



**Arab American University- Palestine
Faculty of Graduate Studies**

**Electricity Consumption Prediction Using Artificial
Neural Networks and Evolutionary Algorithms: A
Case Study Tulkarm City**

By

ALI MOHAMMAD ALI HUSSEIN

Supervisor **Prof. Dr. Mohammed Awad**

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

August 2021

© Arab American University-Palestine 2021

All rights reserved

All rights reserved

**Electricity Consumption Prediction Using Artificial Neural
Networks and Evolutionary Algorithms: A Case Study Tulkarem
City**

By

Ali Mohammad Ali Hussein

This thesis was defended successfully on **00/08/2021** and approved by:

Committee Members	Signature
Supervisor: Prof. Dr. Mohammed Awad .1	
Internal Examiner: Dr. Ahmad Ewais .2	
External Examiner: Dr. Yousef Daraghmi .3	

Declaration

I declare that this thesis is an original report of my research, has been written by me under the guidance of Prof. Dr. Mohammed Awad and has not been submitted for any previous degree. The collaborative contributions have been indicated clearly and acknowledged. Due references have been provided on all supporting literatures and resources. I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Signature : Ali Hussein



18/11/2021

Acknowledgments

I would like to thank Professor Mohammad Awad for his guidance, encouragement and excellent advice throughout this study. This would not have been possible without his expertise.

I would also like to thank Tulkarm municipality, for providing me with the required data to complete this study.

Declaration

I declare that this thesis is an original report of my research, has been written by me under the guidance of Prof. Dr. Mohammed Awad and has not been submitted for any previous degree. The collaborative contributions have been indicated clearly and acknowledged. Due references have been provided on all supporting literatures and resources. I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Signature : Ali Hussein

A handwritten signature in blue ink, appearing to be 'Ali Hussein', with a stylized, cursive script.

Abstract

Electricity consumption is one of the main concerns in any country; as without planning the process of the electricity distribution, there will be major problems in the economy and all the fields that affect daily life. Electric power demand prediction is a systematic procedure that allows the quantitative definition of future demand and, as it is of vital importance depending on the period that is adopted in the analysis. This study presents a model based on a supervised learning approach to predict future electricity consumption. Thus, can help power companies to plan the future demand and to guarantee the service continuation. For this purpose, real data has been collected periodically (monthly, seasonal and yearly) from Tulkarm municipality through the years 2018 to 2020 forming a sample of data points.

In this thesis, we applied the collected datasets on different ANNs approaches from the traditional ones to hybrid novel models. Multilayer Perceptron Neural Networks model (MLPNNs), RNNs, and NARX have been selected from the traditional approaches, along with genetic algorithms integrated with K mean clustering for producing specific initial population seeding techniques and optimizing crossover operators to enhance the efficiency and find the optimal solution. The results showed that RNNs outperformed the other models in terms of short term memory (STM) with Root Mean Square Error (RMSE value = $5.82 \cdot 10^{-24}$, while the hybrid models (Recurrent Neural Networks with Optimized Algorithm (RNNs-OA), Nonlinear Autoregressive Exogenous with Optimized Algorithm (NARX-OA), and Nonlinear Autoregressive Exogenous with Optimized Algorithm with K Mean Clustering (NARX-OA-K Mean cluster)) gave better results, for example when executing 8 neurons for one-year prediction, RNN-OA recorded RMSE value = 0.28128, NARX-OA recorded RMSE value = 0.10382 and NARX-OA-K Mean cluster recorded RMSE value = 0.08759 which perform a good balance with the lowest RMSE especially in the long term forecasting and also outperforms other hybrid prediction models.

Table of Contents

Declaration	I
Acknowledgments	II
Abstract	IV
Table of Contents	V
List of Figures.....	VII
List of Tables.....	IX
List of Abbreviations	X
1. Introduction.....	2
1.1 Problem Statement.....	5
1.2 Objectives.....	5
1.3 Contribution.....	6
1.4 Overview.....	6
2. Background.....	9
2.1 Energy Sector in Palestine	9
2.2 Dataset Description	11
2.3 Literature Review	13
2.4 Summary of Related Works	19
3. The Proposed Method.....	22
3.1 Introduction	22
3.2 Dataset Preparation Process	24
3.3 Applied Models	25
3.3.1 Genetic Algorithms.....	26
3.3.2 K-mean Clustering.....	27
3.3.3 Recurrent Neural Networks – OA (RNNs-OA) Model.....	28
3.3.4 Nonlinear Autoregressive Neural Networks– OA (NARX-OA)	33
3.4 Optimized Approach	37
3.5 Summary	48
4. Result and Discussion.....	50
4.1 Electricity Dataset	50
4.2 Experiment Procedure	51
4.3 Results	60
4.3.1 Multilayer -Perceptron Model (MLP):.....	60
4.3.2 Recurrent Models	63
4.3.3 Hybrid Models	69

4.3.4 Current Intensity prediction	82
4.4 Trial Data Test	85
4.5 Comparison Between Conventional and Hybrid Models	88
4.6 Challenges and Limitation.....	91
5. Conclusion and Future Work.....	94
5.1 Conclusion.....	94
5.2 Recommendations.....	95
5.3 Future work.....	96
References.....	97

List of Figures

Figures	Page
Figure 2.1 Electricity Resources in Palestine	9
Figure 2.2 Main Electricity Problems in Palestine	10
Figure 2.3 Monthly Electricity Consumption in kWh for Tulkarm Through Lines 1 and 2 in 2018	11
Figure 2.4 Total Monthly Electricity Consumption in kWh Through Lines 1 and 2 in 2018	12
Figure 2.5 Electricity Consumption in kWh for Every July Month in 2018-2020	13
Figure 2.6 Current Intensity in Ampere Through Line 2 for Every Day of July 2020	13
Figure 3.1. Primary structure of the proposed hybrid models	23
Figure 3.2 Flowchart of Recurrent Neural Networks	30
Figure 3.3 General structure of the RNNs	31
Figure 3.4 the RNNs-OA Hybrid Model	32
Figure 3.5 Types of training architecture for NARX neural networks	35
Figure 3.6 the NARX-OA Hybrid Model	36
Figure 3.7 Recurrence Model with Enhanced Strategy (Hybrid Model)	39
Figure 3.8 Basic Structure of Chromosome	41
Figure 3.9 the workflow of cluster algorithm	42
Figure 3.10 Two methods for initializing the population of problems.	42
Figure 3.11 16- Period Cluster crossover	45
Figure 3.12 The Affected Mutation Rate	46
Figure 4.1: Real and predicted June month using MLP	61
Figure 4.2 Real and predicted electricity consumption values using MLP, and the error value	62
Figure 4.3: Real and Predicted Summer Line 1 based on the average of 3 years using MLP	62
Figure 4.4: Real and predicted line 1+2 using MLP	63
Figure 4.5: Real and predicted June month using NARX	64
Figure 4.6: Real and predicted summer line 1 of an average of three years	65
Figure 4.7: Real and predicted Line 1+2 using NARX model	66
Figure 4.8: Real and predicted value June month using RNN	67
Figure 4.9: Best Prediction Result of the Training Process for One-Season Using RNN	68
Figure 4.10 Prediction Result of the Training Process for One Year	69
Figure 4.11 Prediction Result of the Training Process for One Month	70
Figure 4.12 Prediction Result of the Training Process for Summer Season	72
Figure 4.13 Prediction Result of the Training Process for Winter Season	72
Figure 4.14: Prediction Result of the Training Process for One Year	74

Figure 4.15: Prediction Result of the Training Process for One Month using NARX-OA K Mean clustering	75
Figure 4.16 Prediction Result of the Training Process for One Month using NARX-OA	76
Figure 4.17 Prediction Result of the Training Process for One Season using NARX-OA	77
Figure 4.18 Prediction Result of the Training Process for Summer Season using NARX-OA-K Mean clustering model	77
Figure 4.19: Prediction Result of Training Process for One season (Winter Season) using NARX-OA- K Mean Clustering.	79
Figure 4.20: Prediction Result of Training Process for One Season (Winter Season) using NARX-OA	79
Figure 4.21 Prediction Result of Training Process for One Year using NARX-OA-K Mean Clustering.	81
Figure 4.22: Prediction Result of Training Process for One Year using NARX-OA	82
Figure 4.23 Prediction Result of Training Process for Current Intensity using NARX-OA-K Mean Clustering	84
Figure 4.24 \Prediction Result of Training Process for current intensity using NARX-OA.	84
Figure 4.25 Prediction Result of Training Process for current intensity using RNN-OA.	85
Figure 4.26 Prediction Result of Trial Dataset Training Process using Modified Model (RNN-OA)	86
Figure 4.27 Prediction Result of Training Process for Trial Dataset using NARX-OA	87
Figure 4.28 Prediction Result of Training Process for Trial Dataset using modified Model (NARX-OA- K-mean cluster)	87
Figure 4.29 Large Data Prediction using RNN-OA Model	89
Figure 4.30 Large Data Prediction using RNN	90
Figure 4.31 Comparison Between Hybrid Models for One-Year Forecasting	91

List of Tables

Table	Page
Table 2.1 Some of the electricity consumption forecasting study in the world	20
Table 3.1 The Mutation Constant Multiplied	45
Table 4.1 Dataset Description	51
Table 4.2 Selection population size based on results of recurrent hybrid model	53
Table 4.3 Selection Generation based on results of recurrent hybrid model (Electricity L1+2)	55
Table 4.4 Selection Generation based on results of recurrent hybrid model (Electricity L1+2 dataset)	55
Table 4.5 Selection Generation based on results of recurrent hybrid model (Summer Electricity L 1+2 dataset)	56
Table 4.6 Selection crossover and mutation probabilities based on results of recurrent hybrid model	57
Table 4.7 Selection crossover and mutation probabilities based on results of recurrent hybrid model (Electricity Line 1+2)	58
Table 4.8 Selection crossover and mutation probabilities based on results of recurrent hybrid model (Summer Electricity Line 1+2)	59
Table 4.9 General Characteristics of the RNNs-OA hybrid model for the dataset	60
Table 4.10 Result of June month MSE for MLP	61
Table 4.11 Results of June month MSE for NARX	64
Table 4.12 Model Training Result for Energy Prediction at June (One-Month)	66
Table 4.13 Model Training Result for Energy Prediction at Summer Line 1 Using RNN	67
Table 4.14 Model Training Result for Energy Prediction (June month)	70
Table 4.15 Model Training Result for Energy Prediction (Summer Season)	71
Table 4.16 Model Training Result for Energy Prediction	73
Table 4.17 Model Training Result for Energy Prediction (June Month)	75
Table 4.18 Model Training Result for Energy Prediction (Summer Season).	76
Table 4.19 Model Training Result for Energy Prediction (Winter Season)	78
Table 4.20 Model Training Result for Energy Prediction	80
Table 4.21 Model Training Result for Energy Prediction (Large Data)	83
Table 4.22 Results of the training model using RNNs	88

List of Abbreviations

Abbreviation	Full name
AI	Artificial Intelligence
ANNs	Artificial Neural Networks
NN	Nearest Neighbour
NNs	Neural Networks
MLP	Multilayer Perceptron
ML	Machine Learning
EAs	Evolution Algorithms
RNNs	Recurrent Neural Networks
NARX	Nonlinear Auto-Regressive Exogenous
BPA	Back Propagation Algorithm
LSTM	Long Short-Term Memory
RMSE	Root Mean Square Error
LMA	Levenberg-Marquardt Algorithm
Gas	Genetic Algorithms
SSE	Sum Square Error
IEC	Israel Electrical Company
MLPNNs	Multilayer Perceptron Neural Networks
MAPE	Mean Absolute Percentage Error
SOM	Self-Organizing Map
ARIMA	Auto-Regressive Integrated Moving Average
TDL	Time Delay Lock
RNN-OA	Recurrent Neural Network with Optimized Algorithm
NARX-OA	Nonlinear Autoregressive Exogenous with Optimized Algorithm
NARX-OA-K MEAN CLUSTERING	Nonlinear Autoregressive Exogenous with Optimized Algorithm with K Mean Clustering
PENRA	Palestinian Energy and Natural Resources Authority
FRNN	Fully Recurrent Neural Network

Chapter 1

Introduction

1. Introduction

Electricity has drastically changed the way people live, as it is used to do many everyday tasks from heating, lighting, cooling homes to powering televisions and computers, also for industrial uses and so much more. Electricity has an effective role in developing economies and societies. Thus, the supply of electricity must meet the individual's increasing demand. For this reason, planning for this demand is essential for any social development along with helping to establish pricing policy, designing distribution systems, and planning for power generation, so governments and electricity companies have paid great attention towards the prediction of electricity consumption especially over different parameters such as the weekdays, seasons, weather conditions, etc (Yang, W., et al 2020).

