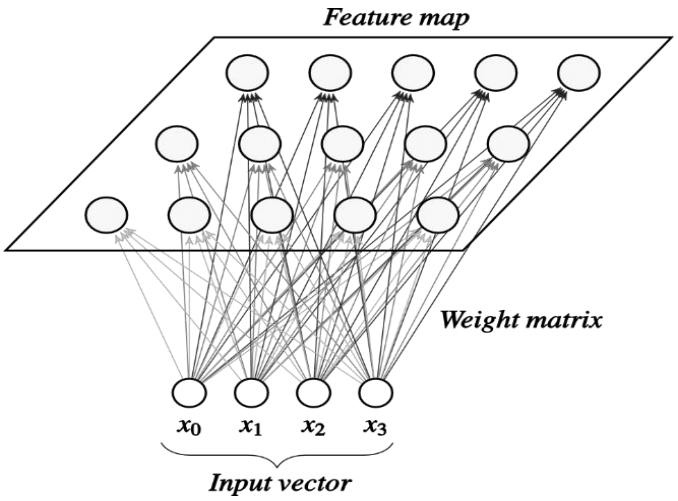

Figure 15 Kohonen Neural Network [91].

During the training process, the location of the neuron remains constant while the weight changes depending on the input value. The process is called self-organized mapping (SOM), which has four phases:

- The first phase is where each neuron is initialized by random weight.

- The second phase is the competition phase where the neurons activated with the inputs in vectors form and the winning neuron is identified, which is only one output unit that has the best-matching weight; it has weights closer to the input vector.
- The next phase is the cooperation phase where weights are modified such that neighboring neurons have similar weights.
- The last phase is the adaptation phase where similar inputs are linked with winner neurons that are closed to each other, while winner neurons for different inputs are located far away in the network.

For an input pattern x = $\{x_i: i = 1, \ldots, D\}$ with D dimensional space, $w_j = \{w_{ji} : j = 1, \ldots, N; i = 1, \ldots, D\}$ weights between the input units i and hidden units j where N is the number of the hidden neurons; to calculate the distance between each input vector and neurons in the competition phase, Euclidean distance is usually used [90].

$$d_j(x) = \sum_{i=1}^{D}(x_i - w_{ji})^2 \qquad (29)$$

Where $d_j(x)$ is the Euclidean distance between the input x and neuron j, x_i is the input, and w_{ji} is the weights between the input units i and hidden units j. This process is repeated until all the related- feature inputs are clustered around a winner neuron representing this feature depending on the distance between the input vector and the weights vector of the winner neuron. At the cooperative phase, the topological neighborhood $T_{j,I}(x)$ is defined by calculating the lateral distance between neurons i and j on the grid of neurons, S_{ij}.

$$T_{j,I(x)} = exp(-S_{j,I(x)}^2/2\sigma^2) \qquad (30)$$

Where I(x) is the index of the winning neuron, and σ is the size of the neighborhood which is decreasing by the time, so

$$\sigma(t) = \sigma_0 exp(-{}^{t}/_{\tau_\sigma}) \qquad (31)$$

Where σ_0 is the initial size of the neighborhood, t is the time and τ_σ is the decreasing factor. At the adaptation phase, the feature map between inputs and output is created and inputs to the network are automatically classified on the map by updating the weights of the winner neurons and its neighbors using the following formula to move the weight vectors of the winning neurons towards the input vector x [91].

$$\Delta w_{ji} = \eta(t).T_{j,I(x)}.(x_i - w_{ji}) \qquad (32)$$

Where Δw_{ji} is the weights update, $\eta(t)$ is the learning rate, $T_{j,I(x)}$ is the topological neighborhood.

3.4 Evolutionary Algorithms for NNs Optimization

One of the most prevalent methods in the learning process for feed forwarding networks is backpropagation, which uses the GD algorithm to reach the best parameters for the network. GD requires defining a cost function to calculate the difference between output and target. It depends on gradually reducing the error by adjusting the weights several times to reach the best values for the weights so that it gives the lowest value for the cost function and the best value for prediction. Although GD is one of the strongest methods of learning, it faces challenges during training that may slow down or stop the learning process without reaching the best weights, which affects the performance of the network. This problem arises when the cost function takes a flat curve such that the slope becomes 0, which is trapped in the training at the local minimum values of weights. This affects the learning process and thus the improvement of the weights. On the other hand, the cost function curve may take a saw tooth shape due to the difference in the gradient value for different samples of training data, which makes the learning process much slower to reach the global minimum. There are several studies, such as [92] that have been directed to the use of evolutionary algorithms (EAs) in the training of ANNs. EAs are computational algorithms inspired

by biological evolution. The most famous of which is the genetic algorithm (GAs). GAs can reach a set of local minima, and thus increase the likelihood of reaching the global minimum. This feature of the GAs can be used to reach the structure and weights of NNs. Another EAs is the particle swarm optimization (PSO) algorithm that is inspired by the social behavior of the birds' swarms.

3.4.1 Genetic Algorithms (GAs)

GAs is a random search algorithm used to reach the best improvements to networks by relying on the advantage of natural selection and the principle of competition between individuals, taken from population genetics [93]. The multiplication of individuals leads to the diversity of their genes, and the best individuals can be chosen by relying on their genes. The successive selection of the best individuals leads to reaching superior individuals according to the rule of survival for the better and by using the principle of randomized exchange of genes. The new generation of individuals is built based on previous generations. To solve a problem using GAs, five major steps should be implemented. The algorithm randomly proposes a population, calculate the fitness of each chromosome, and then performs a series of successive operations to generate new chromosomes better than the previous one. These operations include Selection, Crossover, and Mutation [94].

- Initial population: The first step is to define the problem variables and suggest a randomly set of initial solutions called chromosomes. The population of size N contains N chromosomes with a fixed length of genes.

$$\text{Initial population:} \quad x_1, x_2, \dots, x_N$$

- Fitness function: $f(x)$ is a function that is used to determine how close a particular design solution is to achieving group goals. In this phase, each chromosome gets a fitness score. The GAs selects chromosomes based on these values to direct the training process towards

optimal solutions. For the prediction problems, the fitness function is the error minimizing function which represents the difference between the target and the output.

$$f(x) = \frac{1}{2} \sum \sum (t - y)^2 \qquad (33)$$

Fitness scores: $f(x_1), f(x_2), \dots, f(x_N)$

- Selection: This process depends on the fitness scores on selecting the fittest chromosomes to pass their genes to the next generation, as these scores determine the quality of the chromosome and therefore the chromosomes that have the highest fitness score have a greater probability of producing a new generation of chromosomes. To calculate the probability p_{x_i} of the chromosome x_i:

$$p_{x_i} = \frac{f(x_i)}{\sum_j^N f(x_j)} \qquad (34)$$

- Crossover: A process in which the last generation of selected chromosomes is mated to produce a new generation of chromosomes. The crossover process selects the cross-point randomly to produce a new pair of chromosomes, as the new chromosomes possess the characteristics of parents, which are expected to be improved characteristics for children.
- Mutation: It is a one gene random flipping within the chromosome itself to ensure that the important information of the chains is not lost prematurely and the training process does not trap in the local minimum value.

Figure 16 also shows the process of producing offspring from parents using the crossover and mutation processes.

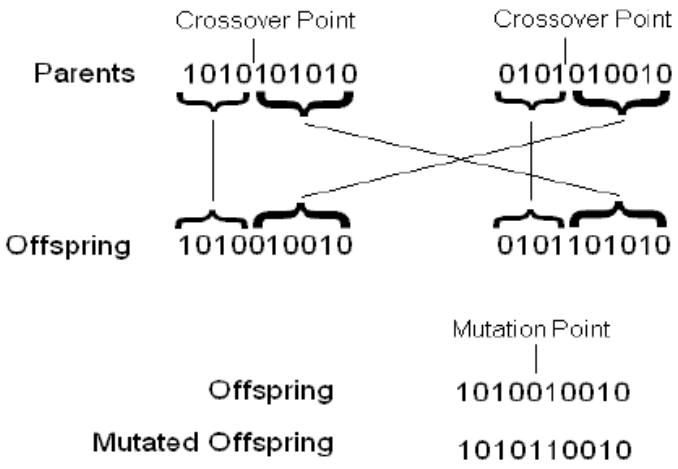

Figure 16 Reproduction Offspring [95].

3.4.2 Particle Swarm Optimization (PSO)

PSO algorithm is an optimization algorithm inspired by the social behavior of flocks of birds in search of food. It is used to improve nonlinear functions and solve complex engineering problems. And it depends on the experiences of individuals to reach the best position using the best individual. Flocks of birds choose the best spot on the ground to land on, which is characterized by abundant food and few risks to the swarm. The birds continue to move for some time until they find this spot and then collectively land. The spot on the ground is discovered by a member of the swarm that can go down to the ground to examine the spot and then gives the information to the swarm to go down in a group. Everyone uses predefined criteria to find the best location for the swarm and exchange information with all swarm members so that they can determine the most appropriate location.

Through this social behavior, in 1995, [96] Kennedy and Eberhart simulate this phenomenon and design the PSO algorithm. PSO algorithm can escape from the local minimum values into which GD is suffered to find the best neural network parameters. To use this algorithm for solving

minimization problems, the population is expressed as a swarm, where each swarm consists of several particles or individuals. Each particle has a position and velocity expressed by vectors.

$$a_i(t) = [\, a_{i,1}(t), a_{i,2}(t), \ldots\ldots, a_{i,D}(t)] \qquad (35)$$

$$v_i(t) = [\, v_{i,1}(t), v_{i,2}(t), \ldots\ldots, v_{i,D}(t)\,] \qquad (36)$$

Where, $a_i(t)$ is the position of particle i at time t, $v_i(t)$ is the velocity of particle i at time t, i is the number of particles in the swarm from 1 to s, where s is the swarm size. D is the dimension of the search space. During the movement of the particles, the best position $p_i(t)$ of each particle is recorded,

$$p_i(t) = [\, p_{i,1}(t), p_{i,2}(t), \ldots\ldots, p_{i,D}(t)\,] \qquad (37)$$

Also, the best position of the swarm is updated,

$$g(t) = [\, g_1(t), g_2(t), \ldots, g_D(t)\,] \qquad (38)$$

Using $p_i(t)$ and $g(t)$ we can determine the new speed of each particle,

$$v_{i,d}(t+1) = \omega . v_{i,d}(t) + c_1\, r_1 \left(p_{i,d}(t) - a_{i,d}(t) \right) + c_2\, r_2 \left(g_d(t) - a_{i,d}(t) \right) \qquad (39)$$

Where d is the velocity directions of each particle, ω is the inertia weight to control the impact of previous velocities on current velocity. c_1, c_2 is the acceleration constants. r_1, r_2 are random values between 0 and 1. The new position of the particle can be expressed by the following formula:

$$a_{i,d}(t+1) = a_{i,d}(t) + v_{i,d}(t+1) \qquad (40)$$

and the best position is:

$$p_i(t+1) = \begin{cases} p_i(t), & f(a_i(t+1)) > f(p_i(t)) \\ a_i(t+1), & else \end{cases} \qquad (41)$$

where f is the fitness function. Then, the global best position of the swarm is $g(t+1)$ for all, i from1 to s.

$$g(t+1) = min(f(p_i\,(t+1)) \qquad (42)$$

The process of the algorithm is repeated until the termination condition is satisfied [96].

Summary

Through this chapter, we discussed the most popular AI methods that have been used for the prediction and classification of the RE systems production. This includes the demonstration of ANNs structures and EAs models that we rely on to design our model. As the implementation of solar and wind systems is increasing day by day, the dependency on these systems is also increasing. More efforts are needed to enhance the precision and reliability of power prediction. So it is important to design a complete model to find the appropriate locations and predict the potential output power of solar and wind systems depending on the metrological data of the site. In the next chapter, we will explain the adopted methodology of our proposed model for improving the performance of RE power prediction and classification.

Chapter Four: Proposed Hybrid Model

Introduction

Predicting the amount of energy generated from natural resources requires the development of a high-precision model to ensure reliability due to the reliance of the renewable energy systems on fluctuated sources such as the sun and wind. The process of building and learning MLPNNs using backpropagation suffers from trapping at the local minimum values for network parameters and their inability to reach the optimal parameters which ensure the most accurate results in forecasting. Therefore, the use of a hybrid model for learning NNs and choose its parameters can provide better results, especially the use of EAs like GAs that can escape from falling into local minimum values. Through this chapter, we discuss the detailed requirements of the model as the data collection and processing. Then we will clarify the structure of the proposed model that we used in our thesis to enhance the performance of MLPNNs for predicting and classifying the power production of RE systems. Also, we mention the main procedure of the model and the performance metrics that we used.

4.1 Data Collection

The main challenge of solar and wind power systems is obtaining high prediction accuracy because many parameters affect the output power of the system. For this model, we used historical and metrological data for predicting and classifying the regions in Palestine. The data collected from six cities in Palestine (West Bank) to guarantee the inclusiveness and feed to SOM-Kohonen Network after applying the prediction model. Historical solar energy data collected for power prediction from a specific solar station in Jenin, Tubas, Salfit, Bethlehem, and Hebron. The required maximum temperature and solar radiation data are collected by the Palestinian Meteorological Department for predicting the potential solar energy from Jenin, Nablus, and Bethlehem. Also, we got the wind speed data for wind power prediction that are collected from

Jenin, Nablus, Bethlehem, and Hebron. Table 3 describes the distribution of the historical dataset and the number of samples for each city that are used for solar energy prediction. Table 4 contains the description of maximum temperature and solar radiation that are used for solar energy calculation.

Table 3 Historical Dataset

City	Years	Period	Training Samples	Test Samples
Jenin	4	2015-2018	1095	365
Tubas	4	2016-2019	1095	365
Salfit	3	2017-2019	730	365
Bethlehem	4	2016-2019	1095	365
Hebron	3	2017-2019	730	365

Table 4 Maximum Temperature and Solar Radiation Dataset

City	Years	Period	Training Samples	Test Samples
Jenin	4	2016-2019	1095	365
Nablus	1.5	2016-2017	365	180
Bethlehem	4	2016-2019	1095	365

Table 5 Wind Dataset

City	Years	Period	Training Samples	Test Samples
Jenin	4	2016-2019	270	90
Nablus	4	2016-2019	270	90
Bethlehem	4	2016-2019	270	90
Hebron	4	2016-2019	270	90

Daily average wind speed is used for wind energy calculation. We used four years' dataset of historical readings for each city to obtain the wind energy. Since the patterns of the wind speed are strongly variable along the years, we used the average wind power of the four years to predict the power using the GAs-MLPNNs model. Table 5 describes the dataset.

4.1.1 Data Preprocessing

After we obtained the data from the Palestinian Meteorological Department and the solar energy production sites, and before we used it in the calculation and forecasting of RE, we performed data selection and pre-processing operations to ensure that the best data was obtained. These operations of pre-processing included the following steps:

- Data selection and cleaning: The first step involves the process of selecting the best years in terms of the number of available values for each year, as well as compensating for lost and irregular values. Missing values were filled out by using the average of available values for the same time point in other years. If the value is missing in more than a year at the same time point, it is compensated for the average of previous and subsequent values in the same year.
- Normalization: Data pre-processing includes normalizing the data using Max-Min normalization where the minimum value is transformed into 0, the maximum value is transformed into 1, and every other value gets decimal value between 0 and 1. The normalization process described by the following formula:

$$Vnormalized = \frac{\boldsymbol{v-min}}{\boldsymbol{max-min}} \qquad (43)$$

  Where $\boldsymbol{v}$ is the value that could be normalized, min and max is the minimum and maximum values in the dataset respectively. This prevents the data with large values from dominating other inputs.
- Filtration: Smoothing the data involves passing the input data through a moving average filter (MAF) by the Matlab environment. MAF used for regulating the sampled data by replaces M number of neighboring samples of input at a time and takes the average of those

to produce a single output point. We used seven neighbors to keep the fluctuations of the values which is the most challenge that examines the strength of the GAs-MLPNNs model.

- Partitioning: We used the K fold cross-validation method for data partitioning to ensure the validation of the model by separating the datasets into two partitions, training, and testing data as in tables 3, 4, and 5. K folding involves dividing all the samples into k-groups of equal sized subsamples, called folds. Every data point gets to be in a test partition exactly once and gets to be in a training partition k-times. Cross-validation is used to guarantee that the model is complex enough to fit the data without causing problems on the test data. This significantly reduces under-fitting as we are using most of the data for training, and also significantly reduces overfitting as most of the data is also being used in test partition.

4.2 Proposed Model Structure

To predict the output of the solar and wind energy systems with improved accuracy, we used the GAs-MLPNNs hybrid model that relies on the EAs to improve the structure and weights of the NNs. The learning process is done in two steps: The first is choosing random weights of the MLPNNs and represent them by a chromosome using GAs. Then GAs optimizes MLPNNs to reach a set of the best network weights by ignoring weak neurons. The second stage involves training the MLPNNs to use the network for prediction and updating the weights using a backpropagation algorithm. Figure 17 is the flowchart of the GAs-MLPNNs proposed hybrid model. The input to this model is the time, the target is the time series data (observed or calculated power production), and the output is the predicted power of the system.

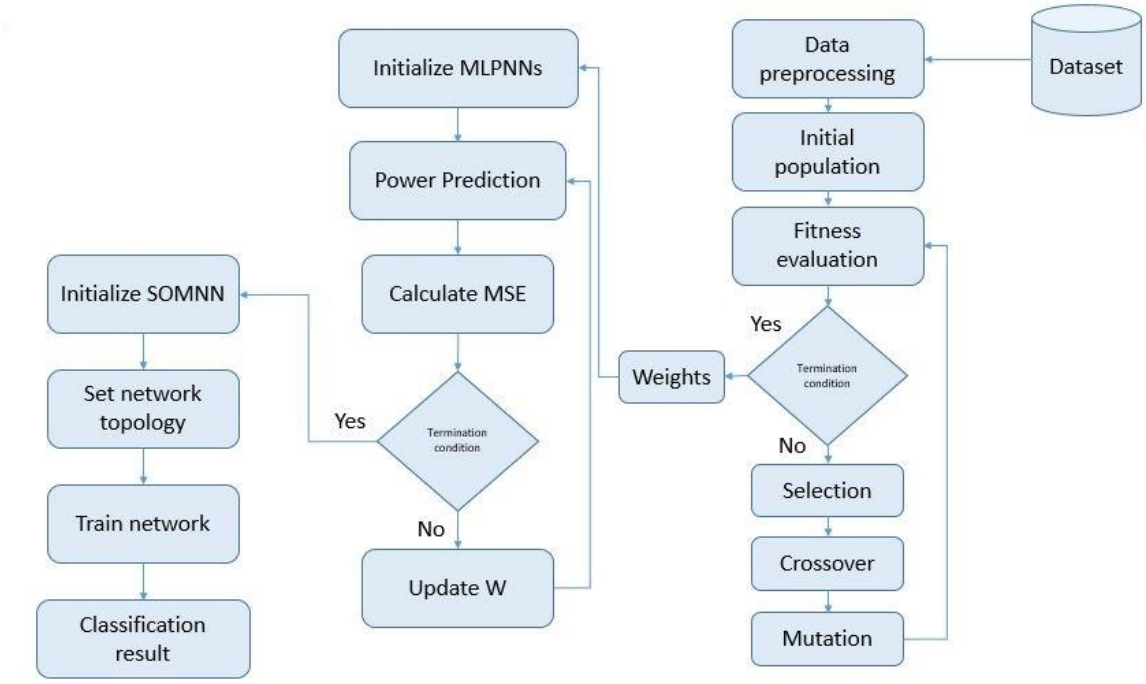

Figure 17 Flowchart of GAs-MLPNNs Model

4.2.1 Multi-Layer Perceptron Neural Networks

MLPNNs is one of the most widely used ANNs for prediction and function approximation. This network consists of three layers, the Input, the Hidden, and the Output layer. As the feed-forward networks, the neurons within the network are linked to each other using weights. Each neuron in the input layer is connected to all the neurons in the hidden layer, and so does the hidden to the output layer. The prediction using MLPNNs is done by mapping the inputs to the corresponding outputs. The network, during the training phase, reduces the error between the actual and required outputs by adjusting the weights. In the beginning, initial weights are chosen for the network, and then the weights are adjusted through the learning process, which takes place in two stages: the first is the feed-forward stage during which the outputs are produced and the error is calculated. The second stage is the backpropagation process during which the error is returned to the network to adjust the weights to obtain the minimum error in the output layer.

The network output Y is calculated using the following formula that is the weighted sum of the network inputs X:

$$Y = \sum W_{ij} X_i \qquad (44)$$

Where W_{ij} is the weight between the *i*th input and the *j*th hidden neuron. The error is calculated by the difference between the actual and desired output, and the most commonly used equation to calculate the error for MLPNNs network is MSE:

$$MSE = \frac{1}{N}\sum_1^n (Y_d - Y_j)^2 \qquad (45)$$

Where Y_d is the desired output and Y_j is the actual output of the neuron j. The training process continues until a predefined error value is reached by adjusting the network weights. The network weights W are adjusted by the following formulas:

$$W = W + \Delta W \qquad (46)$$

$$And, \quad \Delta W = l\, \delta\, X \qquad (47)$$

Where l is the learning rate, δ is the error signal from the output layer and X is the input of the network. The flowchart of training MLPNNs using backpropagation is illustrated in figure 18.