Demand planning for electricity consumption is a key success factor for the development of countries (Trotter, P.A., et al 2017). Due to the importance of electricity, countries have greatly paid attention to the prediction of electricity consumption. Electricity Consumption prediction is a major problem for the power sector, from their results, the electric company can make the right decisions for their work to maximize the profit.

Palestine is considered a developing country and an occupied one, as it is mostly dependent on Israel's occupation for electricity supply, although the demand has increased significantly due to the rapid increase in population, buildings, industries, and business. Which impose many challenges, such as energy insecurity, high fuel prices, and the lack of sustainable consumption (Al Qadi, et al. 2018). Even though occupation policies impose many obstacles regarding a pricing policy, as there is no fair competition when making comparisons with other countries. Thus, we as Palestinians had to find other ways and studies that limit electricity consumption and find alternative energy solutions.

Because the energy demand is increasing, while the energy supply is insufficient, some researchers have suggested using renewable energy at homes and factories, others suggested using electrical generators to produce electricity instead of the original resources to help solve the electricity problem. One of the studies that discussed alternative ways to supply energy to remote Palestinian villages and rural areas using various solutions, such as electric grids, PV systems, and diesel generators (Mahmoud and Ibrik, 2006), However, each has some limitations.

This study takes Tulkarm as an example city of Palestine and uses data for electricity readings of 2018-2020 from the Tulkarm municipality, which is responsible for supplying electricity to seven small outskirts and two large camps for free. Tulkarm's weather has warm winters and very hot summers with humidity as it is located near the Mediterranean Sea north of the West Bank and has a population of approximately 44,169 (worlddata.info 2019). Thus, due to its weather conditions, the natural increase in population which leads to an increase in different facilities which adds more electricity load and relying on one municipality for electricity supply, the city suffers from a shortage in electricity supply, and frequent power interruptions during summer.

Tulkarm municipality issued a report discussing two main reasons for power interruption in Tulkarm (Tulkarm-Municipality. 2018). The first reason behind the power interruption is the municipality itself in the form of programmed power interruptions for maintenance, un-programmed power interruptions in case of electricity problems, and in case of increasing power loads, the management of the electricity demand switch off power lines that supply specific parts of the city for approximately 13 sec to a maximum of 5 min. however, if the power demand continues to be higher than the power supplied, then the power interruption may last for more than an hour for some parts of the city. The second reason behind the power

interruption is the Israeli electrical company, in the form of programmed interruptions with notification to the municipality or un-programmed interruptions, without notification, which could last for hours (Tulkarm-Municipality 2018).

As a result, the municipality started taking actions to reduce the electricity interruptions, such as using diesel generators during power interruptions, schedule the time of electricity use based on region importance. In addition, people suggested using solar energy in their houses to reduce their monthly bills and avoid power interruptions, which can harm sensitive electricity devices. However, each solution can have some limitations, like diesel generators are too expensive and noisy. Continuous power interruptions cause damage to electric devices (Salah, Wael A.; Abuhelwa, Mai, 2020). Therefore, other alternative solutions need to be found.

According to the literature prediction techniques mainly depends on forecasting future data based on learning from previous data. There are conventional models and artificial intelligence (AI)-based models which are widely used for energy prediction. The conventional models are categorized into time series models, regression models, and grey models. The AI-based models are grouped into artificial neural network-based models and support vector regression machine-based (Fouquier et al., 2013). A study by (Wei et al., 2019c) proved that when properly selecting factors and adjusting the parameters weight, then the conventional models can produce results as accurate as of the AI models in terms of electricity consumption prediction. The difference between these two models is that the conventional models do not rely on historical data for making predictions, instead, it constructs a relationship between energy consumption and its influencing factors, such as temperature, gross domestic product (GDP), and population (Deb et al., 2017).

While AI models learn from a large amount of historical data for prediction. One of these techniques is the time-series analysis model, which considers the effect of load for the previous

years on the load for the year in question. The order of this time difference series depends on the accuracy of the prediction as well as the data available from history. Some of the most popular time series forecasting techniques are Autoregressive Integrated Moving Average (ARIMA), Support Vector Machines (SVM), Case-Based Reasoning (CBR), Fuzzy Time Series, Grey prediction model, Moving average and exponential smoothing (MA & ES), and Hybrid models. (Deb, C., et al. 2017)

In this study, we are working on a model that predicts the electricity consumption for each area individually based on previous data using AI techniques. Where decision-makers can predetermine the electrical load and avoid any electricity disruptions, along with financial and human losses due to electricity weakness. This can be done by following up the electrical network and periodic maintenance based on the predicted future readings. Furthermore, regulating electricity throughout the year. resulting in less dependability on foreign resources.

The prediction model proposed in this thesis is represented by a hybrid model that combines NNs with the EAs model whose purpose is to help easily verify the results and provide a value of demanded energy as close to the real future value. Neural Networks (NNs) are inspired by the biological neural networks of the human brain and are made up of elements that behave similarly to the biological neuron in their most common functions. These elements are organized in a way similar to that of the human brain. Neural Networks can be applied to prediction problems, due to their exclusive ability as a mathematical approximation of unknown functions, and classifying patterns, which make them a tool with great advantages over conventional prediction methods.

The development of an NNs model depends on the EAs techniques to discover the main characteristics of the time series to predict their behavior in the future. These techniques will

be applied to the prediction of electricity consumption in Tulkarm city depending on the past /current consumption.

1.1 Problem Statement

The rapid growth of population contributed to an unparalleled increase in the demand, prices, and supply of electrical energy, which resulted in a permanent problem in the electricity in Tulkarm city. Thus, the unpredictability of the energy supply would lead to high consumption and difficulty in setting sustainability goals that can help with energy-saving and environmental protection. Up to this date, there are no-load forecasting studies to predict the possible consumption of energy, nor is there a reliable system of distribution in the city. In general, the electrical markets used models. Also, the need for accurate electricity demand prediction models will help to decrease the risk of effects of an operational and economic nature.

In recent years' artificial intelligence has been applied in the fields of demand forecasting of energy so that in this research a hybrid system of artificial neural networks and EAs will be developed for forecasting electricity demand in the short term.

1.2 Objectives

The basic objective of this thesis is to develop a hybrid model that combines ANNs and EAs that allows an efficient prediction of demand for electricity using a time series for daily, monthly and yearly predictions.

This general objective has been specified based on the following specific objectives:

- Collect and process the energy consumption database of Tulkarm city.
- Design a model for the prediction of electricity demand with adequate precision using a previous analysis provided by an optimized hybrid artificial intelligent model.

1.3 Contribution

This thesis present different hybrid neural networks NNs methods which are expected to have the following contributions:

1. Enhance the performance of the electricity through the predetermined electricity consumption which will help the municipality to devolpe the electricity network.
2. Choosing the suitable optimization algorithms and combining ANNs techniques to provide a new mechanism with particular features to get more accuracy and less RMSE.
3. Comparison between the applied ANN's hybrid models conducted in our dataset to assess which of them have better performance and to find the most efficient hybrid model for the prediction of electricity demand in Tulkarm city as a time series method.

1.4 Overview

The remainder of this thesis is organized into a total of five chapters, including this chapter as an introduction. The second chapter introduce the energy sector in Palestine, a description of datasets, then a literature review of the related works. In Chapter 3, the description of the dataset preparation process is introduced to the reader along with the applied models and the optimized approach.

Chapter 4, shows all results obtained from all models in prediction, in addition to applying our model on a different dataset, then a comparison between conventional and hybrid models is discussed. Chapter 5, summarize the results and give recommendations with further work.

Chapter 2

Background

2. Background

In Palestine, development is considered hard as Israel follows rules of colonial occupation approaches that impose constraints on goods import and export, preventing Palestinians from reaching their natural resources, distributing and isolating areas, obstacles in movements, imposing taxes, and much more (Marei, Ibrahim 2017) (Zeid, M. and Thawaba, S., 2018).

Thus, Palestinians suffer from the lack of energy resources, high growth of population, and high cost of electricity, Which led to finding solutions for the energy crises. In 2018, the energy demand reached 5800 GWH (Juaidi, A., et al 2016). The Israel Electrical company IEC covered about 92.6%, Jordan 1.59%, Egypt 0.6 %, Gaza power plant, and renewable energy resources 0.9% from demand as shown in figure 2.1 (Marei, Ibrahim 2017). The high electricity importation from IEC exhausts the government of Palestine about 574 million dollars. On the other hand, the cost of electricity in Palestine is considered high, CoE in Palestine is about 0.6215 ILS/KWH for the residential sector whereas in Israel it is 0.4516 ILS/KWH. As expected the electricity shortage will increase in the West bank as high as 3.5% till 2030(Juaidi, A., et al 2016) (Marei, Ibrahim., et al 2017).