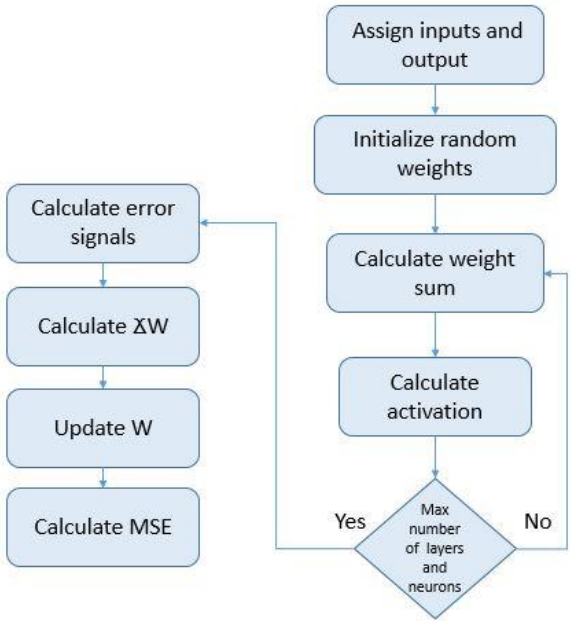

Figure 18 Flowchart of Multi-Layer Perception Neural Network

4.2.2 Genetic Algorithms

Genetic algorithms are one of the evolutionary algorithms that depend on natural selection and the principle of competition to reach the best optimization to the network. The parameters of the network are represented by the chromosome and adjusted by the GAs. Then, the result of the GAs optimization is used in the network to produce the output. The BP algorithm is used to train the network for optimal results. The following procedure describes the GAs learning process for network parameters optimization:

- Chromosome Representation and variables definition: the network parameters are encoded on a chromosome; each weight is a gene. The other variables are defined as the population size, the fitness function, the crossover, and mutation probabilities.
- Initial population: initial weights are randomly selected in a chains form, chromosomes, depending on the network parameters and the population size.
- Fitness Function: used to evaluate the performance of each chromosome.
- Selection: used to select pairs of chromosomes from the population for reproduction based on the relative fitness value of each individual.
- Crossover: the selected pairs are cut at random points and exchanged to produce a new generation (offspring).
- Mutation: a random modification of the new offspring where two genes in the same chromosome are flipped.

The process continues until getting an acceptable solution determined using certain criteria, that is the fitness value of the resultant chromosome or reaching a specific number of generations. The GAs flowchart is illustrated in figure 19.

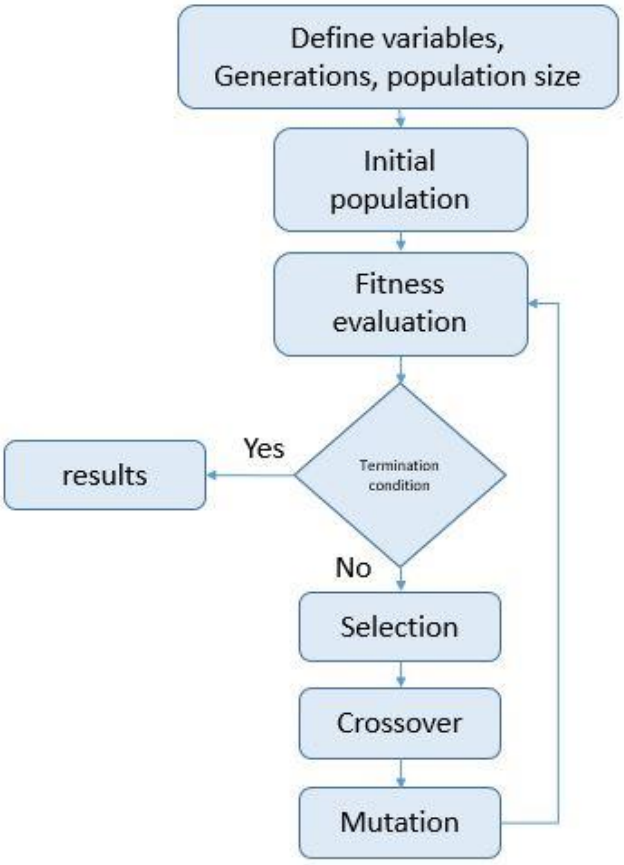

Figure 19 Flowchart of Genetic Algorithm

4.2.3 Particle Swarm Optimization (PSO)

Particle swarm optimization is evolutionary that simulates the social behavior of a swarm of birds and fish to search for a safe place where there is sufficient food for the swarm. It depends on the movement of the individual and his speed to reach the best location for the swarm. The algorithm is used to solve nonlinear and complex problems. Each solution can be represented as a particle in a solution space consisting of D dimensions called population or Swarm. Each particle in the swarm is expressed by a vector X that defines its dimensions in the search space, and the velocity of the particle is represented by a velocity vector V.

$$X_i = [\, x_{i,1}, x_{i,2}, \dots, x_{i,D}] \qquad (48)$$

$$V_i = [\, v_{i,1}, v_{i,2}, \dots, v_{i,D}] \qquad (49)$$

To find the optimal solution, the algorithm generates a random swarm at first, so that each particle in the swarm is given random velocity and position. The number of particles in the swarm depends on the number of solutions proposed in the search space and the dimension' number of space D represents the solution variables. Then, the fitness value of each particle is calculated and compared with its best fitness value. If the present fitness value is better than the previous best value, then the present value is assigned as the new best value of that particle.

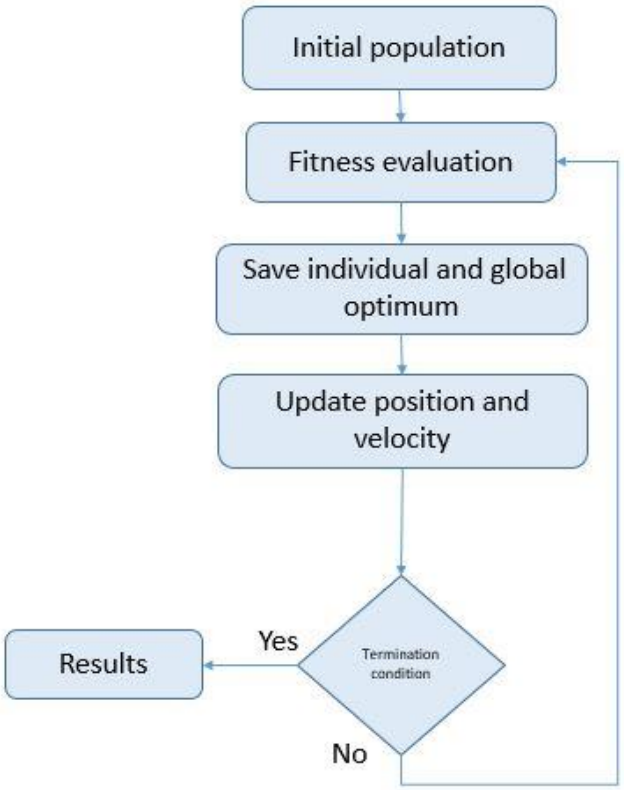

Figure 20 Flowchart of Particle Swarm Optimization Algorithm

Then the best fitness value of the individual is compared with the best value in the swarm. If the value of the individual is better, the value of the swarm is replaced with the value of the individual and assigned as the best value for the swarm. This continues for all members of the swarm until the best swarm fitness value is achieved through the individuals' values which represent the best

solution. It also contributes to reaching a predetermined number of iterations. The flowchart of the PSO algorithm is illustrated in Figure 20.

4.2.4 Genetic Algorithm-Optimization Based MLPNNs

MLPNNs are feedforward NNs that use supervised Learning algorithms of weight training to reach the optimal weights for the network. The most popular SL algorithm is the backpropagation search algorithm that looks for the minimum value of the error function by adjusting the weights using a method called the delta rule or gradient descent. So that weights that minimize the total mean square error of the network are the best solution to the learning problem. The problem with the GD algorithm is that it depends on the derivative of the error function to guide the learning process. And it is often trapped in local minima which slows down the learning speed in searching for a global minimum solution. GAs, as a global optimization tool, can be used to optimize MLPNNs weights by finding the optimal network weights and avoid the local minima problem of the BP. The MLPNNs optimization based on a genetic algorithm can be done in two phases. In the first phase, the network parameters (weights and biases) are represented as a chromosome and adjusted using GAs. Thus, searching a population of solutions for many local minima in parallel can increase the probability of finding a global minimum. In the second phase, the BP algorithm used to train the initial MLPNNs to reach optimal values of weights and biases. Figure 21 is the flowchart of GAs optimization-based MLPNNs.

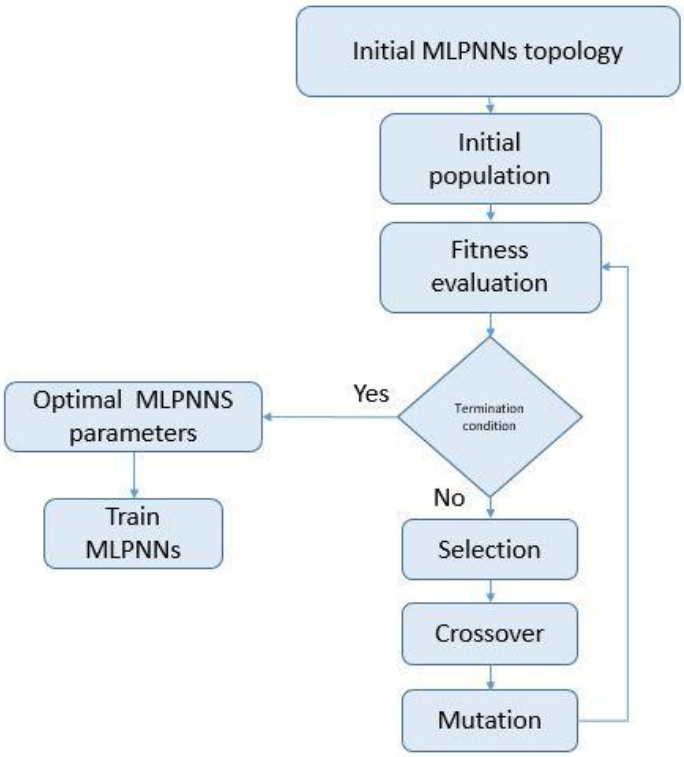

Figure 21 GAs Optimization Based MLPNNs.

4.3 General Method Procedure

GAs-MLPNN is used to predict the potential amount of energy that can be produced from renewable sources using two different approaches. The first approach is based on historical data for sites that exist. The potential renewable energy production of these sites is predicted using observed data. As for the second approach, it depends on the metrological data of a specific geographical area to predict the amount of energy that can be produced in this region. We compare the results of the two approaches, the first is based on real data while the second is based on data computed using the early mentioned energy equations. Then the model classifies the geographical areas according to the electric power productivity from renewable sources using the SOM-Kohonen network. The general steps and phases that were used in performing GAs-MLPNNs on the collected dataset are shown in Algorithm 1.

Algorithm 1: Genetic Algorithm-based optimization MLPNNS

Input: Train dataset, Test Dataset, fitness function, MA function;
Output: Train and test result;

Data Preprocessing Phase:

Step1: Normalize the dataset samples between 0 and 1
for subset **in** dataset **do**
for an item **in** subset **do**

$$\text{subset[item]} \leftarrow \frac{\text{subset[item]} - \min(\text{subset})}{\max(subset) - \min(subset)}$$

Step 2: data filtering
for subset **in** dataset **do**
for an item **in** subset **do**
subset[item] ← MA Function(item)

Genetic Algorithm Training Phase:

Step1: Initialize Parameters
Number of neurons in the hidden layer
Generations
Population size
Crossover fraction
Migration fraction

Step 2: Apply Genetic Algorithm Operations
Initial population
Fitness evaluation
Selection
Crossover
Mutation
Repeat fitness evaluation until converge
Save optimal Parameters

MLPNNs Training Phase:

Get optimal Parameters
Get Train dataset
While MSE less than the threshold
While Termination condition not satisfied
Calculate prediction result
Calculate MSE
Update weights
Number of neurons= Number of neurons+20
Output MSE

Output prediction result
Save network parameters

MLPNNs Testing Phase:

Get network parameters from the training phase
Get Test dataset
Calculate test prediction result
Output MSE
Output test prediction result

4.3.1 Prediction Using GAs-MLPNNs Model Based on Historical Data

The first part of the model is based on historical data of solar energy production from multiple regions in the Palestinian territories. Data were collected from five different regions in the West Bank, including the north, center, and south, to study the impact of geographical diversity in different regions. These areas are Jenin, Tubas, Salfit, Bethlehem, and Hebron. The process of predicting the potential production of these areas includes two methods: The first method in which we used the GAs-MLPNNs model on data for one year, which is the average of all previous readings for each of the mentioned solar energy production sites. The second method we used GAs-MLPNNs model on data for several years, ranging from 3 to 5 years for each site, which is all the data recorded for each site. The first method aims to demonstrate the ability of the model to predict solar energy production over a short period of time that does not exceed a year. It also aims to integrate classical statistical methods to take advantage of the characteristics of all data. While the second method aims to rely on the model without using any statistical methods to prove the efficiency and strength of the model in predicting the long-term that reaches several years.

4.3.2 Prediction Using GAs-MLPNNs Model Based on Metrological Data

In the second part of the model, electrical energy production is calculated based on weather conditions such as solar radiation, wind speed, and temperature using mathematical equations. The outputs are then used to predict the potential electrical energy production using GAs-MLPNNs.

This section has taken two tracks: The first is concerned with forecasting the amount of electric energy that can be generated from solar radiation and maximum temperature in the Palestinian territories. We collected these metrological data for three Palestinian cities: Jenin, Nablus, and Bethlehem. The second track is for the production of electrical energy from wind, which depends on the wind speed. Data was collected for this purpose from four Palestinian cities of Jenin, Nablus, Bethlehem, and Hebron. This part aims to compare the results with the results of the first part, which relied on historical readings of installed sites in different cities of the West Bank.

4.3.3 Classification Using Neural Networks

Ultimately, the model aims to divide and classify geographical regions based on their productivity of electric energy from renewable sources. This model uses the RE productivity values as inputs for the SOM network, which in turn classifies regions so that each group is similar in its ability to produce energy from its renewable sources. To ensure coverage of the largest area of the Palestinian territories, we have used the results of the previous two sections of the model as inputs so that the largest number of cities can be used. SOMNNs use the output of the prediction to produce electrical energy from solar radiation as inputs and transform it into a map showing the ability of each region to produce energy. It also uses wind power forecasting outputs to provide a map showing the potential for each region. What distinguishes this method is that it depends on the amount of potential energy and gives a future vision for the RE sector in Palestine. The steps of applying SOMNNs on the predicted results are in Algorithm 2.

Algorithm 2: Self-Organize Mapping

Input: Set of input patterns
Output: Output patterns

Step1: Initialize Parameters

Number of neurons

Initialize weights to small random values

Step2: Repeat until convergence

Select input pattern x
Find winner neurons w that best match x
Update weights for w and neighbors
Decrease the learning rate
Decrease neighborhood size

4.3.4 Genetic Algorithm - Optimization Based RBFNNs

RBFNNs are feedforward networks consisting of three layers that are input, hidden, and outputs. What distinguishes this network is the use of RBFs in the hidden layer, which is symmetrical function around its center; Gaussian function is usually used. When the central points of the RBF are determined; then the input vector can be directly mapped to the hidden layer. The relationship between the hidden layer and the output layer, it is a linear relationship, where the outputs are calculated through the weighted sum and weights here are the network parameters that must be adjusted. Figure 22 shows the network structure of n inputs, m outputs, and s hidden layer neurons. Where w_{ij} are the weights between input and hidden layer, w_{jk} is weighted between the hidden and output layer.

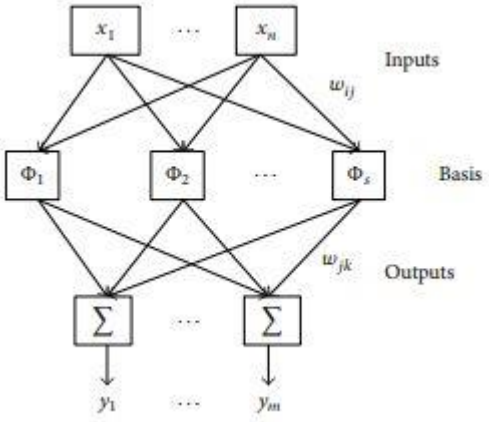

Figure 22 RBFNNs structure [97]

The process of training the network comes in two steps: the first step is to set w_{ij} and the second step involves adjusting w_{jk} of the output layer. The main problem is the process of determining the number of neurons in the hidden layer, as it usually starts from 0 and increases the number to reach the best network structure. The process continues until the required accuracy is reached or the largest number of neurons is reached. While MLPNNS is using a fixed network structure made up of several neurons that does not change during the network training process and thus RBFNNs are superior to MLPNNs. GAs are used to build RBFNNs structure and weights automatically. The GAs-RBFNNs optimization process consists of three basic stages: chromosome representation, applying GAs operations, and then fitness evaluation. The chromosome representation for each network contains the hidden layer neurons, $w_{ij,}$ $w_{jk,}$ and the output layer threshold Θ as follows:

$$c_1\ c_2\ \dots\ c_s\ w_{11}\ w_{21} \dots w_{s1} \dots w_{21}\ w_{22}\ \dots\ w_{s2}\ \dots w_{1m}\ w_{2m}\ \dots\ w_{sm}\ \Theta_1\ \Theta_2 \dots\ \Theta_m. \qquad (50)$$

The phases and steps of GAs-RBFNNs are in Algorithm 3.

Algorithm 3: Genetic Algorithm- based optimization RBFNNs

Input: Train dataset, fitness function, Error goal;
Output: network structure;

Initial RBFNN Phase:

Step1: Initialize Parameters
Max number of neurons in the hidden layer s
Centers distance d

Step2: K-mean clustering
Obtain centers for hidden neurons

Step3: Calculate the width of centers

$$\sigma = \frac{d}{\sqrt{2s}}$$

Genetic Algorithm Training Phase:

Step1: Initialize Parameters
Number of iterations

Generations Max
Population size
Crossover fraction
Mutation fraction
Selection mechanism

Step 2: Apply Genetic Algorithm Operations

Initial population randomly
Repeat
Training network using Train dataset
Calculate output Error

$$E = \sum_{1}^{n} (t - y)^2$$

Fitness evaluation
If E< E goal or G > G max
Select fittest chromosome
Mutation
P= new P
G=G+1
Else
Crossover
Save network parameters

4.3.5 PSO Algorithm-Optimization Based MLPNNs

MLPNN is usually trained using the BP algorithm which relies on a derivative function of the hidden layer neurons. It requires reducing the error over several iterations to reach the best network weights. The weights are adjusted using the same equation for all layers, except that the local error value for each neuron differs. The error is calculated for the output layer through the difference between the actual output and the desired output; the same for the hidden layer. Because the desired output of the hidden layer is unknown, the output layer error is propagated to the hidden layer and the effect on the hidden neurons is calculated using the learning rate. If the value of the learning rate is low, it will slow down the learning process, but if it is large, it will lead to the network not reaching the best weights. Therefore, controlling the learning rate has a great impact on adjusting the weights of the network and reaching the best ones. PSO algorithm is used to optimize MLPNN weights. To train a network using this algorithm, the network solutions are represented by particles

so that the position vector of each particle represents the weights of the network. The error function is used to calculate the fitness value of each particle at its current position and velocity. Algorithm 4 shows the main steps of PSO-MLPNNs Algorithm

Algorithm 4: PSO- based optimization MLPNNs

Input: Train dataset, number of iterations
Output: Optimized network

Step1: Initialize MLPNN

Number of neurons
Initial weights

Step2: Initialize PSO
Initial particle position = Initial weights
Repeat for number of iterations
Train network
Fitness evaluation
Pi= MSE
If Pi < P best
P best= Pi
If P best < g Best
g best= P best
Update particle position
Update particle velocity

4.4 Performance Metrics

There are several methods for measuring the performance of NNs in prediction, which depend on the value of the prediction error. The error value can be measured by comparing the result with the observed or desired values. The high value of the prediction error indicates lower network performance. This section provides definitions and equations for performance measures that were used to calculate prediction error during experiments.

- The Mean Squared Error (MSE) is a measure of how close the results are to the observed values. For every sample, MSE is the difference between the predicted value point to the corresponding value on the observed values, then square the value. Then the summation is

done to obtain the MSE for all samples of the dataset. For a dataset of n samples, the summation should be divided by the number of samples n. The role of squaring is to avoid canceling the negative values by the positive ones.

$$MSE = \frac{1}{n}\sum(\tilde{X_i} - X_i)^2 \qquad (51)$$

Where, n: number of samples, $\tilde{x}_i$ the i^{th} predicted value, and x_i the i^{th} observed value. This measurement tool is used for all experiments to indicate the performance of the GAs-MLPNNs network of prediction.

Summary

This chapter is a detailed explanation of the GAs-MLPNNs hybrid model for predicting and classification the potential renewable energy in Palestine. It contains the whole procedures of the proposed model and the description of the required datasets. In the next chapter, we discuss the experiments and results of using the model.

Chapter Five: Experiments and Results

Introduction

This chapter contains the results of practical experiments for the proposed model GAs-MLPNNs. All the model's procedures were performed through the Matlab environment, Windows 10, Core i3. The chapter consists of five parts that include forecasting and classification experiments. The first section is to examine the proposed model and compare it with Matlab tools and other models; to show the preference of the model performance in the prediction process. The second part is the results of using the proposed model to predict the amount of energy generated by solar energy systems using historical data and data computed from weather factors. The third part includes the results of using the model to predict the amount of wind energy using data computed from weather factors. Part four contains comparisons of the historical and metrological prediction results. The last part includes the results of the classification process for both solar and wind energy.

5.1 Testing and Validation of GAs-MLPNNs Proposed Model

The experiments include applying Matlab tools and other models on data collected from Jenin Governorate for a solar energy production site. This part also includes the proposed model application on the same dataset. The data contained samples for daily readings of four years. This data was used to forecast the amount of potential energy to be produced for the long-term of this location. The results showed that the proposed GAs-MLPNNs model outperformed the other Matlab tools and models on tracking the pattern of the input data.

5.1.1 Results Using Matlab Tools

For testing the performance of the proposed model, we used several tools in Matlab that include ANNs for function approximation. These networks are MLPNNs, RBFNNs, and RNNs. Three different datasets are used; One year, four years, and an average of four years. We initially apply MLPNNs, RNNs, and RBFNNs on data of 2016 in Jenin for training the networks. The dataset of

365 samples was divided into 70% training samples, 15% for validation, and 15% for testing. Then we used a dataset of four years, also divided into 70% training samples, 15% for validation, and 15% for testing. Then we used a dataset of one year that is the average of four years. We used the MSE performance metric for evaluating the training and testing results; and the results were as follows.

- **Testing MLPNN Using *nftool***

To examine the performance of the MLPNNs we used the neural network fitting toolbox of Matlab (*nftool)*. nftool is used for data fitting using a two-layer feed-forward network trained with Levenberg-Marquardt algorithm. Levenberg-Marquardt is a training technique that is used for non-linear real-valued functions in an iterative manner to find a minimum of a function, using the sum of squares of nonlinear functions. We used different levels of network complexity, by increasing the number of neurons in the hidden layer.

Table 6 Result of Appling MLPNNs on 2016 Year

# Neurons	MSE (Train)	MSE (Test)
20	0.0013	0.0016
40	0.0005	0.0005
60	0.0001	0.0003
80	0.0001	0.0004
100	0.00008	0.0003
120	0.00002	0.0004
140	0.00001	0.0004
160	0.00001	0.0029
180	0.000004	0.0083
200	0.000001	0.0051

Table 6 shows the results of applying MLPNNs on data of one year from a site on Jenin in 2016. The results show acceptable error values for the prediction problem. The test error starts to increase with 80 neurons which indicates that the training process is overfitting and a network of 60 neurons

is sufficient for the learning. Figure 23 illustrates the training results of using a different number of neurons.

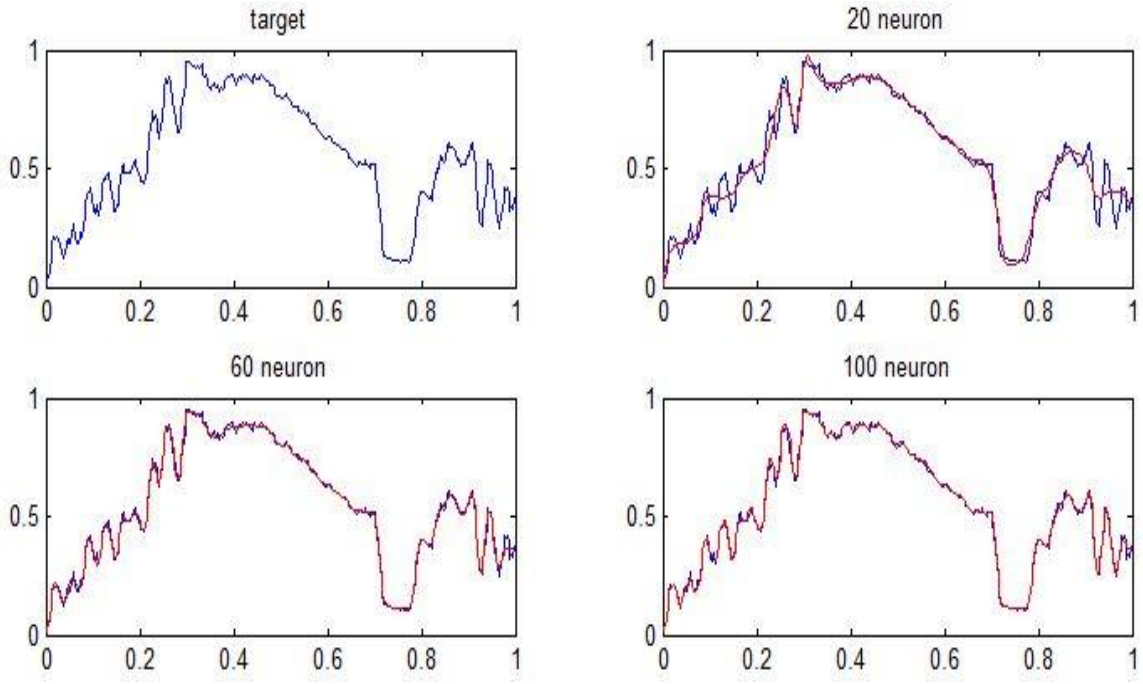

Figure 23 Results of Appling MLPNNs on 2016 Year

Then we apply MLPNNs on four years of the dataset for the same site to examine the performance of the network on a larger set of samples. The result of using a 1460 sample was illustrated in table 7 and figure 24. The table shows that larger train and test errors are recorded comparing with the results of one year. The results are satisfying with fewer numbers of neurons despite the slow improvements in the training process. The difference in the errors is due to the high complexity of the dataset which increases the overall error. The figure shows results for three different numbers of neurons to clarify the enhancement of the network performance as the number of neurons increased. As shown, the network was capable of tracking the target pattern with 40 neurons with test error 0.0041. With 160 neurons, the prediction output exactly fit the target pattern.

Table 7 Result of Appling MLPNNs on Four Years

# Neurons	MSE (Train)	MSE (Test)
20	0.0051	0.0055
40	0.0039	0.0041
60	0.0031	0.0046
80	0.0021	0.0031
100	0.0019	0.0023
120	0.0009	0.0015
140	0.0008	0.0016
160	0.0005	0.0008
180	0.0004	0.0006
200	0.0002	0.0005

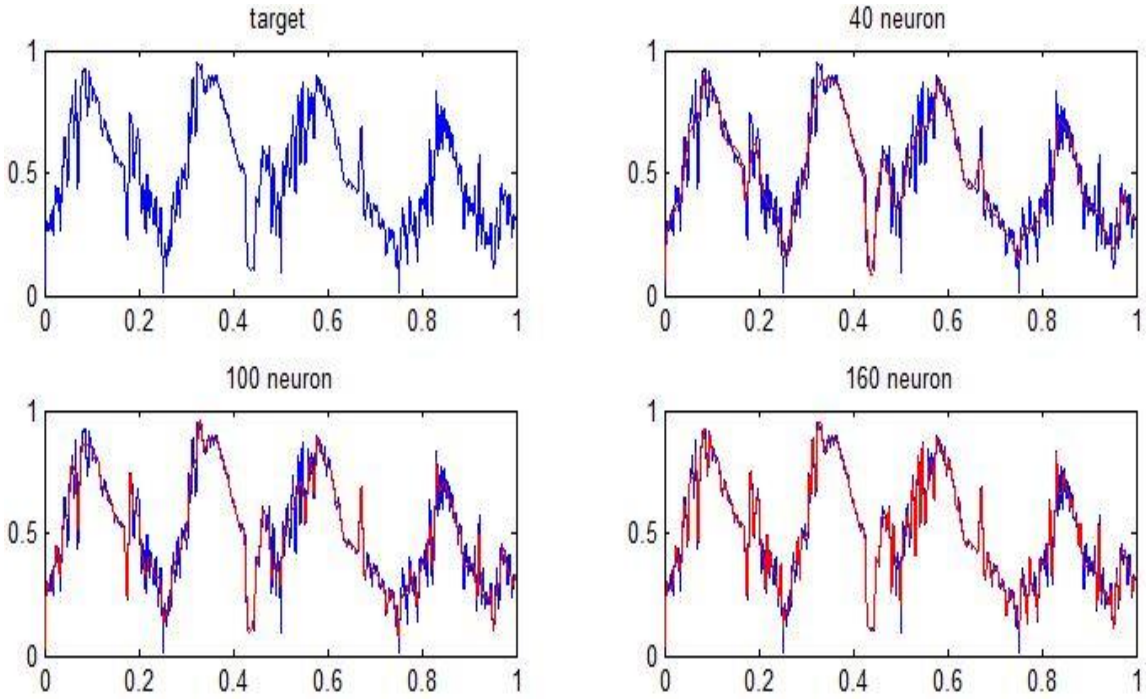

Figure 24 Result of Appling MLPNNs on Four Years

Then we used 365 samples that are the average of the four years to capture the characteristics of all datasets and simplify its complexity. Table 8 and figure 25 represent the results of using different numbers of neurons in the experiment.

Table 8 Result of Appling MLPNNs on Average of Four Years

# Neurons	MSE (Train)	MSE (Test)
20	0.00059	0.00076
40	0.00021	0.00031
60	0.00006	0.00027
80	0.00003	0.00014
100	0.00001	0.00011
120	0.00001	0.00337
140	0.000006	0.00083
160	0.000007	0.00199
180	0.000001	0.00654
200	0.000001	0.00523

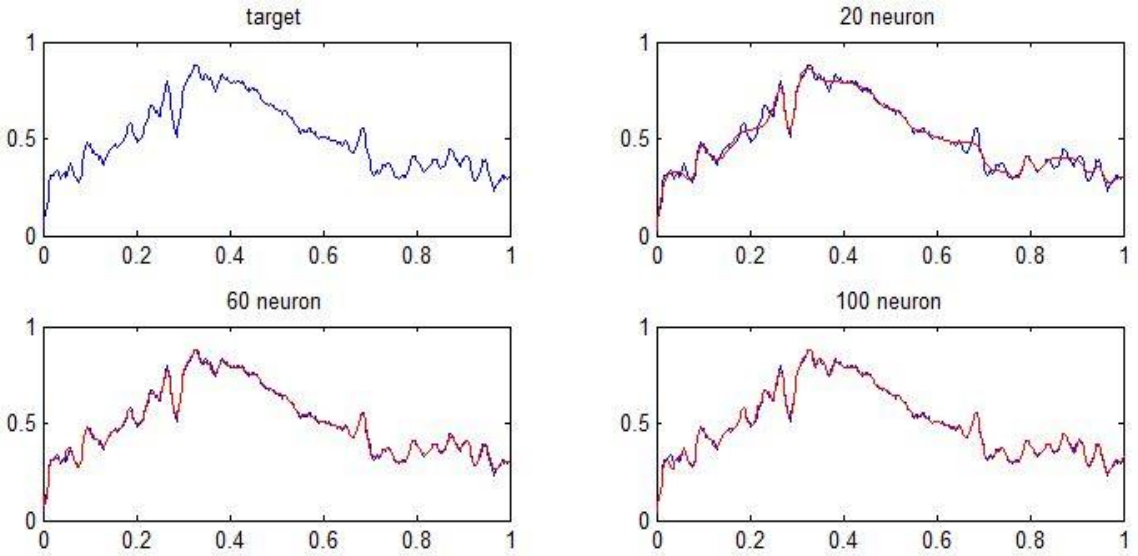

Figure 25 Result of Appling MLPNNs on Average of Four Years

The table shows better improvement in the results than the previous experiment with a rapid decrease in the prediction errors that indicate the high ability of the network to converge. After 100 neurons, the test error starts to increase which means 100 neurons are enough for this dataset. Fewer numbers of neurons can be used for this dataset; as a little difference occurs in the test error between 60, 80, and 100 neurons' networks.

- **Testing RBFNN using *Newrb***

To test the RBFNNs we used the Matlab function *Newrb* with different values of spread constant (δ) and the goal equal to zero. *Newrb* function starts training by 0 neurons and stops when the goal is reached. We train the network using the same sets of data that are used for MLPNNs to compare the performance. Table 9 describes all results of RBFNNs that are represented by applying the network on one-year data of 365 samples. As shown, the increase in neurons does not lead to the improvement of the errors after 50 neurons for different values of spared constant. Figure 26 shows the output for different values of δ. It is evident that, with different values of δ, the network cannot trace the pattern of the dataset; which indicates the complexity of adjusting this parameter.

Table 9 Result of Appling RBFNNs on 2016 Year

# Neurons	MSE(δ=.01)	MSE (δ=.1)	MSE(δ=.5)	MSE(δ=.1)
0	0.062	0.062	0.062	0.062
50	0.001	0.002	0.005	0.008
100	0.00012	0.002	0.005	0.008
150	0.00004	0.002	0.005	0.008
200	0.00003	0.002	0.005	0.008
250	0.00002	0.002	0.005	0.008
300	0.00002	0.002	0.005	0.008
350	0.00001	0.002	0.005	0.008

The results of training RBFNNs using four years were presented in table 10 and figure 27. The network takes a too long time to approximately reach the goal. About 1450 neurons needed to go near the target output which is a huge number. We present a few samples of the results to illustrate the training process.

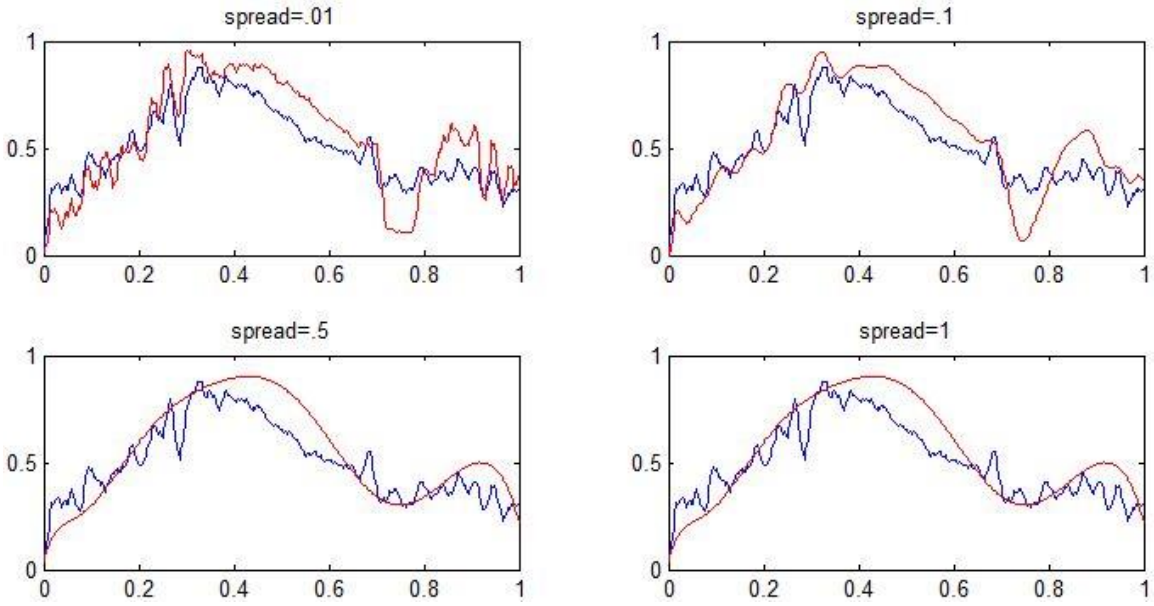

Figure 26 Result of Appling RBFNNs on 2016 Year

Table 10 Result of Appling RBFNNs on Four Years

# Neurons	MSE(δ=.01)	MSE (δ=.1)	MSE(δ=.5)	MSE(δ=.1)
0	0.0463	0.0463	0.0463	0.0463
100	0.0028	0.0073	0.0136	0.0329
300	0.0009	0.0065	0.0136	0.0329
500	0.0008	0.0062	0.0136	0.0329
700	0.0008	0.0062	0.0136	0.0329
900	0.0006	0.0061	0.0136	0.0329
1100	0.0003	0.0062	0.0136	0.0329
1300	0.0003	0.0062	0.0136	0.0329

The results show slow enhancement in the performance of the training when increasing the number of neurons. The figure demonstrates the low performance of RBFNNs on predicting the dataset even for the large number of neurons that are used.

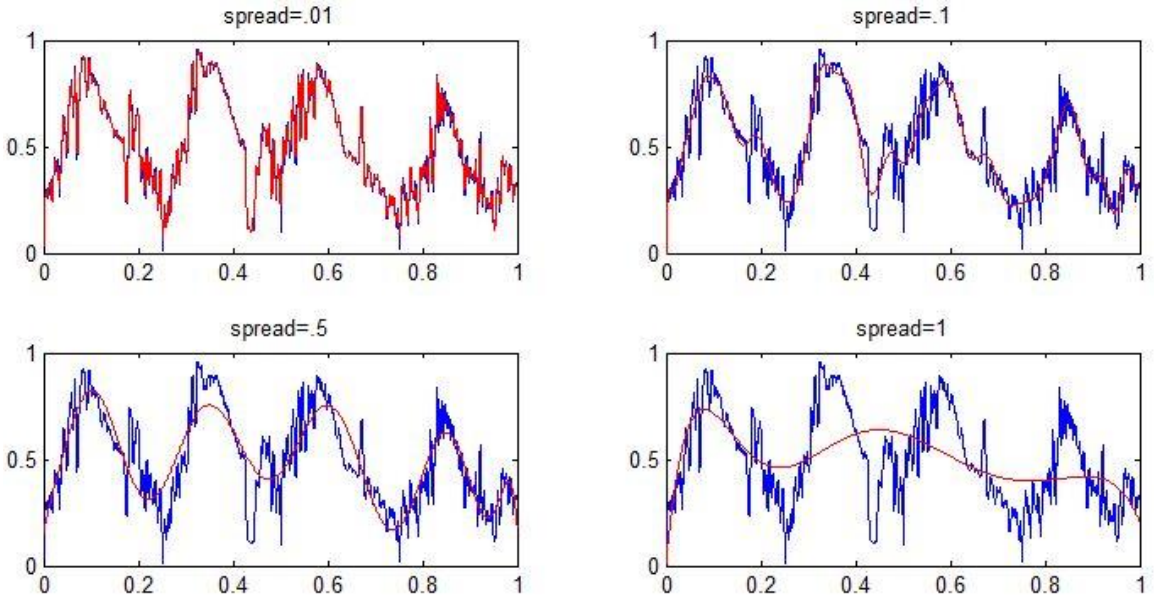

Figure 27 Result of Appling RBFNNs on Four Years

Table 11 and figure 28 show the results of using 365 samples from averaging the data of four years. Generally, results show that tuning the parameters of RBF network is difficult using *newrb* Matlab function. A small value of the spread constant leads to exactly fitting the data which may affect the generalization of the network. And the larger values of the spread constant result in a smooth pattern that lose the detailed characteristics of the data pattern.

Table 11 Result of Appling RBFNNs on Average of Four Years

# Neurons	MSE(δ=.01)	MSE (δ=.1)	MSE(δ=.5)	MSE(δ =1)
0	0.0317	0.0317	0.0317	0.0317
50	0.0008	0.0012	0.0018	0.0022
100	0.00005	0.0012	0.0018	0.0021
150	0.00002	0.0011	0.0018	0.0022
200	0.00001	0.0011	0.0018	0.0022
250	0.00001	0.001	0.0018	0.0022
300	0.00001	0.001	0.0018	0.0022
350	0.00007	0.001	0.0018	0.0022

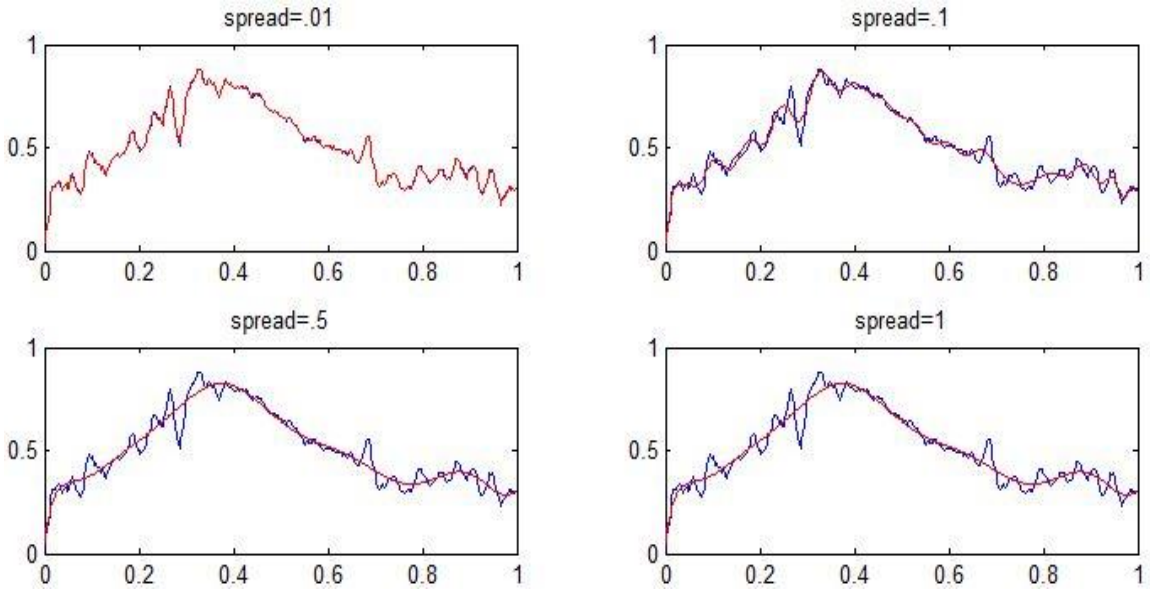

Figure 28 Result of Appling RBFNNs on Average of Four Years

- **Testing RNNs using *nntool***

RNNs are similar to FF networks except that each layer has a recurrent connection with delay associated with it, which allows the network to have an infinite dynamic response to time series input data. For simulating and testing the RNNs we used the neural network toolbox (nntool) of Matlab and trained the network using the same dataset as in previous experiments. nntool use tangent-sigmoid function as a transfer function and the gradient descent as a learning algorithm for data fitting. We used 1 layer delays for the network. The experiment shows that RNNs failed to trace the pattern of the data with different numbers of neurons. Figure 29 shows the training results of applying RNNs on one year of data. And figure 30 shows the results of applying the network on four-year data.