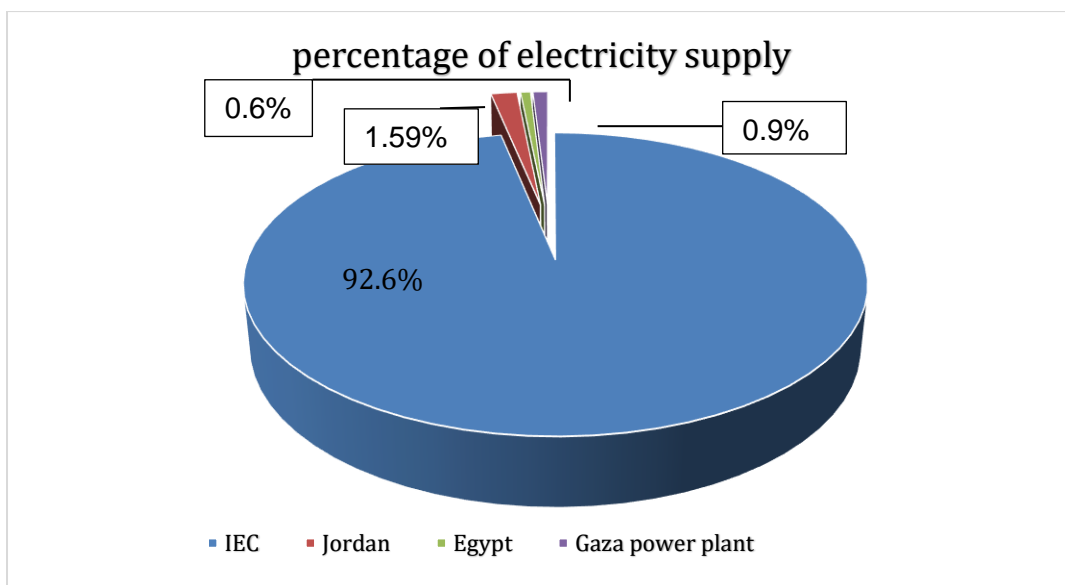


Figure 2.1 Electricity Resources in Palestine

The energy sector situation is different compared with other different countries in the Middle East because of many reasons: unstable politics, rapid growth in the industrial sector and the 100% total dependency on importation of fossil fuel and more than 87% of electricity from Israel (Zeid, M. and Thawaba, S., 2018) (Juaidi, A., et al.2016). The total energy consumption for each person in Palestine is the least in the region with 0.79 KWH/ inhabitant (10 times less than Israel) and at the same time has the highest cost from other countries in the Middle East. As a result, investment in the energy sector is the main drive to attain development and economic growth in the form of upgrading and expansion to meet current demand.

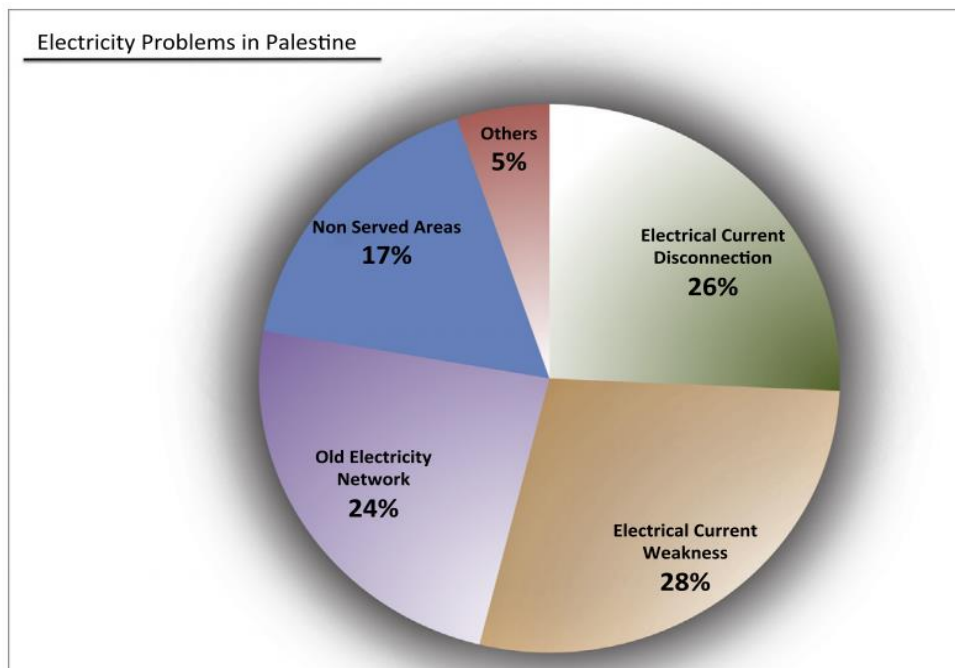


Figure 2.2 Main Electricity Problems in Palestine

Figure 2.2 summarizes the main electricity problems in Palestine which are classified to 26% of electricity current disconnection, 24% for the old electricity network, 28% for current weakness, 17% for rural areas, and 5% for others (Juaidi, A, et al.2016).

This study aims to optimize the service of electricity and improve its performance by using smart systems with concern on the neural network because of its ability to learn through examples and predictions.

2.2 Dataset Description

Tulkarm's main source of electricity is provided by the Israeli main electricity company that supplies the city with two connecting points (line 1 and line 2), each line is supplied through 430 Amp medium voltages as in figure 2.3 that shows the monthly electricity consumption in kWh for Tulkarm through lines 1 and 2, also with the total amount of electricity equal to 860 Amp as shown in figure 2.4. In addition, there are different supplementary sources of energy including solar energy with a total of 0.8 MW. Based on the strategic plan of the Palestinian energy sector, the renewable source should be equivalent to 10% of locally produced electricity by 2020 (Palestine News & Information Agency 2019). In addition to using diesel generators as a temporary solution to compensate for the reduction of electricity as they use diesel as a fuel and have different sizes and capacities. These generators are mainly used to power small-scale areas, such as in homes, shops, and offices; for a short period. These generators are reliable and simple to use but have some problems, such as loud noise and high cost.

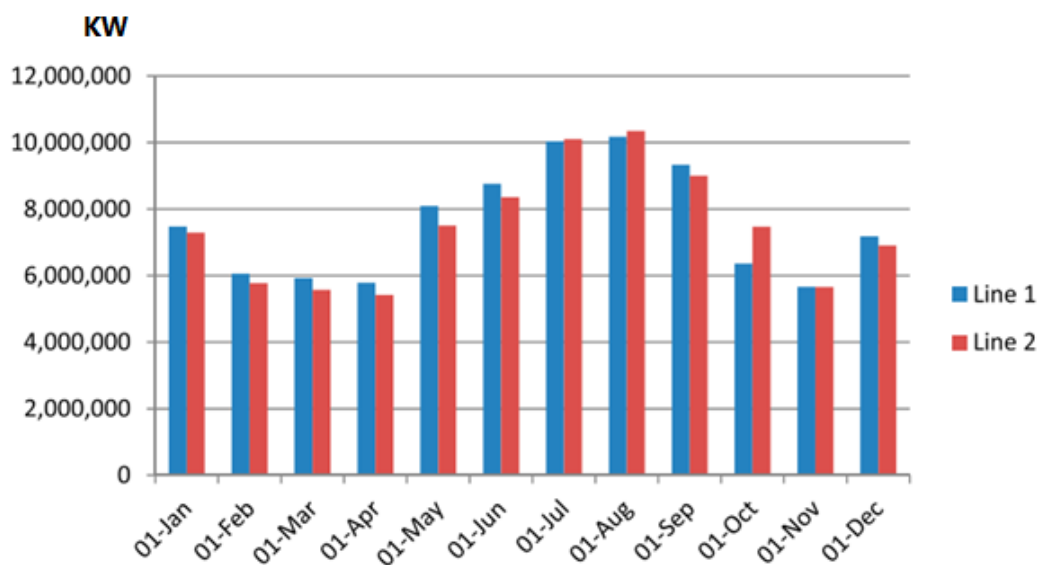


Figure 2.3 Monthly electricity consumption in kWh for Tulkarm through lines 1 and 2 in 2018

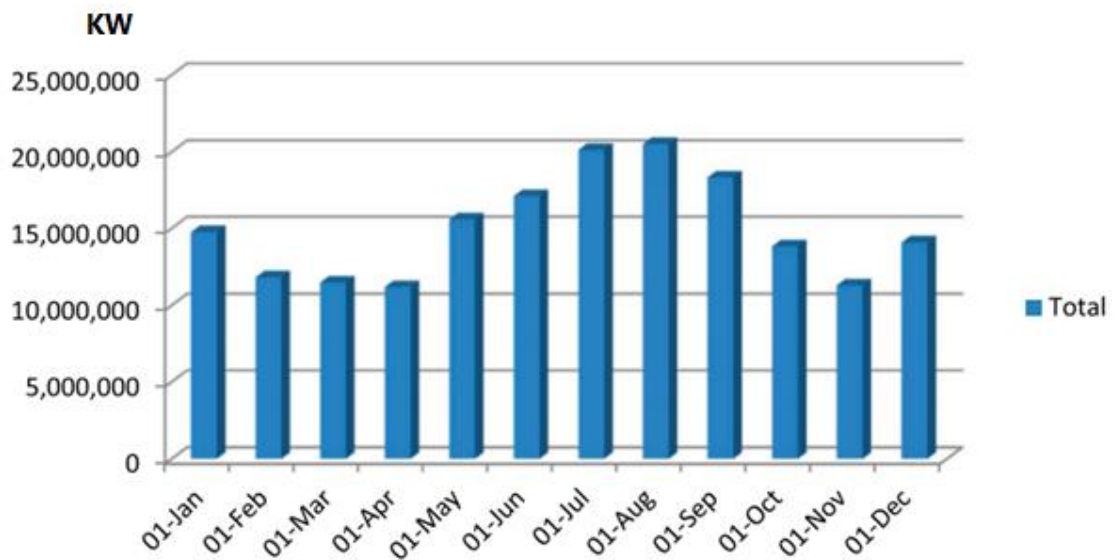


Figure 2.4 Total monthly electricity consumption in kWh through lines 1 and 2 in 2018

Our data is divided into three sections: electricity consumption from 2 main lines feeding the city, intensity of electric current for every 30 minutes from the two lines, individuals electricity consumption for the last three years (2018-2020).

This data is represented in the form of daily, monthly, yearly readings, Also, the forecasted time-series that are based on the collected datasets can be in different forms as (one hour, summer and winter seasons, one year and three years) form. Thus, the used datasets for both electrical lines were in the summer and winter seasons, three years combined to predict the next summer and winter electricity consumption and the next year's electricity consumption. Moreover, we use clustering and classification methods to select the most electricity consumption regions, dividing the tested region into areas depending on more electricity need, using special kinds of clustering algorithms.

2.3 Literature Review

Many papers studied electrical energy production in Palestine, including (Ahmed S A Badawi, et al. 2019) in a study that estimated the energy production along with payback period based on the wind speed records in the coastal Ashdod city. Where the average of the wind speed data was taken daily, then analyzed and fitted to the Weibull probability distribution function and using the graphical method to evaluate the Weibull value. Furthermore, the payback period is calculated with the cost of installing and annual return in addition to maintenance and operating fees. The experiment was conducted for a small-scale project using a wind turbine generator of 5 kW that can operate in different weather conditions from light wind too strong storms, thus it is economically feasible to invest in the energy generated using wind turbines.