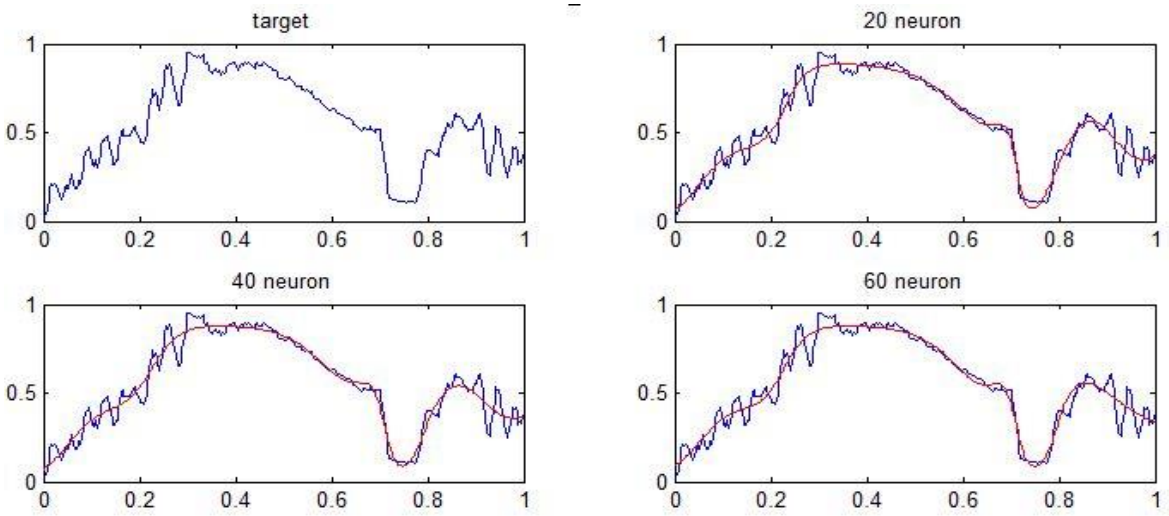

Figure 29 Result of Appling RNNs on 2016 Year

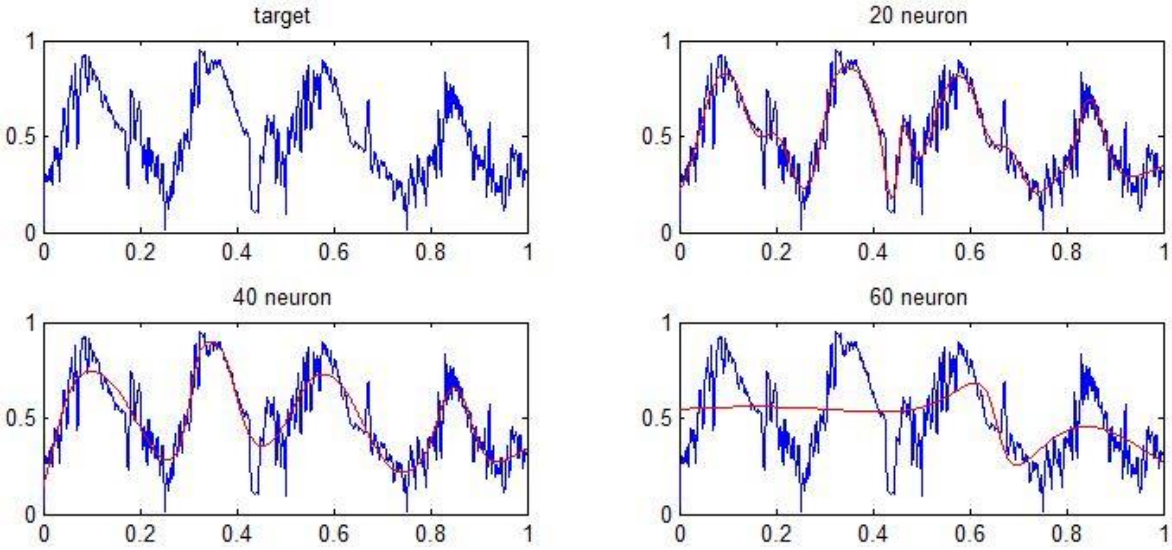

Figure 30 Result of Appling RNNs on Four Years

5.1.2 Results of Testing Hybrid Models

In this part of the experiments, we implemented two hybrid models to examine their performance in predicting solar energy production. The first model is MLPNNs that are optimized using the PSO algorithm (PSO-MLPNNs). The second model is RBFNNs optimized using GAs (GAs-RBFNNs). To obtain the desired results, we apply the models on the data of the Jenin site.

- **Testing PSO-MLPNNs**

We used 365 samples of the Jenin dataset that is an average of four years since this model takes a too long time to process the four-years data and failed on obtaining the results. The results of training the model using 365 samples for different numbers of neurons of MLPNNs are in table 12. The training error of the model is increasing by the increase in the number of network neurons. This means that model is under fitting the dataset.

Table 12 Result of Appling PSO-MLPNNs on Average of Four Years

# Neurons	MSE
20	0.000001
40	0.000293
60	0.000209
80	0.000038
100	0.000941
120	0.000002
140	0.00043
160	0.0347
180	00.0014
200	.0026

Figure 31 also illustrates these results and shows how the model failed of tracking the pattern of the data. The results of 20 neurons were the best but still not sufficient for the training process. Comparing to the MLPNNs results, the PSO algorithm does not improve the performance of the MLPNNs network and the learning process does not optimize the parameters of MLPNNs for this experiment.

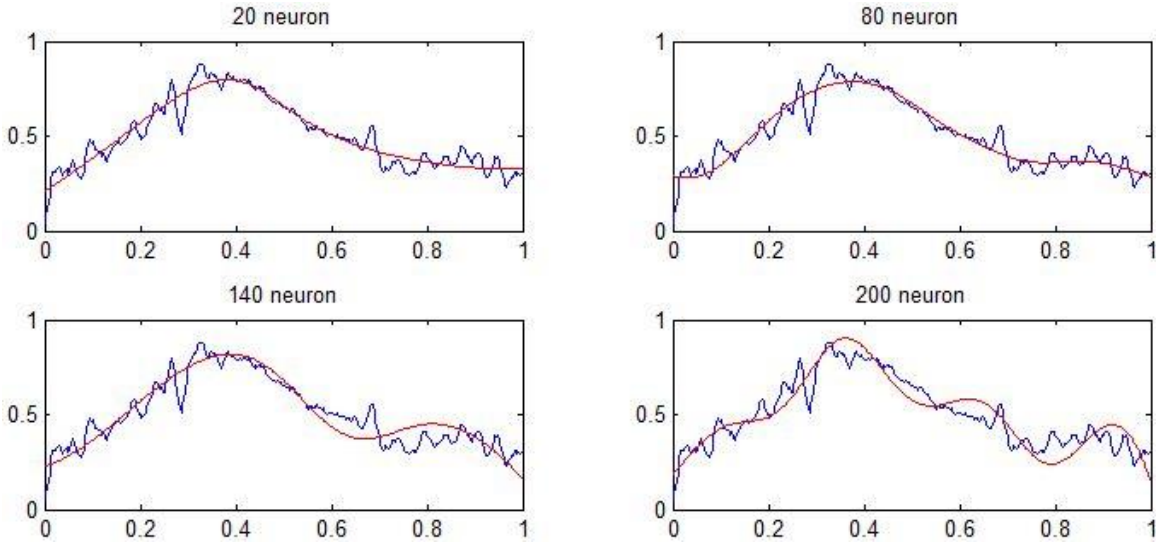

Figure 31 Result of Appling PSO-MLPNNs on Average of Four Years

- **Testing GAs-RBFNNs**

GAs was employed to optimize RBFNNs. For testing the GAs-RBFNNs model we use two datasets. The first is one year of 365 samples for the Jenin solar system. Table 13 shows the results of training and testing RBFNNs using GAs and figure 32 illustrates the results of training the network with 50 neurons.

Table 13 Result of Appling GAs-RBFNNs on 2016 Year

# Neurons	MSE (train)	MSE (test)
10	0.0034	0.0072
20	0.0028	0.0047
40	0.0015	0.0030
50	0.0014	0.0028

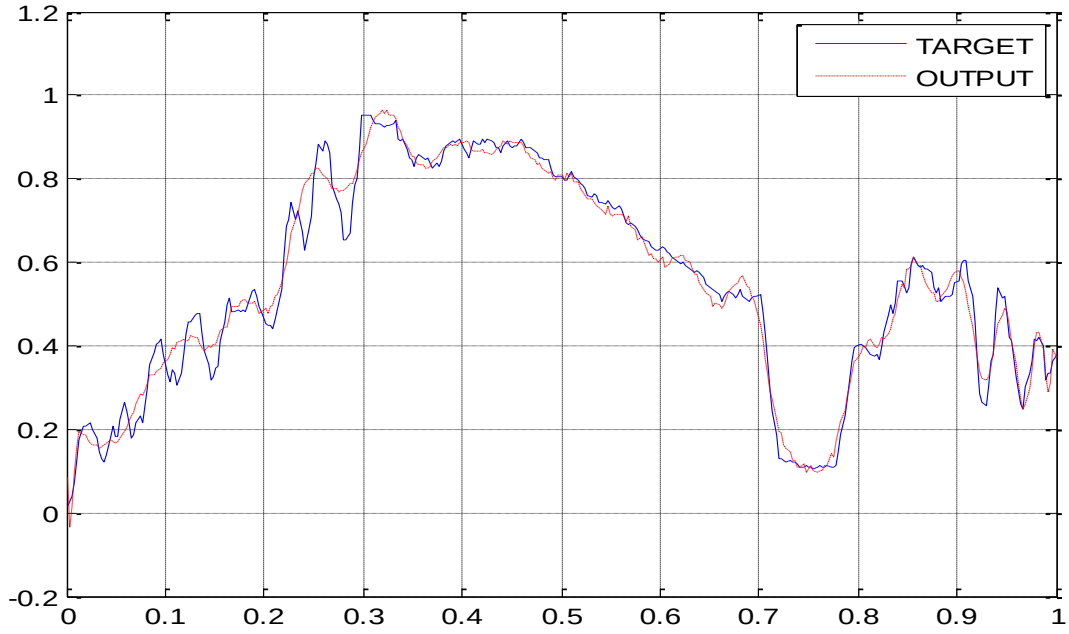

Figure 32 Result of Appling GAs-RBFNNs on 2016 Year

The second dataset is a 1460 sample of historical for the same site. Table 14 contains the training and testing error of applying the GAs-RBFNNs model on the dataset. And figure 33 illustrates the results of the experiment with 80 neurons which represent an overfitting result. The results confirm the superiority of the GAs-RBFNNs model over the RBFNNs model as few numbers of neurons are used, which proves the advantage of the GAs in optimizing the parameters of the RBFNNs.

Table 14 Result of Appling GAs-RBFNNs on Four Years

# Neurons	MSE (train)	MSE (test)
20	0.0077	0.0153
40	0.0059	0.0094
60	0.0060	0.0091
80	0.0073	0.0107

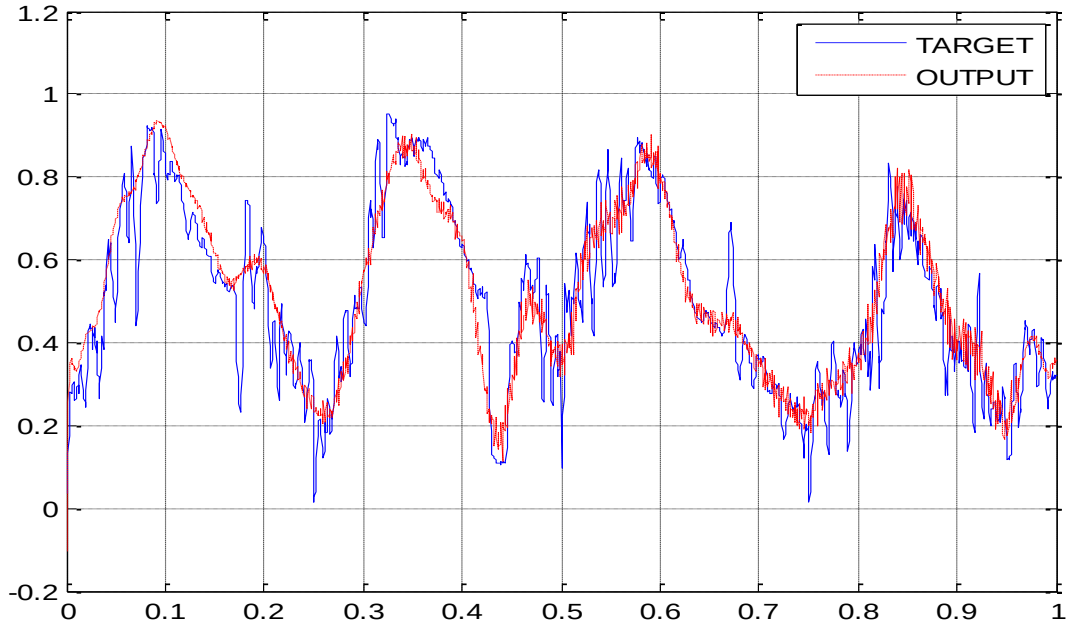

Figure 33 Result of Appling GAs-RBFNNs on Four Years

5.1.3 Results of Testing GAs-MLPNNs Proposed Model

The previous experiments show that MLPNNs outperform the other networks for the prediction of solar energy production. It also demonstrates the superiority of GAs over PSO of optimizing NNs. In this experiment, we examine the combination of MLPNNs and GAs. GAs-MLPNNs model is applied to datasets of 1460 samples. Table 15 consists of the experimental results that are obtained from four years' historical data using a population of 300 individuals and 100 generations for the GAs, and a variable number of neurons for MLPNNs. As we can notice, the enhanced performance of MLPNNs using GAs for network optimization was obtained. GAs-MLPNNs model outperforms the MLPNNs model and reaches satisfying results at a lower number of neurons. With 40 neurons, the MLPNNs model obtains 0.039 and 0.0041 train and test errors respectively; while the GAs-MLPNNs model obtains 0.0025 and 0.0048 train and test errors. On the other hand, the GAs-MLPNNs model outperforms the proposed hybrid models GAs-RBFNNs and PSO-MLPNNs; as

the last failed of tracking the pattern of the dataset while GAs-RBFNNs obtain lower performance than GAs-MLPNNs model. With 80 neurons, the result of training MLPNNs model is overfitting as the test error starts to increase; which indicates that 60 neurons are sufficient to train the network with 0.0017 and 0.0025 train and test errors.

Table 15 Result of Appling GAs-MLPNNs on Four Years

# Neuron	Train error	Test error
20	0.0032	0.0067
40	0.0025	0.0048
60	0.0017	0.0025
80	0.0008	0.0026
100	0.0004	0.0016
120	0.0005	0.0029
140	0.0003	0.0036

Figure 34 represents the training results of applying the model and figure 35 represents the test results to prove the ability of the model to be generalized.

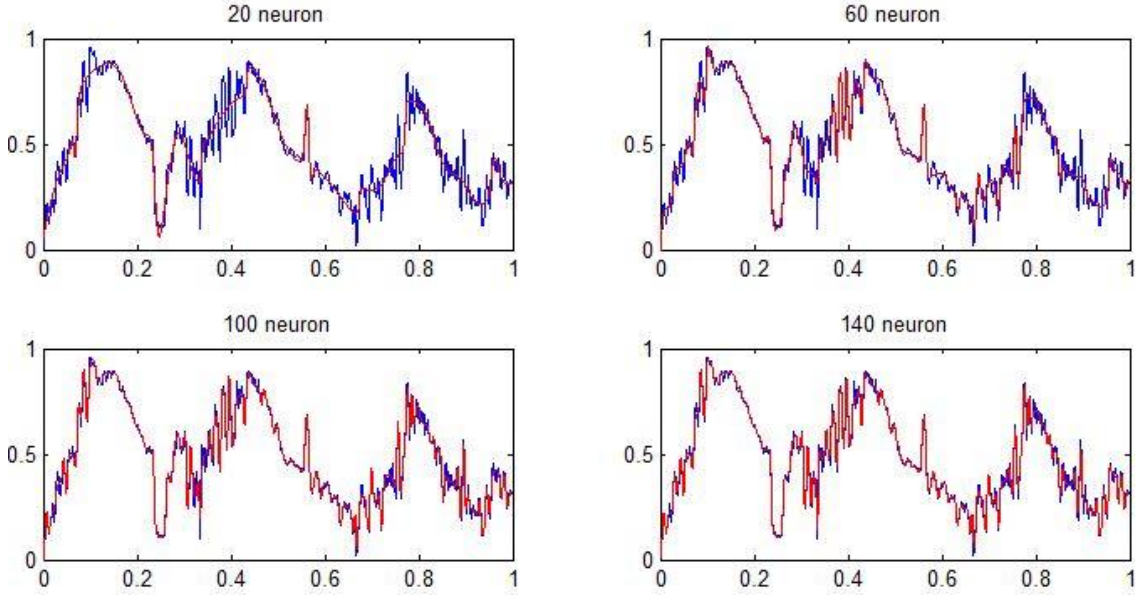

Figure 34 Result of Training GAs-MLPNNs

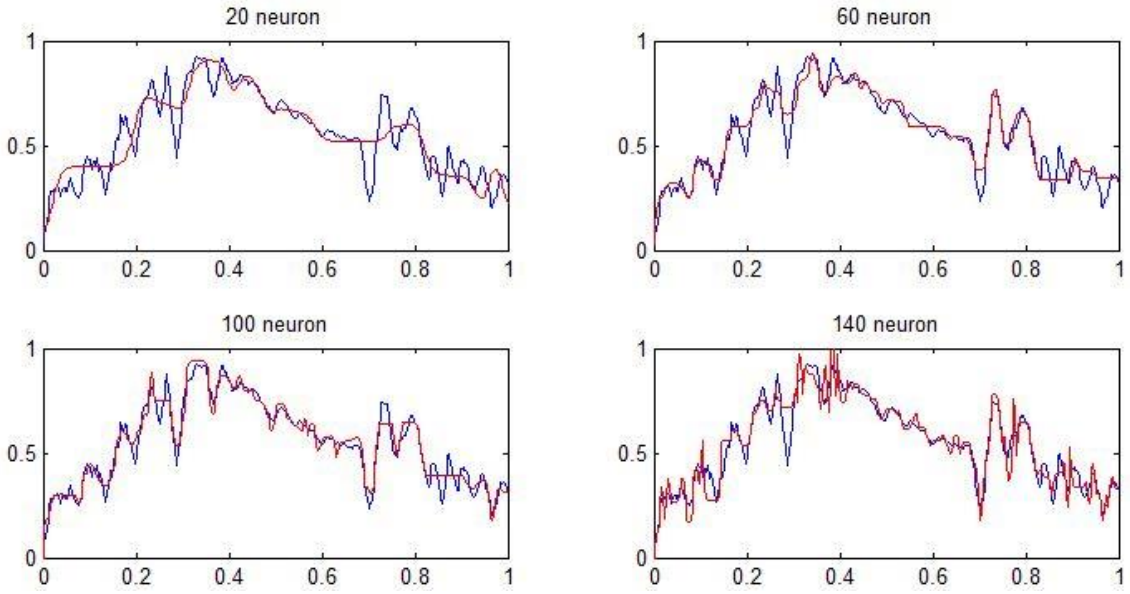

Figure 35 Result of Testing GAs-MLPNNs

5.2 Solar Energy Prediction using GAs-MLPNNs Model

The previous experiments prove the superiority of our proposed model GAs-MLPNNs on learning the behavior of complex data patterns and the ability of the model to generalize. This section contains the experimental results of applying GAs-MLPNNs on datasets from different regions in Palestine territories. The first part includes the results of predicting solar energy production based on historical observations. The second part represents the application of the model on solar energy data that is calculated from metrological data.

5.2.1 Solar Energy Prediction Using GAs-MLPNNs Based on Historical Data

For predicting the potential energy production of an installed PV system; we used previous observations of five different sites in Palestine. The distribution of the sites covers all West Bank regions. All data are preprocessed using the normalization method and moving average filter. We used the k-fold cross-validation method to divide the data for training and testing processes. This reduces the under-fitting as we used all data for fitting, and reduce the overfitting as most of the

data is also being used in the validation set. The length of the fold is 365 samples. We used the early stopping to halt the training of the neural network with suitable number of iterations without causing overfitting or under fitting to the dataset. Early stopping is a method that allows you to specify a large number of training iterations and stop training once the model performance stops improving on validation dataset. We used 100 iterations for training the model.

- **Jenin City**

The first site is a solar rooftop system in Jenin city. We got a dataset of 1460 samples for four years from 2015 to 2018. We apply a 4-fold cross-validation method to guarantee that the GAs-MLPNNs model complex enough to fit the data without causing problems on testing data. 75% of the data were employed for training and 25% for testing the results. Table 16 shows the results of the four experiments using 4-fold cross-validation.

Table 16 Results of GAs-MLPNNs for Jenin Site

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neuron	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0032	0.0067	0.0037	0.0049	0.0039	0.0046	0.0038	0.0039
40	0.0025	0.0048	0.0025	0.0027	0.0025	0.0027	0.0023	0.0024
60	0.0017	0.0025	0.0017	0.0018	0.0019	0.0025	0.0017	0.0031
80	0.0008	0.0026	0.0007	0.0028	0.0008	0.0018	0.0006	0.0027
100	0.0004	0.0016	0.0004	0.0021	0.0005	0.0042	0.0004	0.0025
120	0.0005	0.0029	0.0004	0.0030	0.0005	0.0030	0.0003	0.0039
140	0.0003	0.0036	0.0003	0.0021	0.0003	0.0017	0.0002	0.0014

The results demonstrate that GAs-MLPNNs independent on the dataset since the results are close for all experiments and the average test errors are 0.0035, 0.0028, 0.0029, and 0.0028 respectively. Figure 36 represents the testing result of the third experiment using different numbers of neurons. The data for the year 2018 is used for testing this experiment with an average error of 0.0029. The figure represents how the output pattern is closed to the target with a network of 60 neurons. In

general, the test error starts to increase after 80 neurons, which indicates that less than 80 is suitable for fitting and validating this dataset.