(Nassar F, et al. 2019) evaluated the potential solar energy for the Gaza strip using Solar radiation data for 15 years from 2000 to 2015 as hourly time-series for five cities in the Gaza Strip using the System Advisor Model (SAM) from National Renewable Energy Laboratory (NREL) software. Energy production was discussed in terms of Photovoltaic (PV) and concentrating solar systems (CS). The aim was to provide a possible solution to the electricity crisis in the Gaza strip, as there is not a solid infrastructure for generating and transmitting electricity, in addition to the scarcity in the availability of the typical sources of energy especially electricity and petroleum sources that leads to a total reliance on external energy imports from both Israeli companies and Egyptian suppliers and some Palestinian companies. The study discussed two potential solutions to the energy crises: the first states to establish 5PV solar power plants in the five big cities of the Gaza Strip: Jabalia, Gaza, Deir-Albalah, Khan-Yunis, and Rafah that represent the entire stripe to provide a part of the electricity (552 GWh/year). While the second “independence project” imposed providing Gaza Strip with the demand electricity (714 from IEC and 61 from Egypt, and the 552 demand GWh/year). SAM software was used to simulate and analyze many utility-scale solar energy systems rapidly by

uploading the climate changes and electrical loads files to the program with different alterations concerning the economic parameters, the results appeared in a few seconds.

On another note, each country uses different prediction techniques, as in (Tang L, et al. 2018) the prediction method used is fuzzy Bayesian theory and expert prediction (EP) that is based on a long-term probability forecasting model to predict the Chinese per-capita electricity consumption (PEC) over the period 2010-2030 by econometric methodology. The study compared the adapted approach with six other forecasting approaches, i.e. ARMA, ANN, Logistic Regression (ES), Evolution Strategy (ES), Gray Model (GM), and Expert Prediction (EP). Furthermore, the forecasting results of 4 international agencies were also analyzed. The results showed that adapted prediction methods outperformed the other approaches in flexibility, reliability, and accuracy.

(Vu N, et al. 2017) performed an econometric model with Cobb Douglas production function to predict electricity consumption for Vietnam electricity system (GWH) up to 2030. This approach combines economic theory and statistical models. Five parameters were chosen: Gross Domestic Product (GDP), per capita earnings, population growth rate, industry share of electricity, number of households. The results demonstrated that only three parameters had a great impact on the prediction results that did not include GDP and the industry share, which was very surprising. Moreover, it was predicted that the electricity consumption in 2030 will be more than two times that in 2020 and more than three times as that in 2016.

(Hamzaçebi, C. 2019) studied Turkey's electricity consumption prediction between 2015 and 2018 to identify how season changes affect electricity demand, also four different customized ANNs models were compared to find the most reliable and accurate model. Then this model was compared with a simple time series model (SARIMA). The paper also made a review of all the research concerning electricity prediction in Turkey and other parts of the world.

According to (Amber et al. 2018) a comparison between five different ANN's techniques was conducted to evaluate the prediction capability for electricity consumption in London. These five mechanisms are Artificial Neural Network (ANNs), Genetic Programming (GP), Multiple Regression (MR), Deep Neural Network (DNN), and Support Vector Machine (SVM). Finally, the results demonstrated that ANN's techniques outperform all the other techniques and proved to be one of the most intelligent and widely used technologies for electricity consumption prediction.

Moreover, there are many studies concerning electricity consumption, (Ozoh, P et al. 2014) studied the annual electrical energy consumption of the Malaysia Sarawak university using ANN along with a modified Newton's model (MNM). (Bianco et al. 2014) forecasted annual electricity consumption in Italy by linear regression models, same as (Azadeh et al. 2010) who predicted monthly electricity consumption using fuzzy regression models. (Dedinec et al. 2016) used deep belief networks (DBN) to forecast short-term (hourly and daily) electricity consumption for Macedonia using historical data for the period 2008 to 2014.

Other different parameters that affect electricity forecasting, such as weather conditions, seasons, and previous data. In order to decide whether saving energy for future consumption is possible to meet an individual's needs. Thus, many methods were adopted by different researchers to find the best method for electricity prediction, such as multiple linear regression, and artificial neural network methods including; multilayer perceptron, radial-base function, deep learning, recurrent neural network, etc. (Amber et al. 2018).

Regarding using recurrent neural networks (RNN), (Rahman et al.2018) conducted a study for the Public Safety Building in Utah to predict hourly electricity consumption, and as well as for residential buildings in Texas for aggregated hourly electricity consumption. Recurrent neural

network (RNN) sequence-to-sequence models were compared with the Multi-Layered Perceptron Neural Network for forecasting. It was found that the first method results in a lower relative error in the Public Safety Building, while in the residential buildings there is a decreasing accuracy.

Moreover, (Hochreiter and Schmidhuber 1997) highlighted the problem with RNNs, which is the vanishing gradient that makes it difficult to learn long-term dependencies. Thus they suggested a solution that models both short and long-term temporal dependencies called recurrent neural networks with long short-term memory (LSTM). Another example study by (Hamzacebi et al. 2017) forecasting the monthly electricity demand of Turkey to model the effects of changing seasons. Four different models of ANN were compared with each other and also with SARIMA in terms of acceptability, reliability, and performance measures to determine the stable energy policies and medium-term.

(Khwaja et al. 2017) used an ensemble learning algorithm called boosted neural network model that trains a set of artificial neural networks iteratively, in order to enhance the short-term electrical consumption forecasting. They confirmed that this model reduces the variations in the forecasting errors and reduces computational time compared with simple ANN. In addition, (Salahat 2017) conducted similar research using different ANN techniques to predict future electricity consumption based on previous data. They used Multilayer Feed-Forward with Back-propagation Neural Networks (MFFNNBP) to make predictions.

(A. Azadeh, et al. 2008) focused on forecasting the annual electricity consumption in energy intensive manufacturing industries. ANNs have been used to forecast short-term consumption, but in this study, they showed that they can be used for long-term forecasting, such as annual forecasting, despite the strong changes of demand in high energy consumption industries, which makes forecasting a cumbersome and vague task. Actual data from high energy-

consuming (intensive) industries in Iran from 1979 to 2003 is used to illustrate the applicability of the ANN approach. This study shows the advantage of the ANN approach through analysis of variance (ANOVA).

(Protasiewicz, et al .2020) presented a study in the field of energy informatics specifically on machine learning (ML) models, and how they would be implemented and applied for electricity consumption forecasting. Four neural networks (NNs) were proposed for computational experiments and implemented using PyTorch, an open-source ML framework, as follows: long short-term memory network (LSTM), fully recurrent neural network (fRNN), backpropagation neural network (BPNN), and gated recurrent unit (GRU). After making comparisons between the four neural networks based on their predictive performance, the results demonstrated that LSTM and GRU performed better and had less error than the other two models. Where GRU performed best on daily forecast and LSTM outperformed the monthly forecast.

As for (Salami et al. 2018) they aimed to search in big databases available in the electricity market using simulation-optimization algorithms, which was combined with the Wavelet-Neural Networks-Particle Swarm Optimization (Wavelet-NNPSO) method to develop a new technique of Wavelet-Neural Networks-Particle Swarm Optimization-Simulation-Optimization (WT-NNPSO-SO) in order to improve the accuracy of short-term forecasting of electricity supply and demand. The results showed that using this technique improves the runtime with big data in comparison with previous techniques, due to the use of meta-heuristic algorithms. Forecasting models are divided into traditional models and machine learning models (e.g. neural networks). Traditional models, including linear regression, Stochastic Analysis, Productivity Ratios, and Time Series Analysis are limited when conducted to the nonlinear systems. In contrast, the machine learning models can deal with complex systems (e.g. nonlinear systems) by deeply understanding the rationality, risk, and uncertainty of the data without the need for a

certain structure. Thus, the ANN's models give better results compared with traditional models, especially in our study.

2.4 Summary of Related Works

In this chapter we have reviewed different studies about electricity prediction, where they adopted different approaches with different variables and parameters, some applied the ANNs model for prediction as shown in table 2.1. However, none of them applied a hybrid approach including Evolutionary Algorithms. This research has the advantage of applying a new evolved model for approaching the electricity consumption prediction by using three hybrid models (RNN-OA, NARX-OA, NARX-OA with K-mean Clustering) to predict future electricity consumption, based on the RMSE value to find the best model.

Table 2.1 Some of the Electricity Consumption Forecasting Studies in the World

Reference	Method	Variables	Forecasting for	Accuracy rate
Tang L, et al [2018]	Fuzzy Bayesian theory and expert prediction	Three components: fuzzy relation matrix, prior prediction, and fuzzy Bayesian formula	Annual total electricity consumption for Chinese per-capita electricity consumption (PEC) and its variation interval over the period 2010-2030	MAPE = 2.35%
Vu N, et al [2017]	Econometric model with the Cobb Douglas production function	Five variables proposed in the forecasting function are GDP, income, population, the proportion of industry and service in GDP, and the number of households.	Annual total electricity consumption for Vietnam electricity system (GWH) to 2030	None
Hamzaçebi,C [2019]	ANN and SARIMA	Monthly electricity consumption values of Turkey were collected from the Turkish Electricity Transmission Company (TEIAS).	Seasonal and monthly electricity demand of Turkey has been predicted between 2015 and 2018	MAPE= 2.28% For ANN

Amber et al. [2018]	Multiple Regression (MR), Genetic Programming (GP), Artificial Neural Network (ANN), Deep Neural Network (DNN), and Support Vector Machine (SVM)	Annual electricity consumption data	Forecasting annual electricity consumption of an administration building located in London	MAPE = 6% For ANN
Ozoh, P et al. [2014]	time-series technique (FTST) and ANN with the modified Newton's model (MNM)	Annual electricity consumption data	Electricity consumption for the University of Malaysia Sarawak from 2009 to 2012	MAPE= 2.1% For MNM
Bianco et al. [2014]	Linear regression models	Annual historical electricity consumption, gross domestic product (GDP), gross domestic product per capita (GDP per capita), and population data	Annual total electricity consumption in Italy in the period 1970-2007	None
Azadeh et al. [2010]	Fuzzy regression model	Monthly and seasonal electricity consumption data	Monthly electricity consumption of Iran from 1994 to 2005	MAPE= 0.0082%
Dedinec et al. [2016]	deep belief networks (DBN) made up from multiple layers of restricted Boltzmann machines	Hourly electricity and temperature data	Macedonian system operator (MEPSO) hourly electricity consumption data in the period 2008-2014	MAPE= 3.68%

Chapter 3

Methodology

3. The Proposed Method

3.1 Introduction

This research can be considered both quantitative and qualitative research. It is quantitative research because it is based on the measurement of quantity “electricity consumption “, thus the result will be a set of numbers presented in tables and graphs. On the other hand, it is qualitative research because ANN's techniques that support our proposed model as a dynamic system can summarize the whole information about the past behavior of the system through a set of values that are significant to get a unique description of the future behavior of the system (Ugurlu et al. 2018).

Today, researchers focus on forecasting the customers' usage of electrical power using different forecasting methods some of the most used methods are autoregressive integrated moving average (ARIMA), multiple linear regression (MLP), time series method, and artificial neural networks (ANNs), also the pattern of electrical power usage can be found out using a dataset prepared with real data.

In this chapter, we use artificial neural networks (ANNs) for learning and training from an external environment. We use ANNs built-in models (dynamic networks) in ‘MATLAB R2019a’, which is considered the most used network for time series forecasting. We simulated our data which is represented as a time series for electricity consumption through two wizards. First, with two layers feed-forward neural network. Second, (dynamic time series) is classified to solve three kinds of time series problems, but we are concerned with two kinds specifically; Standard Recurrent Neural Network (RNNs) and Nonlinear Autoregressive with External (Exogenous) input (NARX). Neural networks (NNs) consist of several neurons which are a collection of single processing elements connected in a way that adjusts the weights of input for each layer to reach the desired output through forwarding and backward phases.