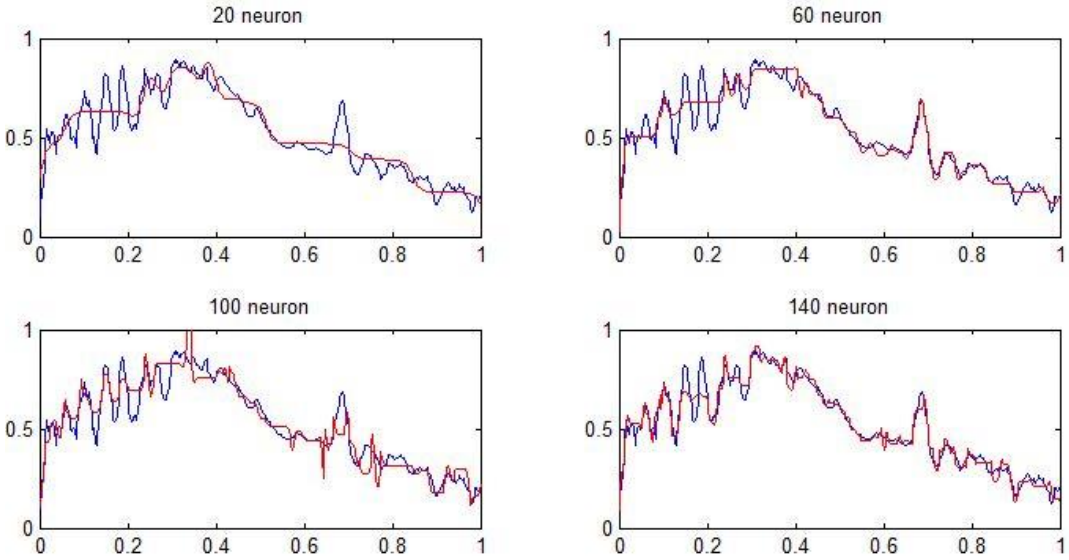

Figure 36 Results of GAs-MLPNNs for Jenin Site

- **Salfit City**

The second site is a PV system in Biddya town, located 16 kilometers from Salfit city. The dataset consists of 1095 samples for three years, from 2017 to 2019. We divided the dataset into training and testing sets by using 3-fold cross-validation of fold length 365 samples. Table 17 contains the results of the three experiments of the GAs-MLPNNs model and the results almost close for all experiments as the average test errors are 0.0028, 0.0031, and 0.0024 respectively. The results of experiment 3 indicate an increase in the test error with 100 neurons, so almost 80 neurons are suitable for this dataset. Figure 37 shows the testing results at 20, 60, 100, and 140 neurons for the third experiment where the data of 2019 is used for the test. The figure represents the overfitting with 100 neurons also represents the closeness of the output pattern with the target using 60 neurons.

Table 17 Results of GAs-MLPNNs for Salfit Site

	Experiment 1		Experiment 2		Experiment 3	
Neuron	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0032	0.0068	0.0033	0.0065	0.0033	0.0044
40	0.0013	0.0030	0.0016	0.0039	0.0011	0.0033
60	0.0007	0.0029	0.0006	0.0027	0.0008	0.0025
80	0.0003	0.0025	0.0005	0.0042	0.0004	0.0015
100	0.0003	0.0019	0.0004	0.0017	0.0003	0.0016
120	0.0003	0.0012	0.0003	0.0015	0.0003	0.0021
140	0.0002	0.0014	0.0003	0.0013	0.0002	0.0011

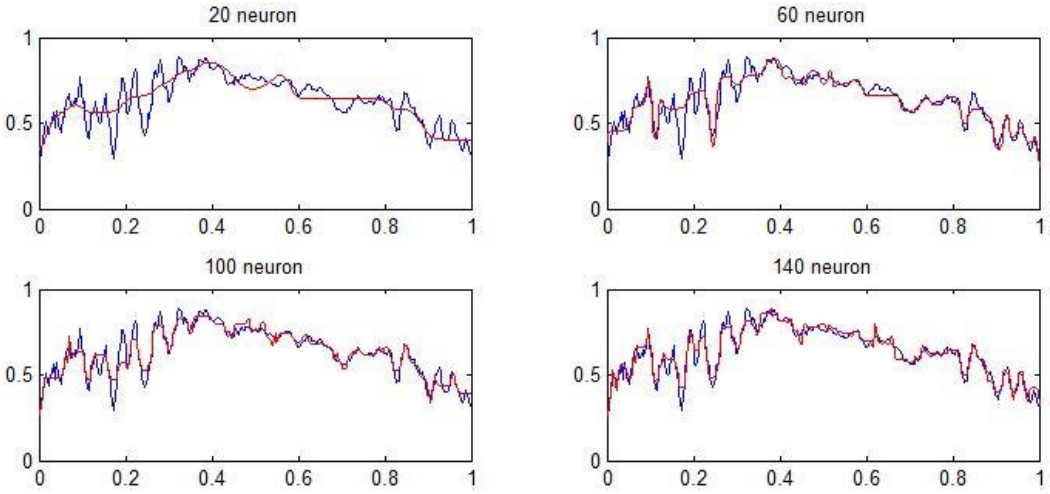

Figure 37 Results of GAs-MLPNNs for Salfit Site

- **Bethlehem City**

The third dataset was collected from Beit Sahour town located 1 kilometer from Bethlehem city. The dataset is four years of 1460 samples divided into training and testing data using 4-folding with fold length of 365 samples. Table 18 displays the results of applying the GAs-MLPNNs model in the four experiments. Results show average test errors of 0.0023, 0.0026, 0.0020, and 0.0021 for the four experiments. Figure 38 illustrates the results of experiment 3 at different numbers of neurons to clarify the prediction process using the GAs-MLPNNs model. With 60

neurons, the network traces the target pattern and obtain satisfactory results even with a few numbers of neurons.

Table 18 Results of GAs-MLPNNs for Bethlehem Site

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0039	0.0053	0.0037	0.0042	0.0038	0.0041	0.0038	0.0036
40	0.0016	0.0030	0.0018	0.0025	0.0026	0.0026	0.0027	0.0022
60	0.0009	0.0019	0.0014	0.0030	0.0013	0.0023	0.0009	0.0013
80	0.0009	0.0014	0.0009	0.0030	0.0006	0.0017	0.0008	0.0021
100	0.0004	0.0011	0.0006	0.0025	0.0005	0.0013	0.0005	0.0016
120	0.0003	0.0021	0.0004	0.0012	0.0003	0.0011	0.0003	0.0018
140	0.0002	0.0016	0.0003	0.0019	0.0003	0.0012	0.0002	0.0019

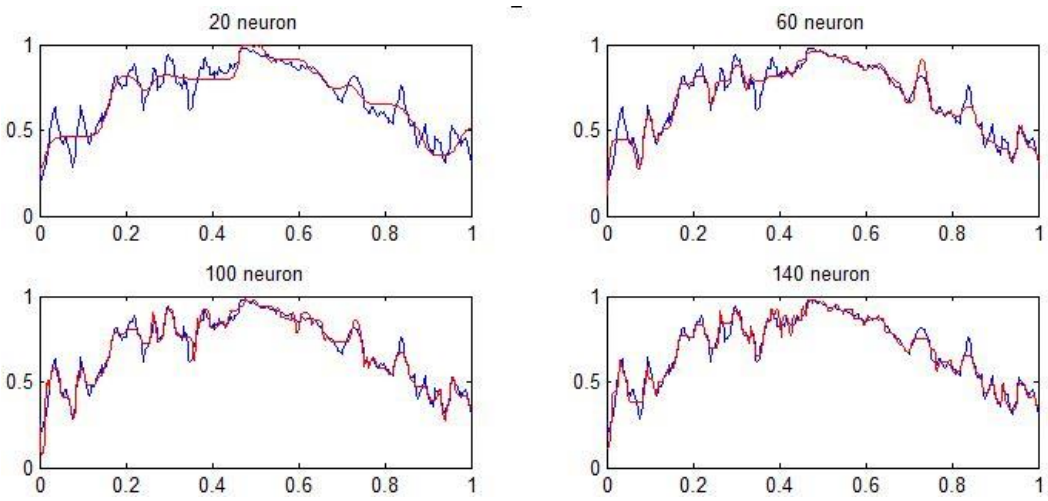

Figure 38 Results of GAs-MLPNNs for Bethlehem Site

- **Hebron City**

This dataset collected from the south of West Bank for three years of 1095 samples from 2017 to 2019; divided into training and testing sets using 3-folding of 365. The results of the three experiments are illustrated in table 19. The average errors obtained from the results of the testing

process are 0.0021, 0.0027, and 0.0033. For the first experiment that has the lowest average error, the test error continues to decrease until 100 neurons.

Table 19 Results of GAs-MLPNNs for Hebron Site

	Experiment 1		Experiment 2		Experiment 3	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0040	0.0047	0.0037	0.0062	0.0038	0.0054
40	0.0020	0.0025	0.0016	0.0032	0.0017	0.0042
60	0.0010	0.0020	0.0009	0.0032	0.0008	0.0036
80	0.0004	0.0018	0.0003	0.0017	0.0003	0.0040
100	0.0004	0.0013	0.0004	0.0017	0.0003	0.0026
120	0.0004	0.0014	0.0003	0.0016	0.0003	0.0015
140	0.0003	0.0012	0.0003	0.0013	0.0003	0.0021

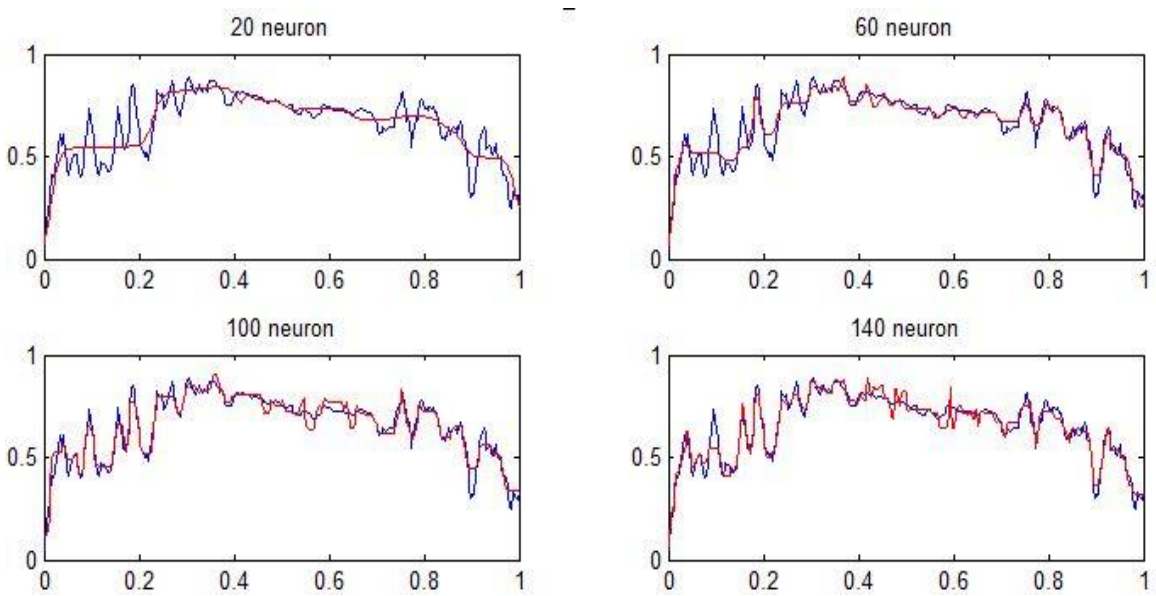

Figure 39 Results of GAs-MLPNNs for Hebron Site

With 120 neurons, the results over fit the target as the error increased. Figure 39 shows acceptable results with a lower number of neurons. The model approximates the target with 60 neurons while 140 exactly fits the target pattern.

- **Tubas City**

Dataset of four years collected from Tammun town, located five kilometers south of Tubas city. The dataset contains 1460 samples from the time period 2016 to 2019. They divided into 75% training data of 1095 samples and 25% testing data of 365 samples using 4-fold cross-validation. Table 20 represents the results of the training and testing of the model. The averages of testing errors for each experiment were 0.0024, 0.0027, 0.0024, and 0.0028, respectively.

Table 20 Results of GAs-MLPNNs for Tammun Site

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0041	0.0034	0.0036	0.0044	0.0036	0.0043	0.0034	0.0039
40	0.0029	0.0021	0.0022	0.0034	0.0026	0.0033	0.0027	0.0036
60	0.0012	0.0026	0.0013	0.0028	0.0012	0.0017	0.0013	0.0023
80	0.0006	0.0018	0.0008	0.0019	0.0008	0.0018	0.0005	0.0019
100	0.0004	0.0032	0.0005	0.0023	0.0004	0.0017	0.0005	0.0025
120	0.0003	0.0024	0.0004	0.0018	0.0004	0.0027	0.0004	0.0023
140	0.0004	0.0017	0.0003	0.0025	0.0003	0.0018	0.0003	0.0032

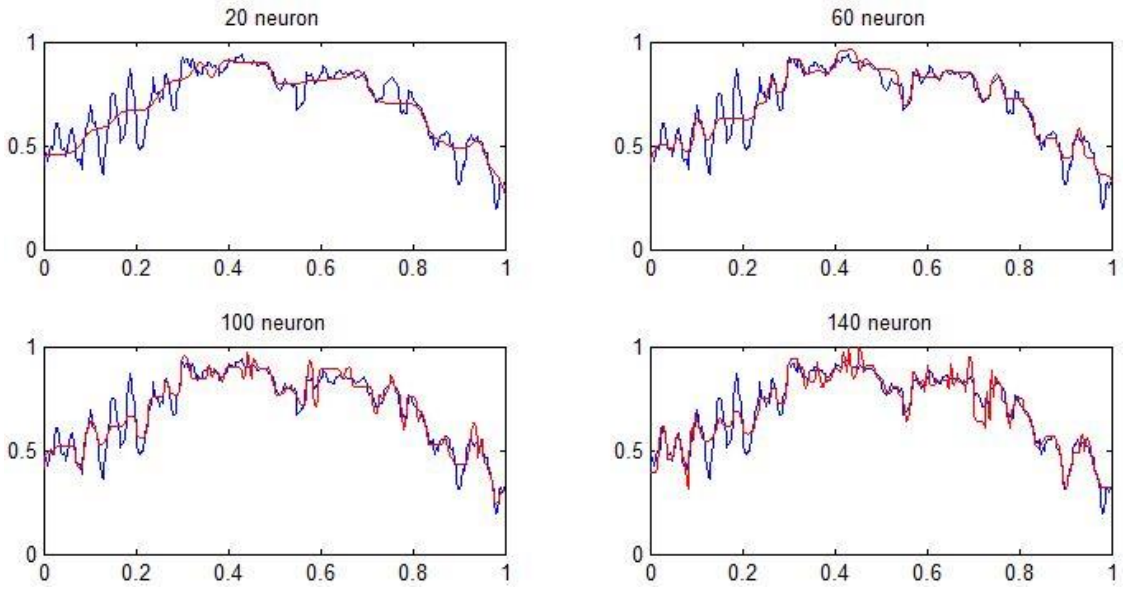

Figure 40 Results of GAs-MLPNNs for Tammun Site

Figure 40 illustrates the testing results of experiment two with different numbers of neurons. The use of 100 neurons and more during the training causes overfitting which is reflected in the testing results. The figure shows how the test results over fit the target at 100 and 140 neurons.

5.2.2 Solar Energy Prediction Using GAs-MLPNNs Based on Metrological Data

In this part of the experiments we used historical metrological data for calculating the potential solar energy production for three cities in West Bank; Jenin, Nablus, and Bethlehem. The data collected by the Palestinian Meteorological Department are the maximum temperature and solar radiation. The maximum temperature is used to calculate the amount of power that can be produced because it represents the temperature during the daylight hours. The average daily solar radiation is obtained by taking the mean of solar radiation for the daylight hours. We used the previously mentioned power generation formula to mathematically obtain solar power production.

$$p = 0.1 \times G \times [1 - (25 - T) \times (-0.0037)] \qquad (2)$$

The results were modified using the normalization formula and MAF. Then we used the calculated power as a target to the GAs-MLPNNs model for power forecasting.

- **Jenin City**

Solar radiation and maximum temperature data were collected for the time period 2016 to 2019. 1460 samples were used for the training and testing processes of the GAs-MLPNNs model. The data divided using 4-folding into test and train datasets for the validation of the model. Table 21 represents the results of the experiments using the proposed GAs-MLPNNs model. The average test errors for the four experiments are 0.0012, 0.0015, 0.0014, and 0.0010 respectively.

The fourth experiment has the least average test error. The test results at different numbers of neurons for this experiment illustrated in figure 41. The figure shows satisfactory results with 60

neurons. With 100 neurons, the GAs-MLPNNs network can reach almost the exact pattern of the target. With 140 neurons, the model over fits the target with a test error of 0.0008.

Table 21 Results of GAs-MLPNNs for Jenin City

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0017	0.0020	0.0016	0.0029	0.0015	0.0030	0.0018	0.0012
40	0.0014	0.0015	0.0014	0.0012	0.0012	0.0018	0.0015	0.0016
60	0.0008	0.0008	0.0006	0.0012	0.0007	0.0011	0.0009	0.0012
80	0.0003	0.0005	0.0004	0.0011	0.0004	0.0008	0.0005	0.0010
100	0.0002	0.0014	0.0003	0.0019	0.0003	0.0014	0.0002	0.0009
120	0.0002	0.0016	0.0002	0.0016	0.0002	0.0014	0.0002	0.0007
140	0.0001	0.0008	0.0002	0.0006	0.0001	0.0009	0.0001	0.0008

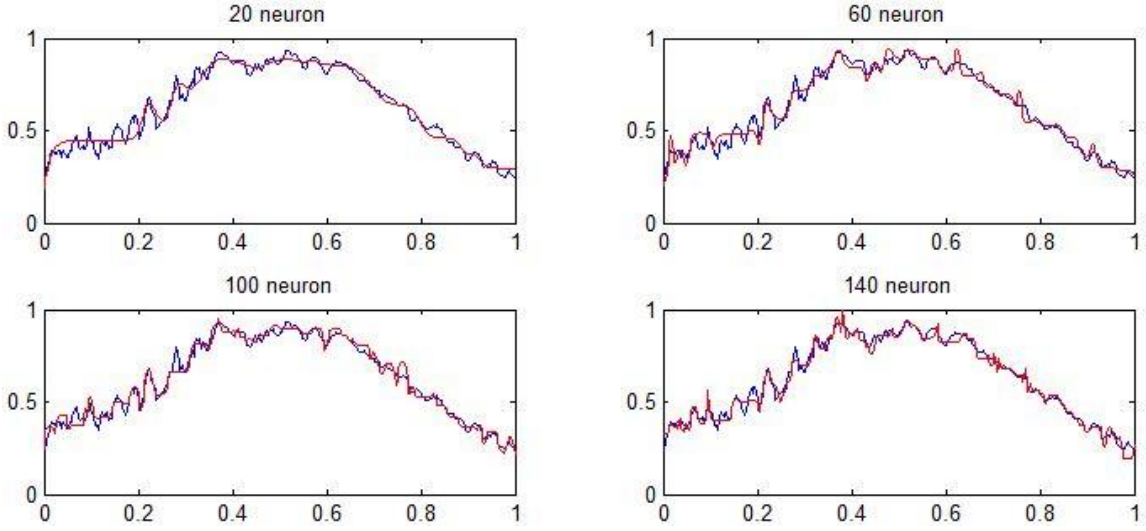

Figure 41 Results of GAs-MLPNNs for Jenin City

- **Nablus City**

The second dataset is for Nablus city. It is a collection of one and a half years for the time period January 2016 to June 2017 of maximum temperature and average solar radiation. The dataset contains 546 samples divided using the Leave-One-Out cross-validation technique by selecting a random part of the data for testing the experiment while the rest of the data remains for the training process. 181 samples were used as a test dataset and 365 samples were used for training. Table 22

represents the results for two experiments with different portions of the test dataset. The average test errors are 0.0008 and 0.0011 for the first and the second experiments respectively. Figure 42 illustrates the results of the first experiment with the lowest average test error at different numbers of neurons. 60 neurons were sufficient to track the data pattern, although the error continued to decrease with the increase in the number of neurons. With 120 neurons, the error increased and the output over fits the target

Table 22 Results of GAs-MLPNNs for Nablus City

	Experiment 1		Experiment 2	
neurons	Train error	Test error	Train error	Test error
20	0.0008	0.0016	0.0008	0.0028
40	0.0001	0.0012	0.00008	0.0011
60	0.0001	0.0010	0.00008	0.0013
80	0.0001	0.0007	0.00005	0.0007
100	0.00007	0.0005	0.00004	0.0007
120	0.00008	0.0005	0.00005	0.0004
140	0.00009	0.0006	0.00006	0.0007

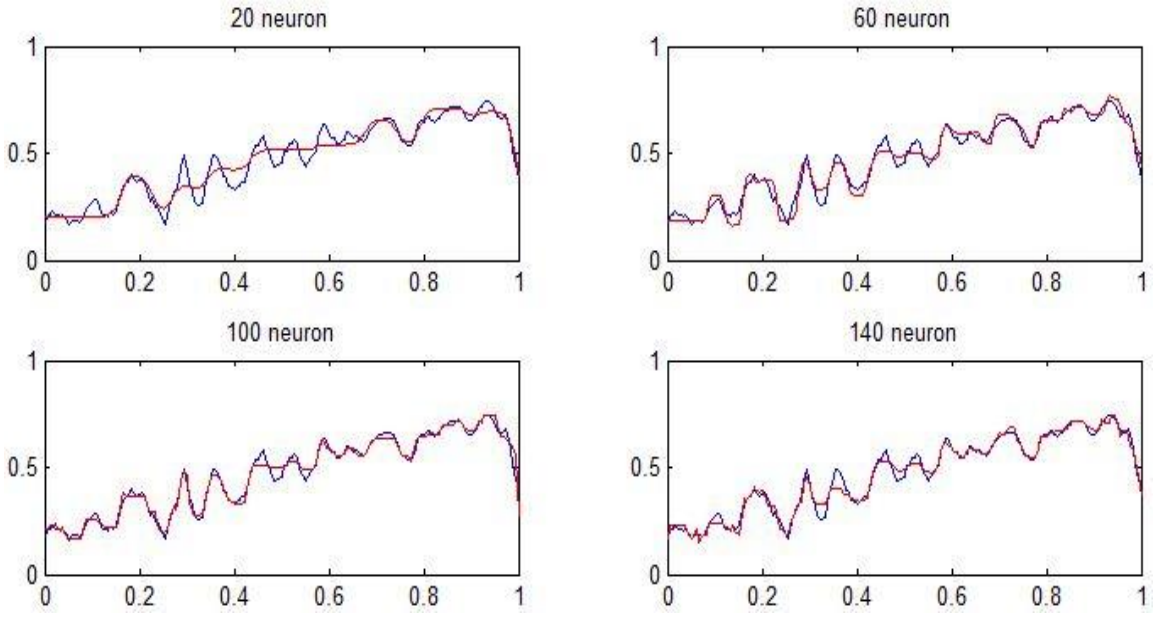

Figure 42 Results of GAs-MLPNNs for Nablus City

- **Bethlehem City**

The dataset used for Bethlehem is 1460 samples for the time period 2016 to 2019 of maximum temperature and average solar radiation. They divided for training and testing datasets using 4-folding. Table 23 represents the results for four experiments of cross-validation using MLPGA NN. The results show average test errors of 0.0022, 0.0011, 0.0012, and 0.0013. Figure 43 illustrates the results for different numbers of neurons for the experiment with the lowest test error, the second experiment. The figure demonstrates satisfactory results at 60 neurons and overfitting results for more than 60 neurons. Less than 60 neurons can approximate the data as the table demonstrates low overall error for all results.

Table 23 Results of GAs-MLPNNs for Bethlehem City

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0013	0.0035	0.0020	0.0019	0.0016	0.0022	0.0016	0.0023
40	0.0007	0.0028	0.0012	0.0012	0.0009	0.0010	0.0009	0.0016
60	0.0004	0.0010	0.0006	0.0007	0.0006	0.0014	0.0007	0.0009
80	0.0003	0.0031	0.0005	0.0011	0.0004	0.0015	0.0003	0.0011
100	0.0001	0.0024	0.0003	0.0005	0.0003	0.0007	0.0003	0.0015
120	0.0001	0.0012	0.0002	0.0015	0.0002	0.0012	0.0001	0.0011
140	0.0001	0.0015	0.0002	0.0008	0.0001	0.0009	0.0001	0.0009

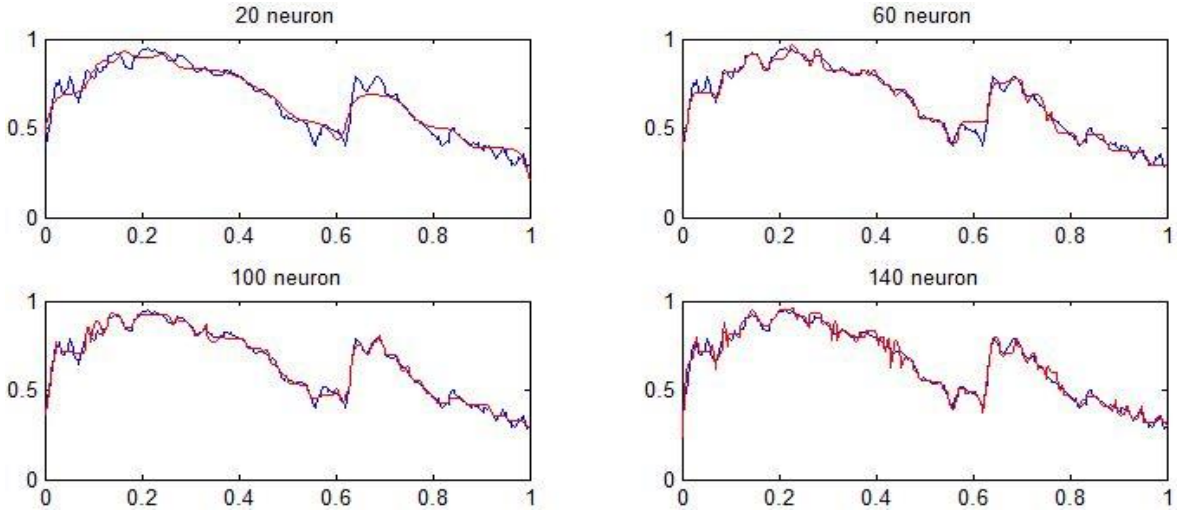

Figure 43 Results of GAs-MLPNNs for Bethlehem City

5.3 Wind Energy Prediction using GAs-MLPNNs Model

Through this section, we present the experimental results of applying the GAs-MLPNNs model for metrological wind data from several cities in the Palestinian Territory. The dataset was collected by the Palestinian Meteorological Department for four cities in West Bank that are Jenin, Nablus, Bethlehem, and Hebron. Historical daily wind speed data used to calculate the wind power production for these cities. The data contains 1460 samples for each city at the time period 2016 to 2019. We noticed significant differences between the power production patterns of each year. Thus, we used the average of the four years for the power prediction to take advantage of the linear and nonlinear characteristics of the patterns. 360 samples that are an average of four years are used for wind power prediction. The dataset divided using 4-folding into a test dataset of 90 samples and a training dataset of 270 samples. 5 samples are ignored. Formula (5) is used to calculate wind power production from wind speed.

$$p = 0.5 \times A \times \rho \times v^3 \qquad (5)$$

Normalized values are used for power forecasting so that the swept area and air density are ignored. The data are filtered by MAF before it is applied for GAs-MLPNNs

- **Jenin City**

Table 24 illustrates the results of applying GAs-MLPNNs on the Jenin dataset for predicting wind power production using wind speed and calculated power. The result shows low errors at the beginning of the training process, due to the use of average wind power production of four years rather than the use of 1460 sample. This makes the learning process faster and reduces the number of neurons. The average test errors for the four experiments are 0.00047 for the first experiment, 0.00042 for the second experiment, and 0.0005 for the third and fourth experiments. Figure 44

illustrates the results for the second experiment at different numbers of neurons. At 40 neurons, GAs-MLPNNs obtain satisfactory results with 0.0006 error, but also at smaller number of neurons, it can trace the pattern of the target data. At 60 neurons the results over fit the target data and the error becomes 0.0009.

Table 24 Results of GAs-MLPNNs for Jenin City Wind Power

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0010	0.0016	0.0010	0.0006	0.0007	0.0011	0.0003	0.0027
40	0.0004	0.0007	0.0005	0.0006	0.0005	0.0006	0.0002	0.0011
60	0.0003	0.0002	0.0003	0.0009	0.0003	0.0005	0.0001	0.0005
80	0.0002	0.0004	0.0002	0.0002	0.0002	0.0005	0.0001	0.0002
100	0.0002	0.0002	0.0002	0.0004	0.0002	0.0004	0.0001	0.0002
120	0.0003	0.0001	0.0003	0.0001	0.0003	0.0002	0.0001	0.0002
140	0.0002	0.0001	0.0003	0.0002	0.0003	0.0002	0.0002	0.0001

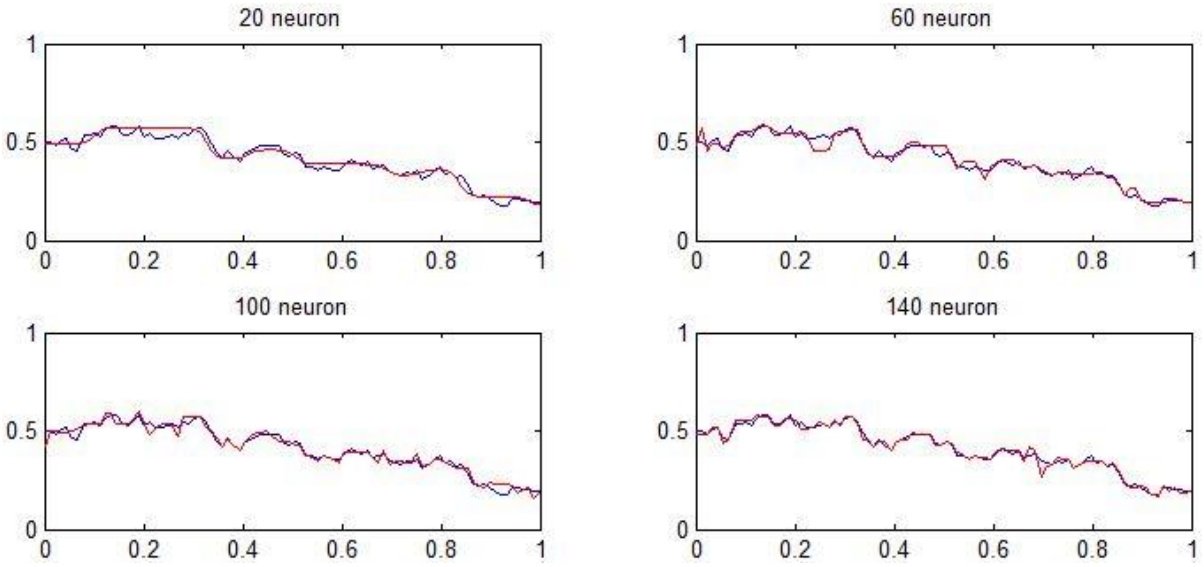

Figure 44 Results of GAs-MLPNNs for Jenin City Wind Power

- **Nablus City**

Table 25 contains the results for applying GAs-MLPNNs on a Nablus dataset of 360 samples that are divided for testing and training using 4-fold cross-validation. The results demonstrate average testing errors of 0.00025, 0.00035, 0.00031, and 0.0004 for the four experiments. Figure 45 illustrates the result of the first experiment which has the least average test error.

Table 25 Results of GAs-MLPNNs for Nablus City Wind Power

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0004	0.0003	0.0005	0.0005	0.0004	0.0004	0.0002	0.0012
40	0.0002	0.0002	0.0002	0.0011	0.0001	0.0002	0.0001	0.0006
60	0.0001	0.0004	0.0001	0.0002	0.0001	0.0004	0.00008	0.0004
80	0.0001	0.0004	0.0001	0.0002	0.0001	0.0002	0.00007	0.0001
100	0.0001	0.0002	0.0001	0.0003	0.0001	0.0004	0.00008	0.0002
120	0.0001	0.0001	0.0001	0.0001	0.0001	0.0002	0.00009	0.0002
140	0.0002	0.0002	0.0002	0.0001	0.0001	0.0004	0.00015	0.0001

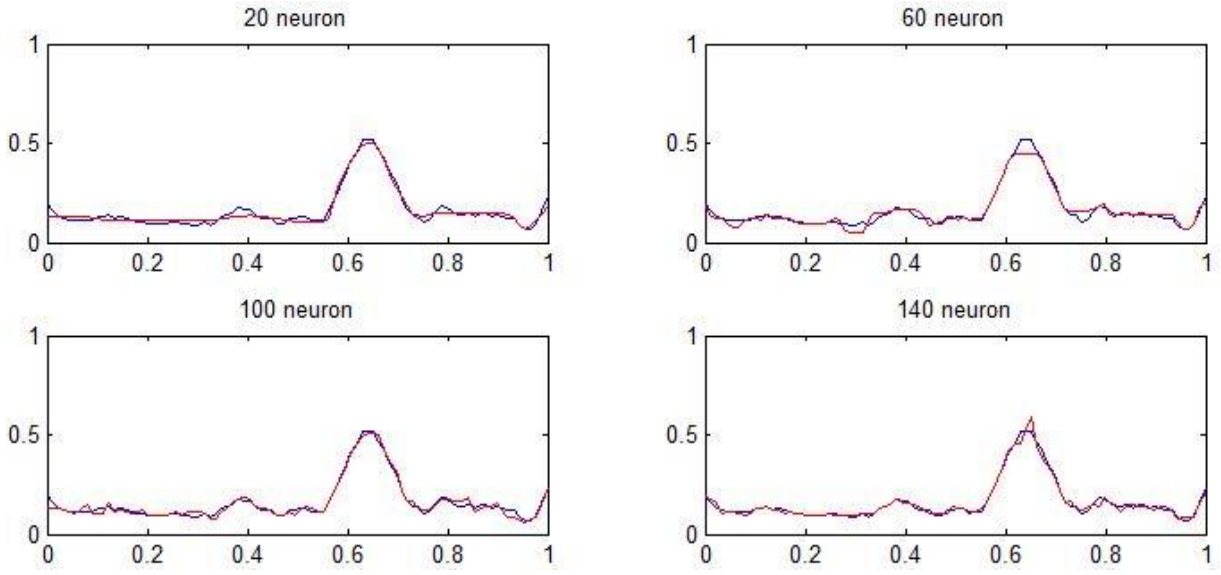

Figure 45 Results of GAs-MLPNNs for Nablus City Wind Power

The results show that a few numbers of neurons are enough for fitting the data pattern. With 60 neurons the test error increased and the results overfit the target.

- **Bethlehem City**

Table 26 contains the results of applying GAs-MLPNNs to predict the wind power production in Nablus city using the calculated wind power from metrological data. The results demonstrate the high performance of the model starting from a few numbers of neurons, 20 neurons. At 40 neurons, the error increased. The average test error for each experiment is 0.00057, 0.00031, 0.00034, and 0.00061. Figure 46 presents the results of the second experiment with the lowest average test error. The figure shows that the model traces the target pattern of 20 neurons. With 60 neurons, the prediction results exactly match the target which causes overfitting

Table 26 Results of GAs-MLPNNs for Bethlehem City Wind Power

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0003	0.0011	0.0006	0.0003	0.0004	0.0006	0.0003	0.0018
40	0.0001	0.0010	0.0002	0.0008	0.0002	0.0009	0.0002	0.0013
60	0.0001	0.0005	0.0001	0.0003	0.0001	0.0004	0.0001	0.0003
80	0.0001	0.0003	0.0001	0.0002	0.00009	0.0002	0.00001	0.0004
100	0.0001	0.0006	0.00008	0.0003	0.00007	0.0001	0.00007	0.0001
120	0.0001	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001	0.0002
140	0.0002	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	0.0002

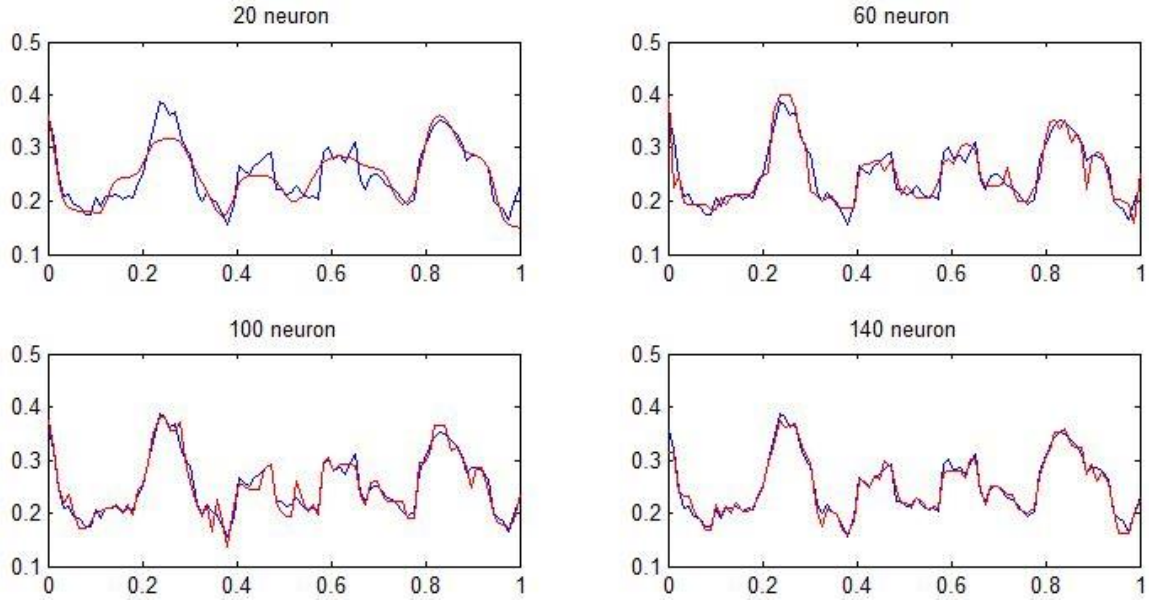

Figure 46 Results of GAs-MLPNNs for Bethlehem City Wind Power

- **Hebron City**

Table 27 presents the results of applying GAs-MLPNNs on the Nablus dataset for wind power forecasting using calculated results. We obtained the average test error from the results for each experiment and it was as follows: 0.00025, 0.00014, 0.00032, and 0.00038. The lowest average error is for the second experiment. The experiment result was satisfying from the beginning of the learning process as the errors start with low values at 20 neurons; despite the late increase in the errors with 60 neurons. Figure 47 shows the test results of the second experiment. With 60 neurons the test error increased and the figure shows that the results are overfitting the target.

Generally, the results demonstrate a small test error for all experiments of each city; which prove the ability of GAs-MLPNNs to predict wind power production with few numbers of neurons.

Table 27 Results of GAs-MLPNNs for Hebron City Wind Power

	Experiment 1		Experiment 2		Experiment 3		Experiment 4	
Neurons	Train error	Test error	Train error	Test error	Train error	Test error	Train error	Test error
20	0.0003	0.0005	0.0003	0.0001	0.0002	0.0005	0.0002	0.0013
40	0.0001	0.0003	0.0001	0.0001	0.00009	0.0006	0.0001	0.0002
60	0.00007	0.0003	0.00009	0.0002	0.00006	0.0003	0.00008	0.0003
80	0.00005	0.0003	0.00007	0.0002	0.00005	0.0003	0.00004	0.0003
100	0.00005	0.0002	0.00006	0.00009	0.00006	0.0002	0.00004	0.0001
120	0.00005	0.0001	0.00008	0.00012	0.00009	0.0002	0.00004	0.0002
140	0.00009	0.0001	0.00013	0.00021	0.00015	0.00017	0.00009	0.0003

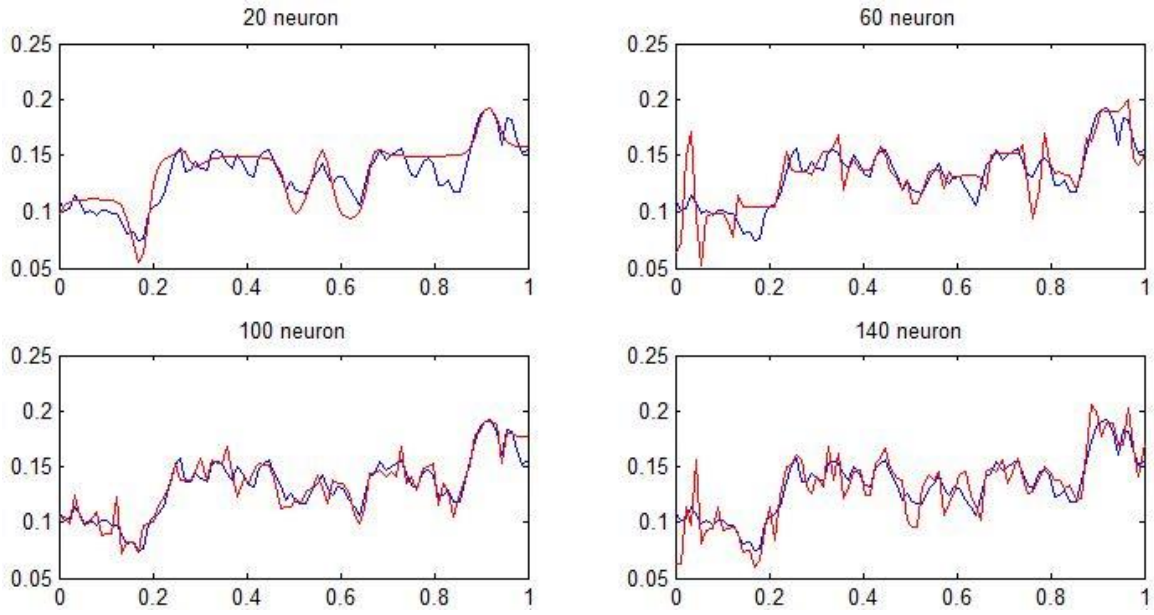

Figure 47 Results of GAs-MLPNNs for Hebron City Wind Power

5.4 Comparison of Prediction Results

To compare the performance of GAs-MLPNNs using historical data with the results of the model using metrological data we use the average errors and the output for each approach. The average errors represent the average values of the test and train errors for all experiments at every level of network complexity. The output is the test output for one experiment at a specific number of neurons, for each city. We make the comparisons between the historical based results and metrological based results for three cities; Jenin, Nablus, and Bethlehem. For Jenin and Bethlehem, we use the results of the four cross-validation experiments to calculate the average errors. For Nablus city, we compare the metrological based results of Nablus with the historical based results of Salfit due to the lack of datasets, the closeness of the two cities, and the similarity of the weather conditions.