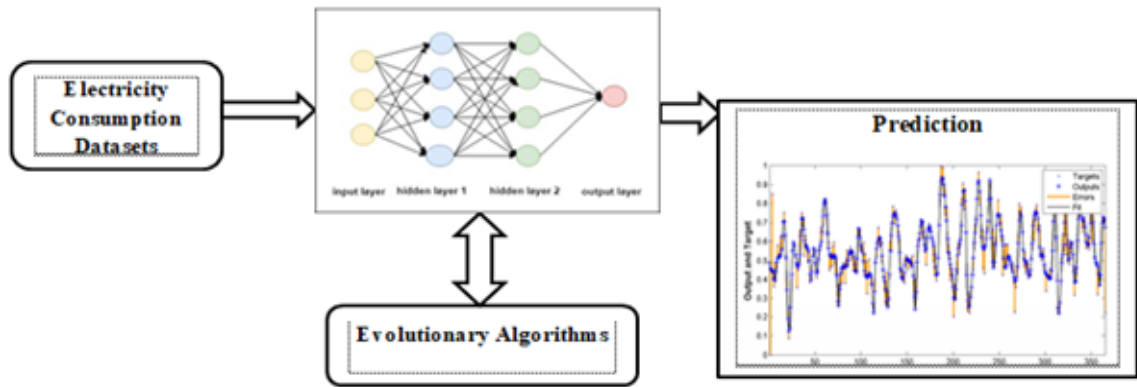


Figure 3.1. Primary structure of the proposed hybrid models

The above figure 3.1 represents the primary structure of the proposed model, we use the neural networks (NNs) technique which is the most used machine learning (ML) technique for time series forecasting to train and test electricity consumption data samples. Then we combined Artificial neural networks (ANN's) with optimization approaches to predict daily, monthly, seasonal, and yearly electricity consumption with the best accuracy. We used one of the most popular neural networks like Multi-layer perceptron (MLP) neural networks. One of the Evolutionary algorithms (EAs) used to optimize the topology and the parameters of the Standard Recurrent Neural Network (RNNs), and Nonlinear Autoregressive Neural Network (NARX).

Furthermore, we used clustering methods to enhance EAs and select electricity consumption regions, dividing the tested region into areas depending on more electricity needs, using special kinds of clustering algorithms such as k-mean clustering.

The network output is the prediction period for the next time series. The main goal of these experiments is to find a model with a closer result to real electricity consumption data. These findings, which will be expressed as Root Mean Square Error (RMSE) as shown in formula 3.1, is a good measure of accuracy.

Root Mean Square Error:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{a=1}^n (T-Y)^2} \quad 3.1$$

Where n is the total number of input data, T is the Target values, and Y is the predicted value.

the error between output and target is calculated as the following formula:

$$e = \sum (T - Y) \quad 3.2$$

3.2 Dataset Preparation Proces

Every individual or company that uses electricity service (or the municipality) is considered the population of our research. Electricity consumption is measured periodically by the local municipality in Tulkarm, so we focused on the measured data from 2018-2020. Data gathered from the two main electricity lines was very large, so this data was pre-processed to improve the performance of ANN's and to increase the prediction accuracy of our proposed model by reducing the noise in the data input. The mean was calculated in order to make predictions for the next year. We performed a processing operation using a smoothing function in Matlab that measured the average and similarity between all neighboring data points to be replaced with input data according to equation 3.3. In case of electricity breakdown, the missing data point is replaced with the mean of its neighbor's data points.

For each (i) from 1 to (n).

New missing sample value (the average)

$$\text{Average value: } avg = \frac{1}{2} (\sum_{i=0}^n pi + \sum_{i=0}^m Ni) \quad 3.3$$

Where n and m are the numbers of previous and next values of irregular value, Pi and Ni are the ith previous and next values.

The next step after the smoothing process is the dataset normalization which is used to restrict the values of data samples between 0 and 1. Also, it's a process of organizing data and reducing diversity between the samples by eliminating data anomalies, redundancy, and uncertainty. This data was normalized using [a,b] scale according to the formula (3.4) (Chung, H., & Shin, K. S. 2018).

$$\text{Normalized Data:} \quad d_n = \frac{(d - \min(d))}{(\max(d) - \min(d))} \quad 3.4$$

Where d is the original value and, d_n is the normalized value, and $\max(d)$ and $\min(d)$ are the maximum and minimum values of the original dataset.

3.3 Applied Models

This section presents different hybrid neural networks NNs methods which are (Standard Recurrent Neural Network (RNNs), and Nonlinear Autoregressive Neural Network (NARX)) based on the nervous system and evolution theory to acquire electricity demand close to real consumption. The development of NNs model depends on the EAs techniques to discover the main characteristics of the time series to predict their behavior in the future. These techniques are applied to predict electricity consumption in Tulkarm city depending on the past and current consumption.

In our proposed practical solution, we implemented electricity consumption datasets on hybrid ANNs techniques. In detail, hybrid models (RNNs-OA, NARX-OA, NARX-OA-K-Mean Clustering) that are RNNs and NARX with enhancement evolution algorithms. Furthermore, initialization and crossover steps were altered by using clustering techniques. On the other hand, there were no previous studies applied on the general NN model, nor any particular hybrid model of (NNs and OAs) exist for electricity consumption prediction of Tulkarm city. Hence,

it was necessary to establish an NNs model with the ability to deal with nonlinear data to forecast electricity usage and formulate a suite of strategies for the work.

3.3.1 Genetic Algorithms

Genetic algorithms (GAs) are a computational search approach used in artificial intelligence, specifically the search (Meshkinfam and Ghassemi Tari 2016, Dong et al. 2018). In the 1970s, GAs were introduced based on Darwin theory by John Holland at Michigan University. GAs are a type of evaluation strategy that depends on the optimal position to achieve high-quality results with the least cost. It is a parallel computation for solving problems versus mathematical and analytical traditional methods that get stuck in local optimum solutions (Elif İpek Uysa, et al.2019). It is also considered an explanation and mimicry of genetic evaluation for all living things which is the core concept for surviving for the fittest as all individuals compete with others for survival and reproduction where the winner (the fittest) who has children, possess the ability to inherit their genes to the individuals of the next generation (Murthy, et al.2012). Genetic algorithms are divided into three operators, which are selection, crossover, and mutation. Starting from the selection operator where the best individuals are chosen from the initial generation, as the individuals are randomly generated or using any other method to form the population, thus these individuals are considered the solution to the proposed problem. The crossover takes the best individuals to produce children or the next generation. The mutation operator improves certain qualities in a particular individual. For each problem, the solution's (individual) quality is called fitness, which sets adequate criteria to form the solution to the proposed problem. If the fitness of a certain individual is high, it is most likely that it will inherit some or all of its genotype to the next generation. (Rinchon, et al.2017).

The principle of improving genetic algorithms is based upon evolutionary approaches, as the proposed space of solutions is represented in the chromosomes to find the best one and optimize

solutions to problems. (Hassanat, et al.2018) discussed the leverage of genetic algorithms using some seeding techniques like nearest neighbor (NN), random, and proposed linear regression to improve GAs population initialization for solving transfer salesman problem (TSP) under the same conditions as the experiment. The error rate of the proposed linear regression approach is the least as it is the best of the other techniques (9.2% NN, 18.6% random). This results in creating individuals of GAs that are sure to be the fittest. The resulting difference is due to dividing the problem into sub-problems and then applying linear regression to each of them. The regression-based method achieved 98.9% of average convergence than other seeding mechanisms and it also gave better results especially with big problems.

3.3.2 K-means Clustering

The clustering approach aims to partition related data with similar features into groups that are already determined (K clusters), it's not a classification or regression technique. Clustering is a procedure that can be useful for classifying data and it is widely used in different fields, including data mining, business, geology, biology, healthcare, image processing, analysis and forecasting methods, etc. For example, in biology, clustering is used to partition the animal and plant taxonomy and label genes into a set of disjoint groups where each group contains genes with similar features and functions (Naeem, S. and Wumaier, A., 2018).

K-mean clustering is an unsupervised mechanism and a powerful tool advantageous for analyzing data, prediction models, and efficiently extracting related information based on distance then categorize it into disjoint clusters. It was found that there is a high similarity within the same cluster, while there is a high dissimilarity between different clusters (Tari, F.G. and Hashemi, Z., 2018).

K-mean technique is used to discover existing objects and cluster datasets based on mathematical calculation for forecasting analysis. It works as a random selection of central

points which are called centroids (K) based on the Euclidean distance formula. Then Euclidean distance was applied for items(x_i) to compute the mean for each cluster, in order to categorize and assign each object to the nearest centroid (Bansal, A., et al.2017).

3.3.3 Recurrent Neural Networks – OA (RNNs-OA) Model

Forecasting the future consumption should be an accurate process close enough to the actual output in order to accept the predicted model. This model depends on the previous data periodically collected, where the data input $x(t)$ is represented according to the time as a sequential series using known methods such as numerical schema, statistical procedures, theoretical approaches, and time series prediction. This may lead to high bias and high Sum Square Error (SSE) (Qasrawi and Awad et al.2015), thus the time series model with neural network (NN) called Recurrent neural networks (RNNs) makes decisions based on the current input with the previously received input (Dyer et al. 2016).

RNNs architecture is a standard feed-forward multilayer perceptron neural network with at least one feedback connection at the hidden layer with a specific delay in time. Besides that, RNN techniques are based on short-term memory (STM) which means that updating a new state depends only on the previous neuron state (Chung, H., et al.2018) (Dyer et al. 2016).

Learning in RNN performs like MLPNNs in the feed-forward phase. Moreover, the RNNs approach is built to overcome the shortage of ANNs with back-propagation through time plus a loop in hidden units (Qin et al. 2017). The significant feature that distinguishes (RNN) from (MLPNNs) is that it has feedback connection in the sigmoid hidden units, with sigmoid activation function f_H . Thus, the output of any layer represents the input for the next layer passing through the input layer, hidden layer/s to the output layer (Biswas et al. 2016). The following formula describes how the new state can be calculated:

$$\text{General formula}(t) = f(h(t - 1), x(t)) \quad 3.5$$

Where $h(t)$ is the new state, $h(t-1)$ is the old state represented as input in the hidden units, $x(t)$ is the current input, and $f(h, x)$ is the activation function.

Regarding formula 3.5, the hidden layer $h(T)$ decision is affected by the past hidden layer decision $h(t-1)$. Thus there is a correlation between the current, next, and past state in the RNN technique like the thinking way of the human brain which depends on the present and past situation to acquire a decision (Hiransha, M., Gopalakrishnan, et al.2018).

The total cost function resulted from training the data according to time (t_0) to end time (t_1) is the sum squared error (SSE) over each time-step, calculated using this formula:

$$E_{total} = \sum_{t_0} E_{sse}(t) \quad 3.6$$

$$SSE(t) = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Where y_i is the predicted value and \hat{y}_i is the actual value.

Relying on the equations above, the gradient descent weights are updated for each time step.