- **Jenin City**

Table 28 contains the average errors for both approaches, historical-based, and metrological based on Jenin datasets. The results indicate better prediction performance for the metrological approach than the calculated approach. For example, less than half of the historical approach errors are obtained by the metrological approach with 20 neurons. Figure 48 shows the actual, calculated, and predicted outputs of Jenin city using datasets of the year 2018 with 60 neurons. The figure demonstrates the superiority of the metrological approach to approximating the pattern of power production.

Table 28 Average Errors for Jenin City

	Historical		Metrological	
Neurons	Average Train error	Average Test error	Average Train error	Average Test error
20	0.0037	0.0051	0.0017	0.0023
40	0.0025	0.0031	0.0014	0.0015
60	0.0017	0.0025	0.0007	0.0010
80	0.0007	0.0025	0.0004	0.0009
100	0.0004	0.0026	0.0002	0.0014
120	0.0004	0.0032	0.0002	0.0013
140	0.0003	0.0022	0.0001	0.0008

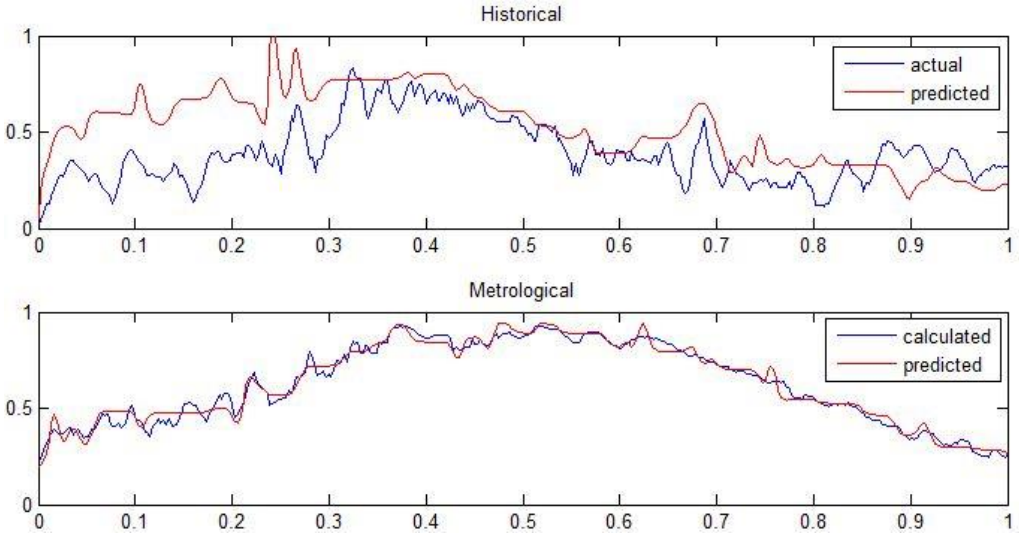

Figure 48 Comparisons for Jenin City

- **Bethlehem City**

Table 29 present the average results of the historical based and metrological based datasets of Bethlehem. The table shows that less than half of the error value can be obtained by the metrological approach with the same number of neurons. With 20 neurons, the training error is 0.0038 using the historical data while it is 0.0016 using metrological data. With 60 neurons, exactly half of the error can be obtained. Figure 49 shows the actual, calculated, and predicted outputs of Bethlehem city using datasets of the year 2019 with 60 neurons.

Table 29 Average Errors for Bethlehem City

	Historical		Metrological	
Neurons	Average Train error	Average Test error	Average Train error	Average Test error
20	0.0038	0.0043	0.0016	0.0025
40	0.0021	0.0026	0.0009	0.0016
60	0.0012	0.0021	0.0006	0.0010
80	0.0008	0.0020	0.0004	0.0017
100	0.0005	0.0016	0.0003	0.0013
120	0.0003	0.0016	0.0002	0.0013
140	0.0003	0.0016	0.0001	0.0011

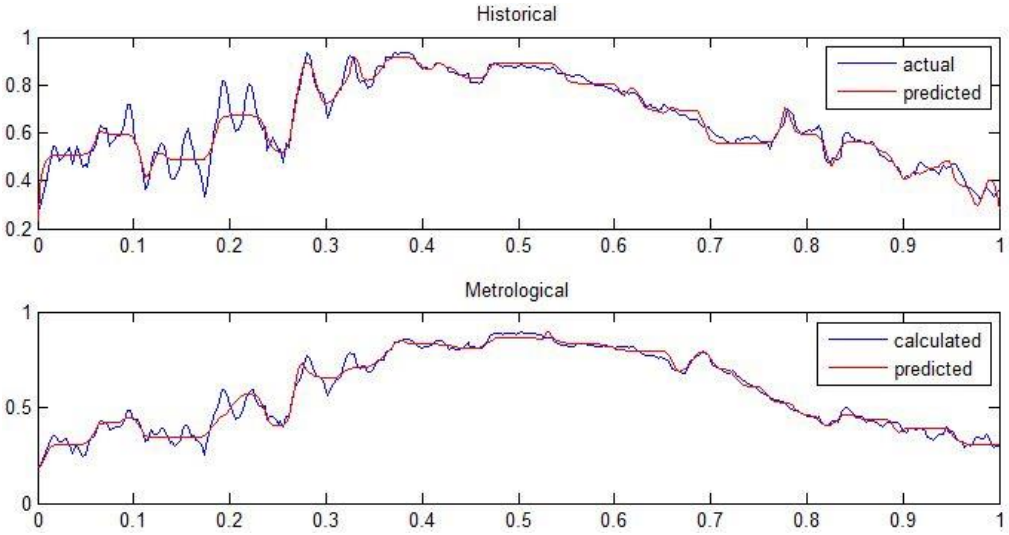

Figure 49 Comparisons for Bethlehem City

- **Nablus City**

Table 30 represents the average test and train errors of the historical based and metrological based datasets of Nablus city. The average errors show that GAs-MLPNNs obtain convergent performance using metrological data and historical data at the same number of neurons; which is illustrated in figure 50 that represents show the actual, calculated, and predicted outputs of Nablus city using datasets of the year 2016 with 60 neurons.

Table 30 Average Errors for Nablus City

	Historical		Metrological	
Neurons	Average Train error	Average Test error	Average Train error	Average Test error
20	0.0011	0.0051	0.0008	0.0022
40	0.0003	0.0025	0.0001	0.0012
60	0.0002	0.0012	0.0001	0.0011
80	0.0002	0.0015	0.00008	0.0007
100	0.0002	0.0010	0.00006	0.0006
120	0.0001	0.0011	0.00006	0.0005
140	0.0002	0.0007	0.00007	0.0006

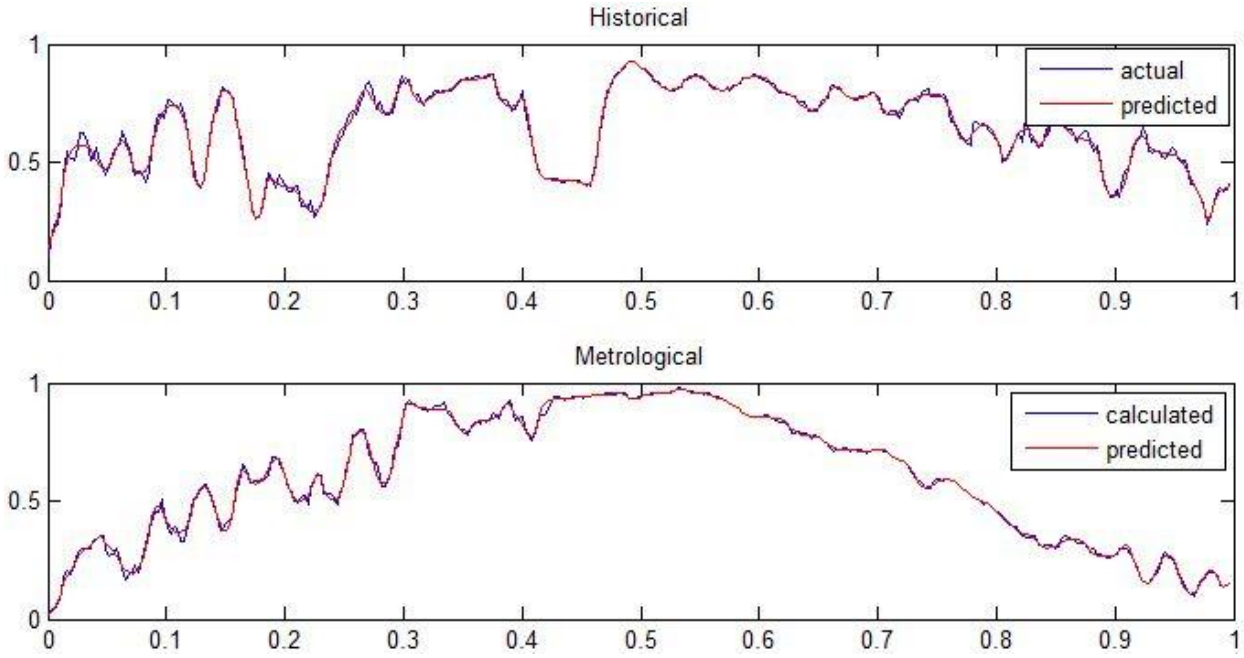

Figure 50 Comparisons for Jenin City

5.5 Classification Based on Renewable Energy Production

This section includes the geographical area classification based on the potential power production of solar and wind energy. We employ the test results of the previous experiments with 60 neurons to obtain annual solar power production by averaging the prediction outputs. The results from metrological data for Jenin, Nablus, and Bethlehem are used because the model proves higher accuracy than the historical based results. For Tubas, Salfit, and Hebron, we used historical based results due to the lack of metrological data. SOMNNs on the Matlab network clustering application was used to classify the cities of Palestine territories based on their solar power production. SOM network has one layer of neurons connected in a grid form that is moving during the training phase to become the centers of the input clusters. We used a six-dimensional input space that represents the data of the six cities. SOMNNs of 36 neurons are used for the classification. Figure 51 illustrates the results of the sample hits tool which indicates the number of clusters obtained using SOMNNs for solar power classification. We can see four clusters using the SOM network topology of 36 neurons.

Figure 52 shows the classification of the cities depending on their potential solar power production. The weight planes tool is used to distinguish the four clusters. The result shows that Jenin, Bethlehem, and Salfit have almost similar weights from input to neurons layer and they belong to the same cluster while the other cities have different patterns that indicate different clusters. The black connections are negative, the red is zero, and the strongest connections are yellow.

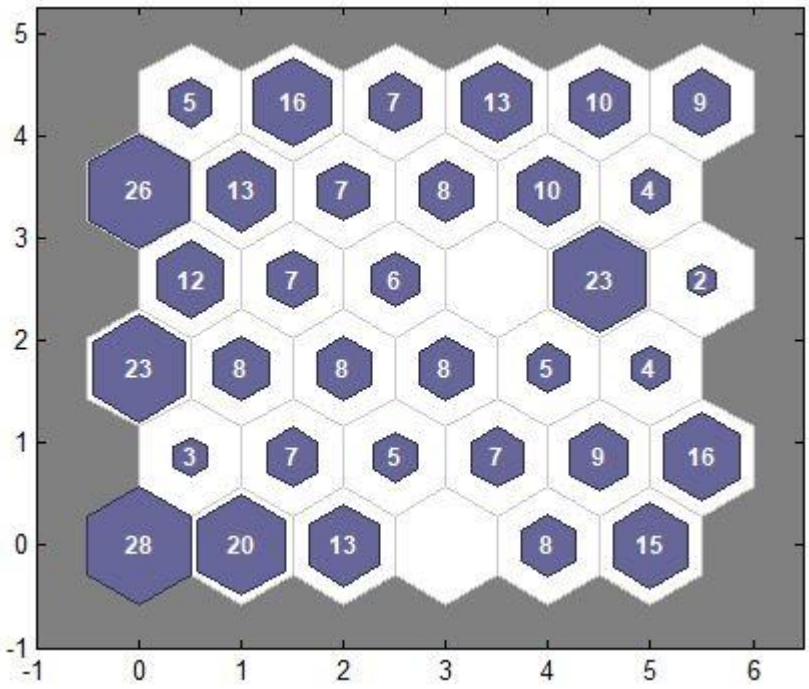

Figure 51 Sample Hits for Solar Classification

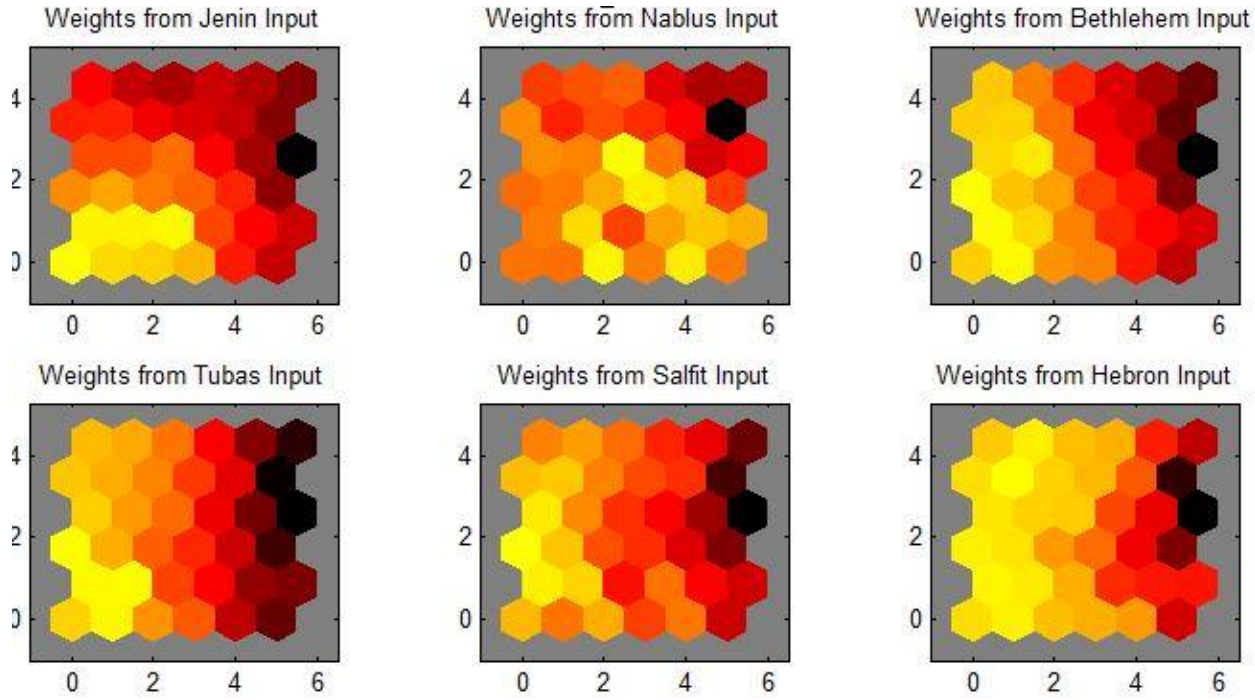

Figure 52 Weight Planes for Classification Using Solar Power

For the classification of the cities depending on their potential wind power, we employ the annual wind power production based on forecasting results of the GAs-MLPNNs model. Four-dimensional input space is used to represent the data of the four cities, Jenin, Nablus, Bethlehem, and Hebron. SOMNNs of 36 neurons are used to classify these cities depending on wind power production. Figure 53 illustrates the sample hits that include the number of clusters and the winner neurons. The figure shows four winner neurons which means each city has its pattern of wind power production. The weight planes of the input to the neurons layer also show four diverse planes which are illustrated in figure 54.

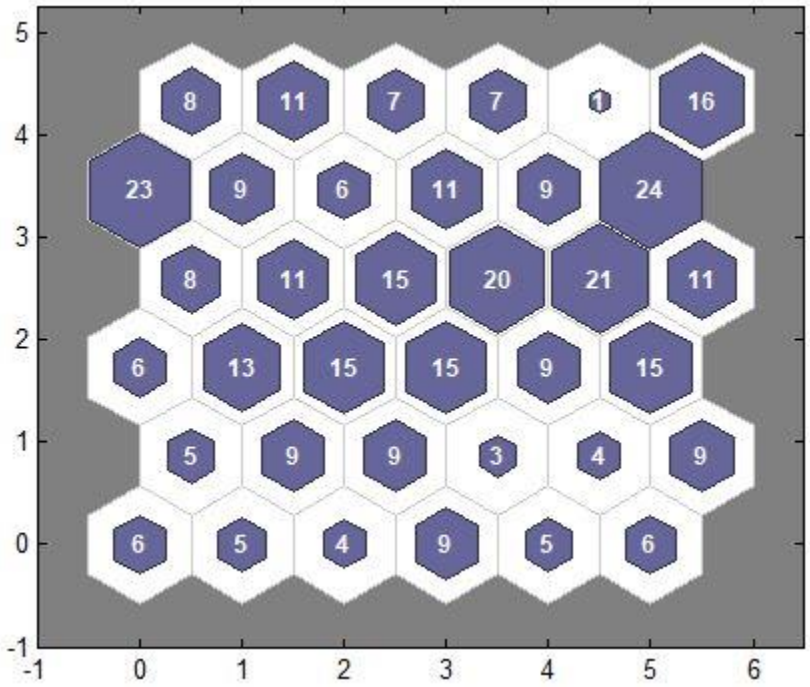

Figure 53 Sample Hits for Wind Classification

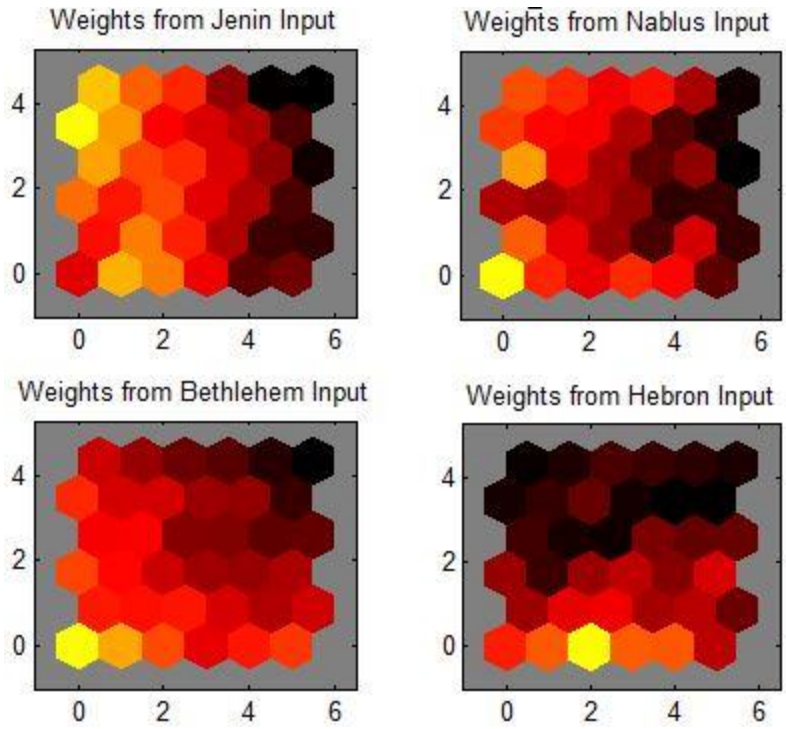

Figure 54 Weight Planes for Classification Using Wind Power

5.6 Challenges and Limitation

In this thesis, we aim to predict and classify the potential of RE for all cities of West Bank. During the work, we encountered several challenges and obstacles. The first challenge was in the data collection phase. We were unable to collect historical data for all Palestinian cities due to a lack of societal awareness about the data storage in many locations that contain solar energy systems. Historical data for Ramallah, Jericho, Qalqilya, and Tulkarem cities are unavailable for this reason. On the other hand, the dataset from the Palestinian Meteorological Department which contains the metrological data for all West Bank cities was incomplete, where the data for Tulkarem, Ramallah, and Jericho not available, even though they are major cities in the West Bank. Also, there is a political difficulty for Jerusalem, which is under the control and management of the occupation. The second difficulties were at the processing data phase. The major challenge was the lack of daily readings of solar energy data. Where the registration tools monitor the reading every 15

minutes throughout the day, which required a long time and effort in preparing and processing the data. This applies to solar radiation data, which gives a reading every 15 minutes throughout the day that includes both night and daytime hours. The last challenge is the unavailability of the wind fields on the West Bank.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

The renewable energy is considered as the most resource that is contributing to the economic development for many communities in addition to solving the electricity sector problems as the continuous shortage of the fossil fuel and the toxic gas emissions. Thus, the prediction of the potential electricity that can be produced from renewable resources is also important for sustainable development for any country. Moreover, the geographical classification depending on renewable resources is also considered an essential reference for the government and community awareness toward these resources. Artificial neural networks are the most popular methods for prediction and classification that provide data processing tools outperform the physical and statistical methods in precision and speed of processing.

In this thesis, we used the artificial neural networks and evolutionary algorithms to develop a GAs-MLPNNs model that is responsible for predicting the potential renewable energy production for each city in the West Bank. The model consists of MLPNNs that are optimized using GAs for prediction, in addition to SOMNNs that is used for classifying cities based on the prediction output. We used a historical dataset of five different PV systems for Jenin, Salfit, Tubas, Bethlehem, and Hebron. Also, we used metrological data that include wind speed, maximum temperature, and solar radiation; to calculate and predict the potential solar and wind power for Jenin, Nablus, Bethlehem, and Hebron. The GAs-MLPNNs model obtains better results using the metrological dataset rather than the use of a historical dataset, where the MSE using the historical dataset was double that for results of the metrological dataset. The GAs-MLPNNs model outperforms the GAs-RBFNNs and PSO-MLPNNs hybrid models for prediction purposes. GAs-MLPNNs obtain the lowest MSE among the other models. For instance, with 40 neurons, the MSE using the proposed model was 0.0048, where the MSE for GAs-RBFNNs was 0.0094 using a dataset of 1460 samples;

while the PSO-MLPNNs model failed to handle the dataset for the power prediction process. SOMNNs also were used to classify West Bank cities according to the annual solar and wind energy production. The results of classification demonstrate that Jenin, Tubas, and Bethlehem have similar potential solar power production, while Nablus, Salfit, and Hebron have diverse capacities of electricity production from the solar resource. The results also show a variety of wind power production for each city of Jenin, Nablus, Bethlehem, and Hebron.