The Levenberg Marquardt algorithm uses the Gaussians Newton approach and gradient descent approach to improve SSE (Biswas et al. 2016). as the following formula:

$$\Delta W_{ij} = -\frac{\mu \partial E_{total}}{\partial W_{ij}} = -\mu \sum_{t_0} \frac{\partial E_{sse}(t)}{\partial W_{ij}} \quad 3.7$$

Where μ is the learning rate. The derivative expression $\partial E_{sse}(t) / \partial W_{ij}$ depends on multiple weights which are $W_{ij} \in \{W_{IH}, W_{HH}\}$, input $x(t)$, and the hidden unit activation on all earlier time steps, where errors from time and network are back propagated.

Flowchart of Recurrent Neural Networks (RNN):

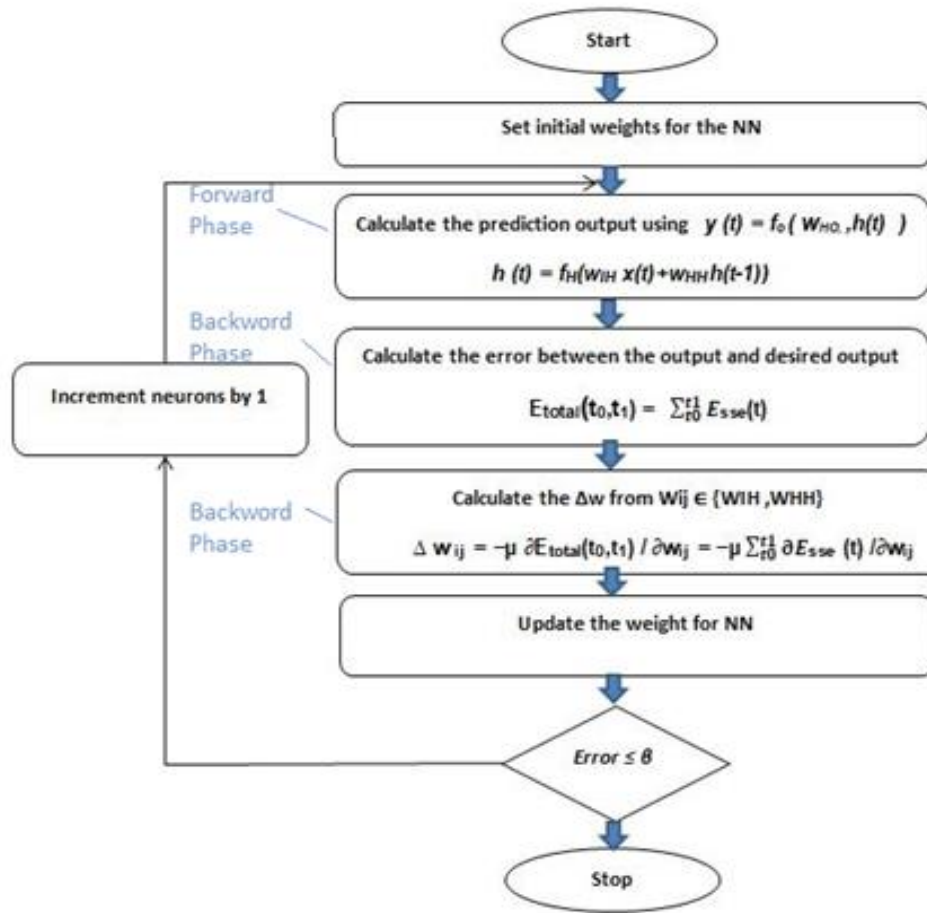


Figure 3.2 Flowchart of Recurrent Neural Networks

As illustrated in Figure 3.2, the flowchart of standard recurrent neural network models describes three phases: First, the forward phase, where the data and weights are passed from the input layer to the hidden layer then the resulting output goes back as feedback to the hidden layer. Second, the backward phase calculates the error rate between time steps through gradient back-propagation. Third, the backward phase modifies the weights.

The proposed RNN model (RNNs-OA) was customized along with making adjustments and combining it with optimization algorithms to get better results and less Mean Square Error (MSE) for predicting electricity consumption applied on the past real data gathered over two

years (2018-2020) from two main electricity lines. This data runs ten execution times starting from 2 neurons to 20 neurons based on the general structure of RNN as shown in figure 3.3.

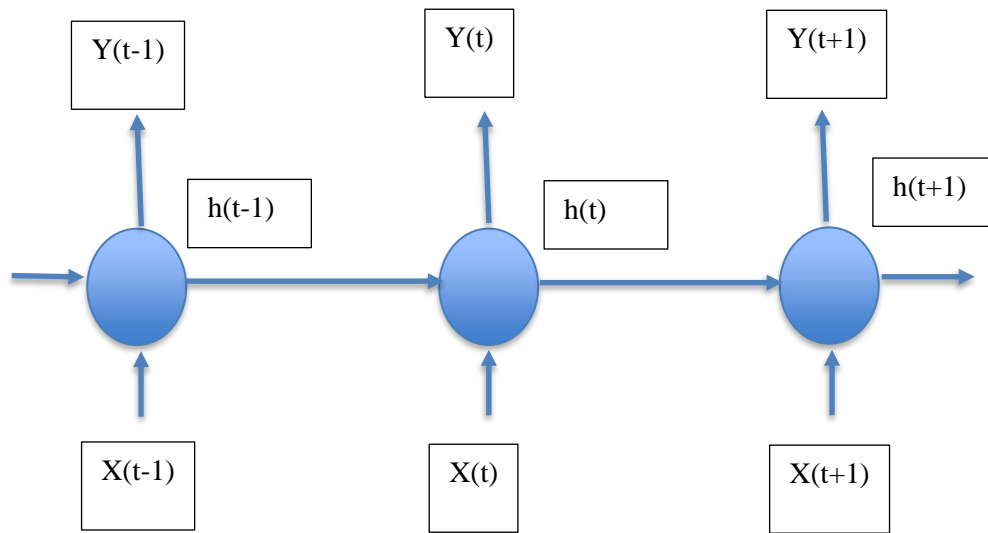


Figure 3.3 General structure of the RNNs

For our optimized RNN-OA model, we updated the weights of the NNs connections of our used neural networks based on the GAs as the optimization algorithm where any modification on the GAs parameters and the selection of fitness function can have an impact on the performance and the effectiveness of the algorithm. Figure 3.4 illustrates the details of the RNNs-OA hybrid model, where GAs is used to update RNNs weights, by following the model, this is achieved through RNN initialization and configuration, then initializing GAs and evaluating fitness function, after that creating a new population and evaluate it by applying the basic operations of the GAs to find the fittest individual of the population to set new weights for the neural network.

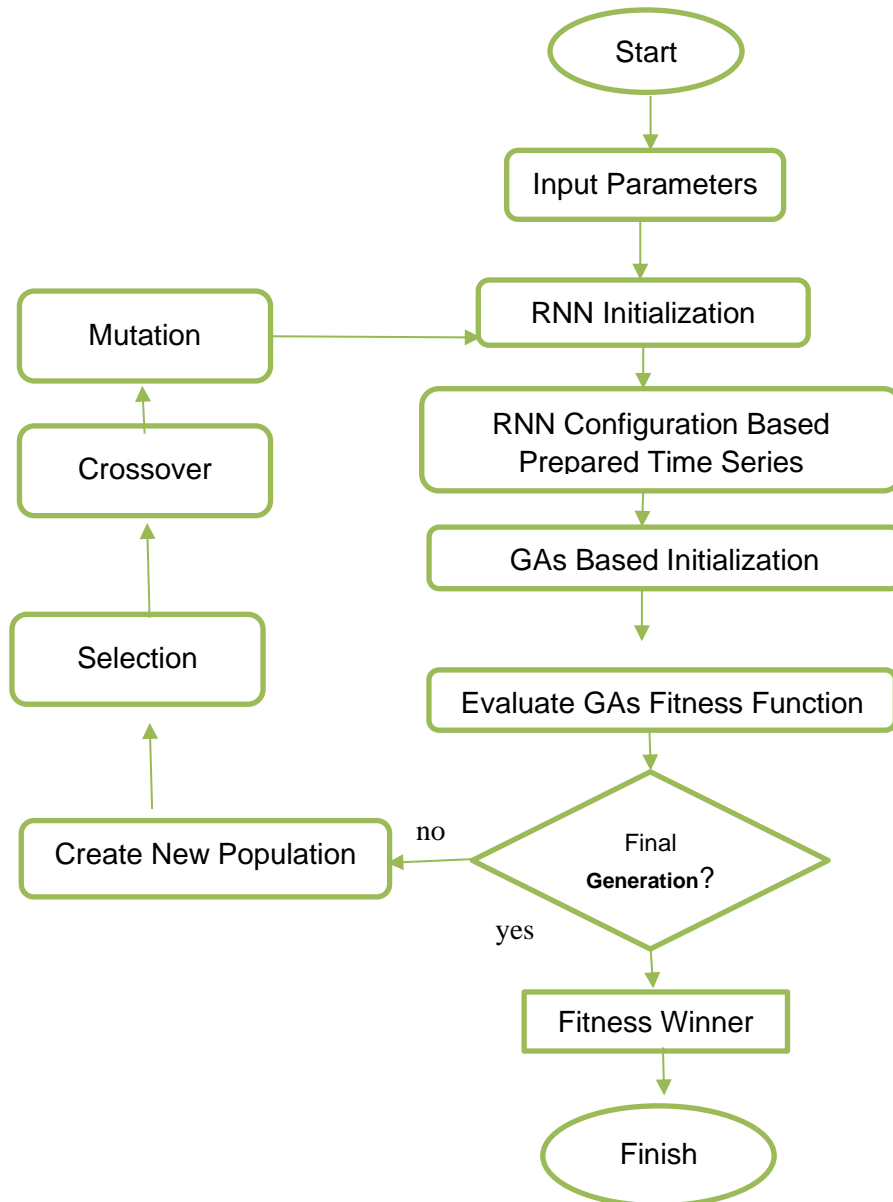


Figure 3.4 the RNNs-OA Hybrid Model

The RNNs-OA Pseudo Code:

Algorithm 1: The general procedure of RNNs-OA

D ← Load the electricity datasets (line 1 & 2 datasets)

Input: time series (monthly, seasonal, yearly electricity consumption), number of hidden layer neurons, choose a Training Function, and layer delay.

Output: Prediction electricity consumption and MSE, RMSE matrices
 Read $\leftarrow D$
Data Preprocessing:
 Preparation $\leftarrow D$
 Normalize $\leftarrow D$
 Separate $D \leftarrow$ to train and test data
 For index in the length of $D \leftarrow$ do
 Partition Normalized $D \leftarrow$ to X-train and X-test
 End loop
 Start
Build and Executing RNN model
 $D \leftarrow$ input dataset
 $W \leftarrow$ weights
 $\Delta Wij \leftarrow$ updated weights as \leftarrow Eq.3.3
 Initialize initial hidden state to 0
 For each state
 Update new hidden state: $(D*W) + (\Delta Wij + \text{hidden state})$
 Start Optimized Approach (GAs)
 Set epoch =500
 Itr =0
 Create initial Population Randomly, $P(\text{Itr})$
 Evaluate Fitness for all Individual in Population, $f(P(\text{Itr}))$
 While Itr < epoch
 Itr = Itr +1
 Select parent pair from Population
 Select $P(\text{Itr})$ from $P(\text{Itr} -1)$
 Recombine the Population (with Probability P cross)
 Crossover $P(\text{Itr})$
 Conduct a Population Mutation (with Probability P mut)
 Mutation $P(\text{Itr})$
 Evaluate $f(P(\text{Itr}))$
 .33Compute Performance evaluation, MSE and RMSE Matrices
 Make Prediction

3.3.4 Nonlinear Autoregressive Neural Networks– OA (NARX-OA)

NARX is a dynamic nonlinear recurrent network architecture, it's a powerful technique used for solving time series applications because of its ability to use true historical value prediction and building forecasting models (Boussaada Z, et al 2018). Furthermore, NARX proves its effectiveness in learning compared with traditional NN algorithms, as it is good in long-term memory which is essential for improved forecasting models (Wei, M., et al 2020).

NARX predictors are classified into two categories: open-loop (also called series-parallel architecture), where the future time series value is acquired from the past output values that

represent a dynamic memory, and the past input with the present input values that represent static memory. The second approach is a close loop (also called parallel architecture) with the forecasting values acquired from the historical output values and the past input with present input values (Banihabib, M.E., and Mousavi-Mirkalaei, P., et al 2019) (Boussaada, Z, et al 2018).

NARX model has some features like dealing with nonlinear time series models. It's considered a recurrent dynamic system with exogenous inputs. The exogenous inputs are the external variables from previous predetermined series applied on a specific series to predict the next value. This input data goes through three layers. The input layer with at least two inputs (current input, and the past true output if series-parallel NARX architecture is used, or the past predicted values if the parallel NARX architecture is used). The next layer is the hidden layer that can take the form of simple or complex architecture and the output layer that makes NARX a dynamic time-series predictor with its feed-forward connections.

These two improved types of architectures are trained by NARX to model a dynamic system as described in figure 3.3.1. And the equations are as follows:

For Series-Parallel Architecture:

$$y(t) = f (y(t - 1).y(t - 2). \dots .y(t - ny).u(t).u(t - 1).u(t - 2).0 \dots .u(t - nu)) \quad 3.8$$

Where $\{y(t-1), y(t-2) \dots, y(t-ny)\}$ is the past true output (time series), $\{u(t), u(t-1), u(t-2) \dots, u(t-nu)\}$ is the present and past inputs, nu is the number of input delays, ny is the number of output delays and f is the activation function (i.e. tanh).

For Parallel Architecture:

$$\check{y}(t) = f(\check{y}(t-1), \check{y}(t-2), \dots, \check{y}(t-ny), u(t), u(t-1), u(t-2) \dots, u(t-nu)) \quad 3.9$$

Where $\{ \hat{y}(t-1), \hat{y}(t-2), \dots, \hat{y}(t-n_y) \}$ is the past predicted output (time series), $\{u(t), u(t-1), u(t-2), \dots, u(t-n_u)\}$ are the present and past inputs, n_u is the number of input delays, n_y is the number of output delays, and f is the activation function (i.e. tanh).

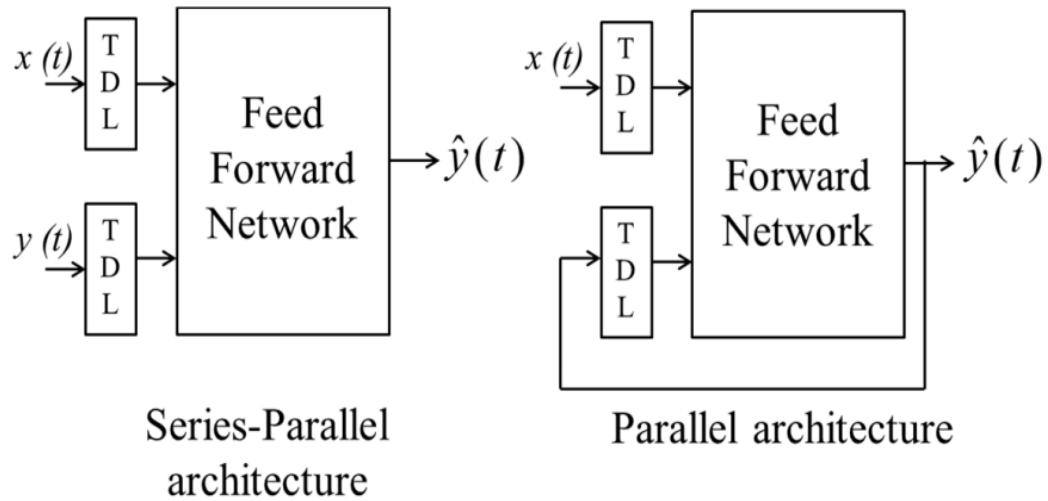


Figure 3.5 Types of training architecture for NARX neural networks

The general equation of NARX model is as follows:

$$y(k) = f(y(k-1), y(k-2), \dots, y(k-n_a), u(k-1), u(k-2), \dots, u(k-n_b)) + e(k) \quad 3.10$$

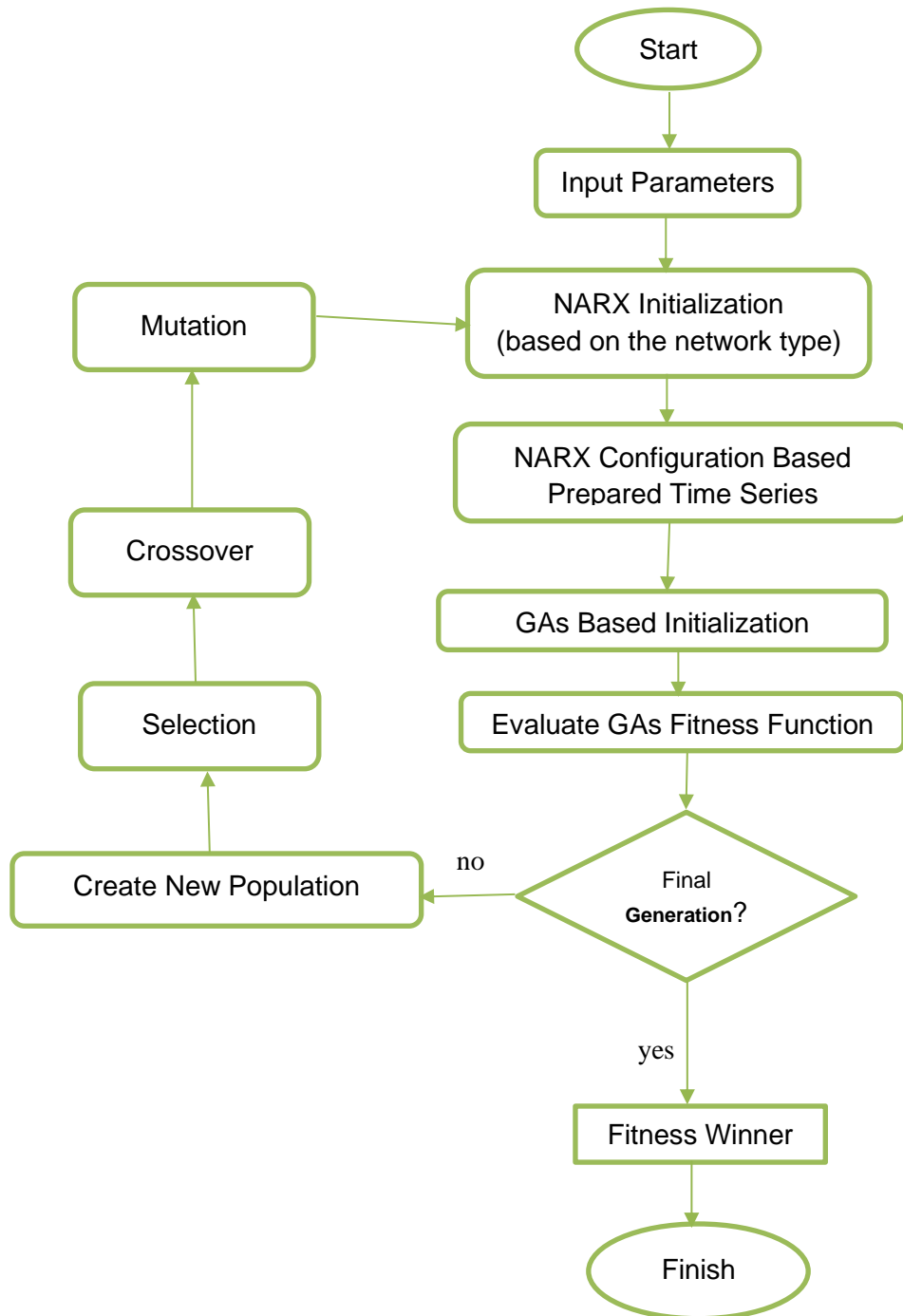


Figure 3.6 NARX-OA Hybrid Model

In the NARX-OA model, we also used the GAs for optimizing the NARX network, which's similar to what we did with the RNNs-OA model. It also includes some customizations

regarding the used NARX network architecture in the initialization phase. Which could be an open or closed loop.

NARX-OA Pseudo Code:

Algorithm 1: The general procedure that was used in electricity consumption prediction

D ← Load the electricity datasets (line 1 & 2 datasets)

Input: time series (monthly, seasonal, monthly electricity consumption), number of hidden layer neurons, choose a Training Function, input Delays, feedback Delays

Output: Prediction electricity consumption and MSE, RMSE matrices

Read ← D

Data Preprocessing:

Preparation ← D

Normalize ← D

Separate D ← to train and test data

For index in the length of D ← do

Partition Normalized D ← to X-train and X-test

End loop

Start

Build and Executing RNN model

D ← input dataset

W ← weights

ΔW_{ij} ← updated weights as ← Eq.3.3

Initialize initial hidden state to 0

For each state

Update new hidden state: $(D*W) + (\Delta W_{ij} + \text{hidden state})$

Start Optimized Approach (GAs)

Set epoch =500

Itr =0

Create initial Population Randomly, P(Itr)

Evaluate Fitness for all Individual in Population, $f(P(Itr))$

While Itr < epoch

Itr = Itr +1

Select parent pair from Population

Select P(Itr) from P (Itr -1)

Recombine the Population (with Probability P cross)

Crossover P(Itr)

Conduct a Population Mutation (with Probability P mut)

Mutation P(Itr)

Evaluate $f(P(Itr))$

Calculate Performance Evaluation Matrices

Make Prediction

3.4 Optimized Approach

In this section, we illustrate how to acquire good initial weights for a neural network. Referring to acceptable initial weights, neural networks achieve a new state with higher accuracy and

optimal solution than the standard algorithm that depends on backpropagation methods like the Levenberg-Marquardt algorithm (LMA) (Puig-Arnavat, M., & Bruno, J. C., et al 2015). The pre-trained network applies an optimization technique implementing a Genetic Algorithm (GA) based search and transfers learning to fine-tune the network; it's a stochastic approach where initialization and selected population seeding phase are generated randomly.