6.2 Future Work

As GAs-MLPNNs model results prove the superiority of the metrological data over historical data for predicting the potential power production from RE resources. This encourages expanding the predictions to include very wide areas, including cities and residential communities, to give an integrated map that shows the potential capacity of the renewable energy systems in the globe using the satellite and GIS data that is available on metrological sites. This will provide the governments and communities with a reference for sustainable development plans in the electricity sector and renewable energy systems. Also, for future works, we aim to investigate more input parameters that can improve the efficiency of the RE prediction. For classification, we plan to replace the SOMNNs as it failed to classify the closed patterns of power production; by Fuzzy inference system for more accuracy, which can show the partial relations between the input and the output.

References

[1] Hussain, A., Arif, S. M., & Aslam, M. (2017). Emerging renewable and sustainable energy technologies: State of the art. Renewable and Sustainable Energy Reviews, 71, 12-28.

[2] Khare, V., Nema, S., & Baredar, P. (2016). Solar–wind hybrid renewable energy system: A review. Renewable and Sustainable Energy Reviews, 58, 23-33.

[3] Blaga, R., Sabadus, A., Stefu, N., Dughir, C., Paulescu, M., & Badescu, V. (2019). A current perspective on the accuracy of incoming solar energy forecasting. Progress in Energy and Combustion Science, 70, 119-144.

[4] Shi, J., Lee, W. J., Liu, Y., Yang, Y., & Wang, P. (2012). Forecasting power output of photovoltaic systems based on weather classification and support vector machines. IEEE Transactions on Industry Applications, 48(3), 1064-1069.

[5] Rodríguez, F., Fleetwood, A., Galarza, A., & Fontán, L. (2018). Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control. Renewable energy, 126, 855-864.

[6] Zhu, H., Lian, W., Lu, L., Dai, S., & Hu, Y. (2017). An improved forecasting method for photovoltaic power based on adaptive BP neural network with a scrolling time window. Energies, 10(10), 1542.

[7] Hu, K., Cao, S., Wang, L., Li, W., & Lv, M. (2018). A new ultra-short-term photovoltaic power prediction model based on ground-based cloud images. Journal of Cleaner Production, 200, 731-745.

[8] Liu, L., Zhao, Y., Chang, D., Xie, J., Ma, Z., Sun, Q., ... & Wennersten, R. (2018). Prediction of short-term PV power output and uncertainty analysis. Applied Energy, 228, 700-711.

[9] Esen, H., Esen, M., & Ozsolak, O. (2017). Modelling and experimental performance analysis of solar-assisted ground source heat pump system. Journal of Experimental & Theoretical Artificial Intelligence, 29(1), 1-17.

[10] Kannan, N., & Vakeesan, D. (2016). Solar energy for future world:-A review. Renewable and Sustainable Energy Reviews, 62, 1092-1105.

[11] Kumar, Y., Ringenberg, J., Depuru, S. S., Devabhaktuni, V. K., Lee, J. W., Nikolaidis, E., ... & Afjeh, A. (2016). Wind energy: Trends and enabling technologies. Renewable and Sustainable Energy Reviews, 53, 209-224.

[12] Murdock, H. E., Gibb, D., André, T., Appavou, F., Brown, A., Epp, B., ... & Sawin, J. L. (2019). Renewables 2019 global status report.

[13] Kreitem, G. M., & Khatib, I. (2018). Renewable energy exploitation in Palestine: current practice & future.

[14] Juaidi, A., Montoya, F. G., Ibrik, I. H., & Manzano-Agugliaro, F. (2016). An overview of renewable energy potential in Palestine. Renewable and Sustainable Energy Reviews, 65, 943-960.

[15] Kitaneh, R., Alsamamra, H., & Aljunaidi, A. (2012). Modeling of wind energy in some areas of Palestine. Energy conversion and Management, 62, 64-69.

[16] Perspectives, S. E. (2011). Executive Summary International Energy Agency. 2011. Archived from the original on, 12-03.

[17] Burnett, D., Barbour, E., & Harrison, G. P. (2014). The UK solar energy resource and the impact of climate change. Renewable Energy, 71, 333-343.

[18] McEvoy, A., Castaner, L., & Markvart, T. (2012). Solar cells: materials, manufacture and operation. Academic Press.

[19] Lee, S. H. (2013). Advancements in n-type base crystalline silicon solar cells and their emergence in the photovoltaic industry. The scientific world journal, 2013.

[20] Wilson, A. (2011, February 28). From Sunlight to Electricity: How Solar Cells Work. Retrieved from https://www.buildinggreen.com/primer/sunlight-electricity-how-solar-cells-work

[21] Furkan, D., & Mehmet Emin, M. (2010). Critical factors that affecting efficiency of solar cells. smart grid and renewable energy, 2010.

[22] Palewicz, M., & Iwan, A. (2011). Photovoltaic phenomenon in polymeric thin layer solar cells. Current Physical Chemistry, 1(1), 27-54.

[23] Sariciftci, N. S., & Sun, S. S. (Eds.). (2005). Organic Photovoltaics: Mechanism, Materials, and Devices. New York: Taylor & Francis.

[24] Dixon, A. E. (1981). PHOTOVOLTAIC ENERGY CONVERSION: THEORY, PRESENT AND FUTURE SOLAR CELLS. In Solar Energy Conversion II (pp. 243-259). Pergamon.

[25] Chikate, B. V., Sadawarte, Y., & Sewagram, B. D. C. O. E. (2015). The factors affecting the performance of solar cell. International journal of computer applications, 1(1), 0975-8887.

[26] Mandati, S., Misra, P., Sarada, B. V., & Rao, T. N. (2019). Copper chalcopyrites for solar energy applications. Transactions of the Indian Institute of Metals, 72(2), 271-288.

[27] Markvart, T., & Castañer, L. (2018). Principles of solar cell operation. In McEvoy's Handbook of Photovoltaics (pp. 3-28). Academic Press.

[28] Tiwari, G. N., Mishra, R. K., & Solanki, S. C. (2011). Photovoltaic modules and their applications: a review on thermal modelling. Applied energy, 88(7), 2287-2304.

[29] A decentralized, renewable-energy-powered business hub for rural areas: A case study of Ilakan community, Nigeria - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Components-of-solar-PV-system12_fig2_301935559 [accessed 30 Aug, 2020].

[30] https://www.greenmatch.co.uk/blog/2014/11/how-efficient-are-solar-panels [accessed 10 Sep, 2020].

[31] Omar, M. A., & Mahmoud, M. M. (2018). Grid connected PV-home systems in Palestine: A review on technical performance, effects and economic feasibility. Renewable and Sustainable Energy Reviews, 82, 2490-2497.

[32] https://www.weatherwizkids.com/weather-wind.htm [accessed 5 Aug, 2020].

[33] https://www.noaa.gov/education/resource-collections/weather-atmosphere/weather-systems-patterns [accessed 10 Sep, 2020].

[34] Anaya-Lara, O., Jenkins, N., Ekanayake, J. B., Cartwright, P., & Hughes, M. (2011). Wind energy generation: modelling and control. John Wiley & Sons.

[35] Price, T. J. (2005). James Blyth—Britain's first modern wind power pioneer. Wind engineering, 29(3), 191-200.

[36] Wu, B., Lang, Y., Zargari, N., & Kouro, S. (2011). Power conversion and control of wind energy systems (Vol. 76). John Wiley & Sons.

[37] Pieralli, S., Ritter, M., & Odening, M. (2015). Efficiency of wind power production and its determinants. Energy, 90, 429-438.

[38] Blackwood, M. (2016). Maximum efficiency of a wind turbine. Undergraduate Journal of Mathematical Modeling: One+ Two, 6(2), 2.

[39] www.aaup.edu [accessed 10 May, 2019].

[40] www.monitoring.solaredge.com [accessed 30 Oct, 2019].

[41] www.pmd.ps [accessed 31 Dec, 2019].

[42] Anwar, K., & Deshmukh, S. (2018). Assessment and mapping of solar energy potential using artificial neural network and GIS technology in the southern part of India. International Journal of Renewable Energy Research (IJRER), 8(2), 974-985.

[43] Chine, W., Mellit, A., Lughi, V., Malek, A., Sulligoi, G., & Pavan, A. M. (2016). A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks. Renewable Energy, 90, 501-512.

[44] Puri, V., Jha, S., Kumar, R., Priyadarshini, I., Abdel-Basset, M., Elhoseny, M., & Long, H. V. (2019). A hybrid artificial intelligence and internet of things model for generation of renewable resource of energy. IEEE Access, 7, 111181-111191.

[45] Bou-Rabee, M., Sulaiman, S. A., Saleh, M. S., & Marafi, S. (2017). Using artificial neural networks to estimate solar radiation in Kuwait. Renewable and Sustainable Energy Reviews, 72, 434-438.

[46] Zhou, S., Hu, Z., Gu, W., Jiang, M., & Zhang, X. P. (2019). Artificial intelligence based smart energy community management: A reinforcement learning approach. CSEE Journal of Power and Energy Systems, 5(1), 1-10.

[47] Zhao, X., Wang, C., Su, J., & Wang, J. (2019). Research and application based on the swarm intelligence algorithm and artificial intelligence for wind farm decision system. Renewable energy, 134, 681-697.

[48] Haupt, S. E., McCandless, T. C., Dettling, S., Alessandrini, S., Lee, J. A., Linden, S., ... & Wiener, G. (2020). Combining Artificial Intelligence with Physics-based Methods for Probabilistic Renewable Energy Forecasting. Energies, 13(8), 1979.

[49] Said, M. I., Steiner, M., Siefer, G., & Arab, A. H. (2020). Maximum power output prediction of HCPV FLATCON® module using an ANN approach. Renewable Energy, 152, 1274-1283. Inivan, S., Jebaraj, S., Suganthi, L., & Samuel, A. A. (2020). Energy models for renewable energy utilization and to replace fossil fuels. Methodology, 67, 28-37.

[50] Inivan, S., Jebaraj, S., Suganthi, L., & Samuel, A. A. (2020). Energy models for renewable energy utilization and to replace fossil fuels. Methodology, 67, 28-37.

[51] Heydari, A., & Keynia, F. (2016). Prediction of wind power generation through combining particle swarm optimization and elman neural network (El-PSO). International Energy Journal, 15(2).

[52] Quan, D. M., Ogliari, E., Grimaccia, F., Leva, S., & Mussetta, M. (2013, July). Hybrid model for hourly forecast of photovoltaic and wind power. In 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1-6). IEEE.

[53] Lee, D., & Kim, K. (2019). Recurrent neural network-based hourly prediction of photovoltaic power output using meteorological information. Energies, 12(2), 215.

[54] Dumitru, C. D., Gligor, A., & Enachescu, C. (2016). Solar photovoltaic energy production forecast using neural networks. Procedia Technology, 22, 808-815.

[55] Fadzail, N. F., & Zali, S. M. (2019). Fault detection and classification in wind turbine by using artificial neural network. International Journal of Power Electronics and Drive Systems, 10(3), 1687.

[56] Shi, J., Lee, W. J., Liu, Y., Yang, Y., & Wang, P. (2012). Forecasting power output of photovoltaic systems based on weather classification and support vector machines. IEEE Transactions on Industry Applications, 48(3), 1064-1069.

[57] Fadda, M. L., & Moussaoui, A. (2018). Hybrid SOM–PCA method for modeling bearing faults detection and diagnosis. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40(5), 268.

[58] An Ultra-short-term Forecasting Model for High-resolution Solar Irradiance Based on SOM and Deep Learning Algorithm.

[59] Sheela, K. G., & Deepa, S. N. (2013). Neural network based hybrid computing model for wind speed prediction. Neurocomputing, 122, 425-429.

[60] Graditi, G., Ferlito, S., & Adinolfi, G. (2016). Comparison of Photovoltaic plant power production prediction methods using a large measured dataset. Renewable Energy, 90, 513-519.

[61] Doucoure, B., Agbossou, K., & Cardenas, A. (2016). Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. Renewable Energy, 92, 202-211.

[62] Jamil, W., Kalnishkan, Y., & Bouchachia, A. (2016). Aggregation algorithm vs. average for time series prediction.

[63] Inman, R. H., Pedro, H. T., & Coimbra, C. F. (2013). Solar forecasting methods for renewable energy integration. Progress in energy and combustion science, 39(6), 535-576.

[64] Perry, M. B. (2010). The Weighted Moving Average Technique. Wiley Encyclopedia of Operations Research and Management Science.

[65] Valipour, M., Banihabib, M. E., & Behbahani, S. M. R. (2013). Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir. Journal of hydrology, 476, 433-441.

[66] Buitrago, J., & Asfour, S. (2017). Short-term forecasting of electric loads using nonlinear autoregressive artificial neural networks with exogenous vector inputs. Energies, 10(1), 40.

[67] Yang, X., Fu, G., Zhang, Y., Kang, N., & Gao, F. (2017). A naive Bayesian wind power interval prediction approach based on rough set attribute reduction and weight optimization. Energies, 10(11), 1903

[68] https://www.saurenergy.com/solar-energy-blog/here-is-how-you-can-calculate-the-annual-solar-energy-output-of-a-photovoltaic-system [accessed 15 Mar, 2021].

[69] Osama, O. (2001). Peak-Load Shaving by Decentralized Grid-Connected Photovoltaic (PV) Systems (Master thesis, Bolton Institute of Higher Education).

[70] Sarkar, A., & Behera, D. K. (2012). Wind turbine blade efficiency and power calculation with electrical analogy. International Journal of Scientific and Research Publications, 2(2), 1-5.

[71] Kazem, H. A., Yousif, J. H., & Chaichan, M. T. (2016). Modeling of daily solar energy system prediction using support vector machine for Oman. International Journal of Applied Engineering Research, 11(20), 10166-10172.

[72] Mohandes, M. A., Halawani, T. O., Rehman, S., & Hussain, A. A. (2004). Support vector machines for wind speed prediction. Renewable energy, 29(6), 939-947.

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

[74] Voyant, C., Muselli, M., Paoli, C., & Nivet, M. L. (2012). Numerical weather prediction (NWP) and hybrid ARMA/ANN model to predict global radiation. Energy, 39(1), 341-355.

[75] Paoli, C., Voyant, C., Muselli, M., & Nivet, M. L. (2010). Forecasting of preprocessed daily solar radiation time series using neural networks. Solar energy, 84(12), 2146-2160.

[76] Cadenas, E., & Rivera, W. (2010). Wind speed forecasting in three different regions of Mexico, using a hybrid ARIMA–ANN model. Renewable Energy, 35(12), 2732-2738.

[77] Love, B. C. (2002). Comparing supervised and unsupervised category learning. Psychonomic bulletin & review, 9(4), 829-835.

[78] Anderson, D., & McNeill, G. (1992). Artificial neural networks technology. Kaman Sciences Corporation, 258(6), 1-83.

[79] Rosenblatt, F. (1957). The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory.

[80] Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. Machine learning, 37(3), 277-296.

[81] Sharma, S. (2017). Activation functions in neural networks. Towards Data Science, 6.

[82] Che, Z. G., Chiang, T. A., & Che, Z. H. (2011). Feed-forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm. International journal of innovative computing, information and control, 7(10), 5839-5850.

[83] Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

[84] Zhang, Y., Wang, S., Ji, G., & Phillips, P. (2014). Fruit classification using computer vision and feedforward neural network. Journal of Food Engineering, 143, 167-177.

[85] Jia, W., Zhao, D., Shen, T., Su, C., Hu, C., & Zhao, Y. (2014). A new optimized GA-RBF neural network algorithm. Computational intelligence and neuroscience, 2014.

[86] https://towardsdatascience.com/understand-neural-networks-model-generalization-7baddf1c48ca [accessed 2 Aug, 2020].

[87] BOUZGOU, H. (2012). Advanced Methods for the Processing and Analysis ofMultidimensional Signals: Application toWind Speed (Doctoral dissertation, Université de Batna 2).

[88] Dupond, S. (2019). A thorough review on the current advance of neural network structures. Annual Reviews in Control, 14, 200-230.

[89] https://medium.com/hackernoon/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860 [accessed 15 Mar, 2021].

[90] Kohonen, T., & Honkela, T. (2007). Kohonen network. Scholarpedia, 2(1), 1568.

[91] Ali, A (2019, may 26). From self organize map (SOM), retrieved from https://mc.ai/self-organizing-mapsom/. Copy right by Deep Learning on Medium. [accessed 17 Sep, 2020].

[92] Vikhar, P. A. (2016, December). Evolutionary algorithms: A critical review and its future prospects. In 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC) (pp. 261-265). IEEE.

[93] Er, M. J., & Liu, F. (2009). Parameter Tuning of MLP Neural Network Using Genetic Algorithms. In The Sixth International Symposium on Neural Networks (ISNN 2009) (pp. 121-130). Springer, Berlin, Heidelberg.

[94] Lim, S. M., Sultan, A. B. M., Sulaiman, M. N., Mustapha, A., & Leong, K. Y. (2017). Crossover and mutation operators of genetic algorithms. International Journal of Machine Learning and Computing, 7(1), 9-12.

[95] Sahin, Ferat & Abbate, Giuseppe. (2004). Genetic algorithm parameter optimization: applied to sensor coverage. Proceedings of SPIE - The International Society for Optical Engineering. 10.1117/12.542404.Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. Soft Computing, 22(2), 387-408.

[96] Jia, W., Zhao, D., Shen, T., Su, C., Hu, C., & Zhao, Y. (2014). A new optimized GA-RBF neural network algorithm. Computational intelligence and neuroscience, 2014.

الملخص

منذ عدة عقود، استخدمت مصادر الطاقة المتجددة كبديل مستدام وصديق للبيئة لتعويض النقص في موارد الطاقة التقليدية. تعد أنظمة الطاقة الشمسية وطاقة الرياح أكثر مصادر الطاقة المتجددة شيوعًا لإنتاج الكهرباء. هذه الأنظمة ذات طبيعة متقلبة نظراً لتأثرها بالظروف المناخية. لذا فإن التنبؤ الدقيق بكمية الطاقة المحتملة في موقع معين أمر مجدي اقتصاديًا لتصميم وتركيب أنظمة الطاقة المتجددة. ولهذا السبب، من الضروري وجود نموذج للتنبؤ والتصنيف لتنظيم وإدارة هذا القطاع في فلسطين والدول الأخرى. تقترح هذه الرسالة نموذجًا مزدوج من الشبكات العصبية وخوارزميات التحسين التي تستخدم البيانات التاريخية وبيانات الأرصاد الجوية للتنبؤ بكمية الطاقة الكهربائية الممكن انتاجها من المصادر المتجددة في فلسطين. تم جمع بيانات إنتاج الطاقة التاريخية من مواقع مختلفة في فلسطين تشمل الخليل في الجنوب وبيت لحم وسلفيت في الوسط وطوباس وجنين في شمال الضفة الغربية. كما جمعت بيانات الأرصاد الجوية باستخدام دائرة الأرصاد الجوية الفلسطينية لمدن جنين ونابلس وبيت لحم والخليل. يبين أداء النموذج أن الخوارزمية الجينية تتفوق على غيرها من خوارزميات التحسين في بناء الشبكات العصبية متعددة الطبقات وتحسين معاملاتها للتنبؤ بإنتاج الطاقة.

حيث أن نظام النموذج الذي يجمع بين الخوارزمية الجينية GAs والشبكة العصبية متعددة الطبقات MLPNNs تفوق على النوذج الذي يجمع خوارزمية تحسين سرب الجسيمات PSO مع MLPNNs ونموذج RBFNNs مع GAs لـ توقع إنتاج الطاقة. حيث حقق GAs-MLPNNs أدنى معدل خطأ بين النماذج الأخرى. على سبيل المثال، باستخدام 40 خلية عصبية ، كان MSE باستخدام النموذج المقترح 0.0048، اما MSE لـ GAs-RBFNNs 0.0094 باستخدام مجموعة بيانات لأربع سنوات؛ بينما فشل نموذج PSO-MLPNNs في التعامل مع مجموعة البيانات. يوضح نموذج GAs-MLPNNs أيضًا أنه يمكن الحصول على نتائج أفضل باستخدام بيانات الارصاد الجوية بدلاً من استخدام مجموعة البيانات التاريخية ، حيث كان MSE باستخدام مجموعة البيانات التاريخية ضعف ذلك بالنسبة لنتائج بيانات الأرصاد الجوية. كما تم استخدام نتائج النموذج المستخدم لتصنيف المناطق الجغرافية حسب إنتاجها المحتمل للطاقة. استخدمنا خوارزمية تصنيف عامة؛ خريطة التنظيم الذاتي SOM لتصنيف المناطق بناءً على الإنتاج السنوي المتوقع للطاقة. تظهر نتائج التصنيف أن جنين وطوباس وبيت لحم لديها إمكانات مماثلة لإنتاج الطاقة الشمسية، في حين أن نابلس وسلفيت والخليل لديها قدرات متنوعة لإنتاج الكهرباء من مصادر الطاقة الشمسية. كما أظهرت النتائج تنوعًا في إنتاج طاقة الرياح لكل مدينة في جنين ونابلس وبيت لحم والخليل.