Genetic Algorithms are one of the evolutionary mechanisms that not only manipulate a potential solution but keep a pool of solution candidates with search space to solve optimization problems that cannot be generally tackled using other analytical approaches, its meta-heuristic search. GAs use a genetic approach, like natural selection and evolutionary ideas of genetics, which a lot of biologists find very spectacular. It's a very interesting field where, unlike the linear approaches, it's rather like genes where evolutionary ideas of inheritance are primal ingredients in this approach that makes it unique (+Meena, D.C. and Devanshu, A., 2017).

In this work, our contribution includes GAs as an enhanced algorithm with a new approach for producing specific initial population seeding techniques and optimizing crossover operators. We combined the Genetic Algorithm with the neural network for updating weights connection for our used neural network. Therefore, using cluster methods like K-mean clustering for the population phase increases the efficiency of GAs in order to find the optimal solution. The following figure 3.7 illustrates our hybrid model.

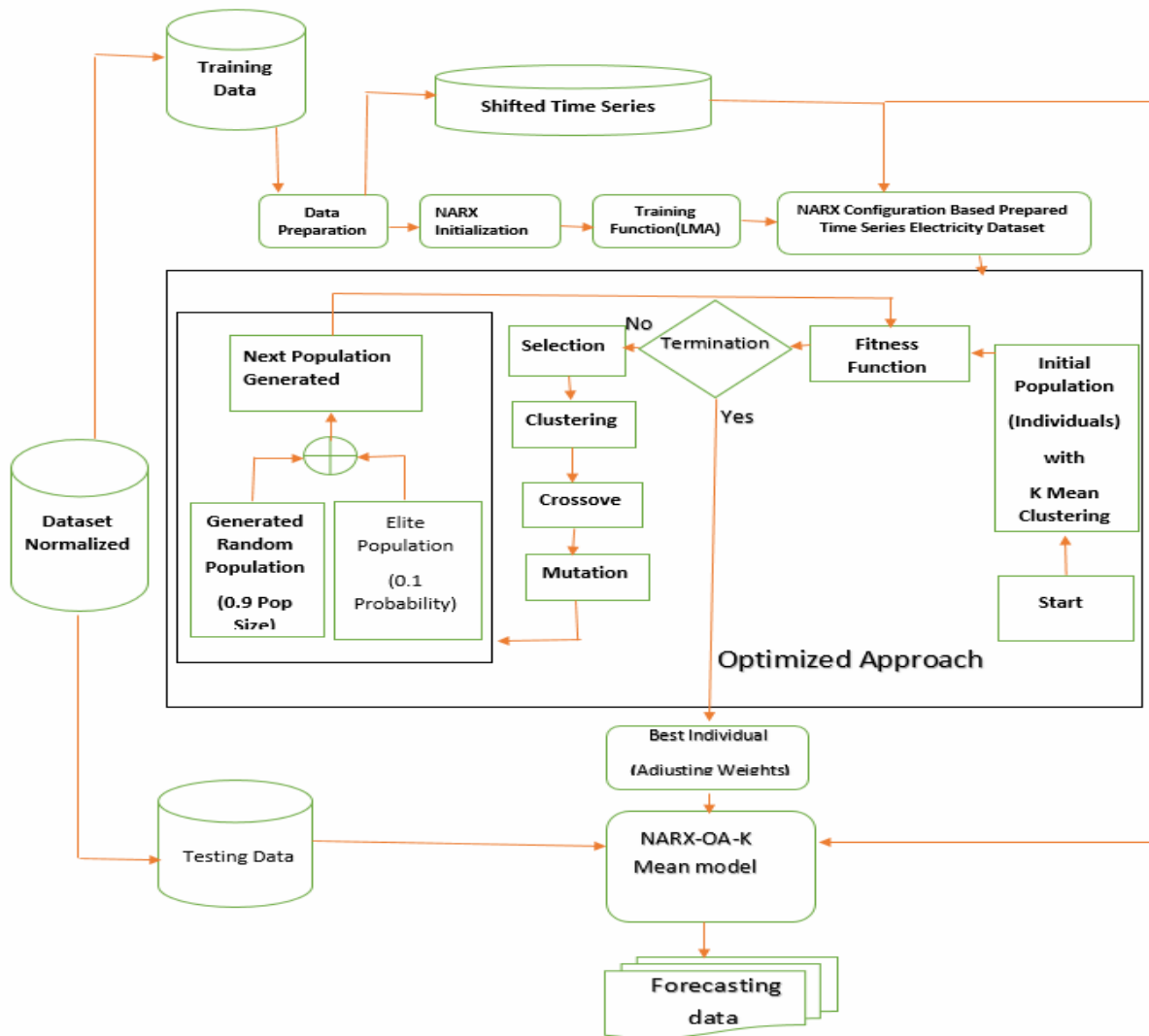


Figure 3.7 Recurrence Model with Enhanced Strategy (Hybrid Model)

Genetic Algorithms have been affected by our proposed model through some parameters such as population size, number of generations, crossover probability, mutation probability, and the initial population seeding which is considered to have a manifest role in gaining the optimal or near-optimal solution.

The GAs process used for updating connection weights of NNs are:

1. We proposed a new approach for extracting the initial population from pre-trained neural networks based on clustering (K-mean clustering).
2. Tuning population size parameter as input which represents the length of the chromosome.
3. Tuning input number of evaluation or generation parameter.
4. Modifying crossover operator by clustering parents.
5. The fundamentals of GAs are selection, crossover, and mutation.
6. Reproduction of new offspring and evaluate it.
7. Used the fittest chromosome which represents new weights for the neural network.

there are three main fundamentals of the GAs structure known as reproduction, crossover, and mutation.

This will be illustrated in detail as follows:

1. Initial Population

The first step of any Genetic Algorithms (GAs) application is to determine the population size which is considered the most significant phase for enhancing the performance. The random initial population seeding approach is executed only once differently from other GAs phases; it's the most common and appropriate process used (Hassanat, et al. 2018). In our work, the initial populations which are (individuals/ chromosomes) initiated by random natural distribution (randn) optimized with a K-mean clustering approach. Chromosomes, potential solutions, involve a set of genes that are represented by a sequence of weights. The population size is the length of the chromosomes involving weights NNs, distinct by trial and error. The following figure 3.8 describes the chromosome's structure.

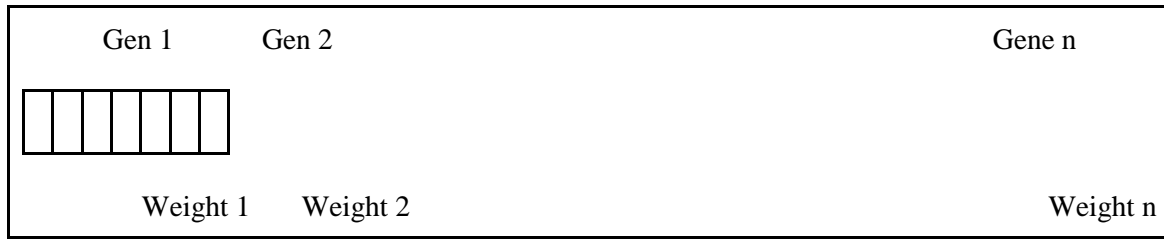


Figure 3.8 Basic Structure of Chromosome

Where n is the number of genes or weights.

While the (n) can be calculated as:

If assumed that the NNs contain (x) input, (h) hidden, and (s) output neurons, then the chromosome must contain “ $(x+s) * h$ ” genes.

$$\text{Number of weights: } n = (x+s) * h \quad 3.11$$

- The workflow of cluster algorithm

Clustering is an unsupervised learning method, it's a greedy iterative technique specified by randomly chosen centroids. The clustering algorithm is a subset of data mining techniques that we used for its efficiency and simplicity (Xu, H., et al 2018). In this work, we initialized the individuals using the K-mean cluster algorithm which depends on how to find the optimal number of K clusters.

Steps for the process that is implemented with the K-mean clustering are:

- I. Declare the number of centroids randomly $c_1, c_2 \dots c_n$.
- II. Find the mean between ith data point, p_j from the cluster center, c_i and the mean of ith points n between clusters points using Euclidean Distance (d_{ji}) expressed as the following formula:

$$\text{Euclidean Distance} = |p_j - c_i|^2, 1 \leq j \leq n. \text{ and } 1 \leq i \leq k. \quad 3.12$$

There are several factors that could influence the performance of GAs: parameters, genetic operators and the main one is the initial population. Random initial population is considered a traditional method, however it is insufficient to produce the best initial population. On the other hand, K-mean clustering initial population approach divides the initial weights (genes) into chromosomes which allows for clustering them further repeatedly, in this way the coordinates of each chromosome determines which cluster it belongs to, the algorithm works repeatedly until the size of the chromosome becomes very small, four or less for instance, which are more likely to be adjacent to each other, so connecting them to each other creates a good solution then it is mutated several times to form the initial population as shown in figure 3.10 (Hassanat, et al. 2018).

Fitness Function

This genetic algorithm uses an adaptive function to evaluate the quality of the potential solutions. In our proposed approach, the fitness function which is applied for the individuals to determine its performance is Root Mean Square Error (RMSE). The fitness function must be chosen carefully due to the generated information from the fitness function that will be used to select the individuals from reproduction, mutation (offspring), and finally, the best solution chosen based on the fitness value (Lamini, C., et al 2018).

2. Selection Operator

There are several techniques used for the selection: Roulette Wheel Selection, Boltzmann selection, rank selection (linear and exponential), uniform selection, study state selection, etc. The selection operator is one of the main features in the Evolutionary Algorithm that is responsible for choosing the next individual (offspring) based on the fitness function. That means, the potential solution with the minimum RMSE, best fitness function, has a higher probability to be selected. Selection pressure is a crucial point in genetic algorithms which is a deterministic function to evaluate the strength of the selection approach. When the selection pressure is high we reach the genetic convergent state quickly, limiting the solutions space (Lu, H., et al 2017) (Lamini, C., et al 2018).

3. Cluster Crossover Operator

It's one of the main features of genetic algorithms, it's also called recombination. The main evolutionary crossover techniques are random crossover, order crossover, heuristic crossover, alternating crossover, and arithmetic crossover. The arithmetic crossover operator is divided into the single, simple and whole arithmetic crossover. After the selection phase is completed, the crossover approach is initiated. It exchanges the genetic information (genes) between two chromosomes to get new reproductions (two offspring) from the parents for the next generation. The importance of the crossover method is to produce new Childs which are more likely better than the old individuals, also enhancing the search process to get the best solution that mainly depends on the new chromosomes (Meena, D.C. and Devanshu, A., 2017).

In our hybrid model, we improve the crossover technique through a clustering approach with a crossover probability of 0.9 and population size (60-300) as shown in figure 3.11. We perform k-mean clusters on parents then apply an arithmetic crossover that randomly chooses two parents. As expressed in the following formula 3.2.2.4.1 which describes a linear combination of parents (P1, P2) to generate new Childs (C1, C2) (Furqan, M., et al 2017).

$$c1 = \alpha P1_{gene} + (1 - \alpha)P2_{gene} \quad 3.14$$

$$c2 = (1 - \alpha)P1_{gene} + \alpha P2_{gene}$$

where α is a random number between zero and one.

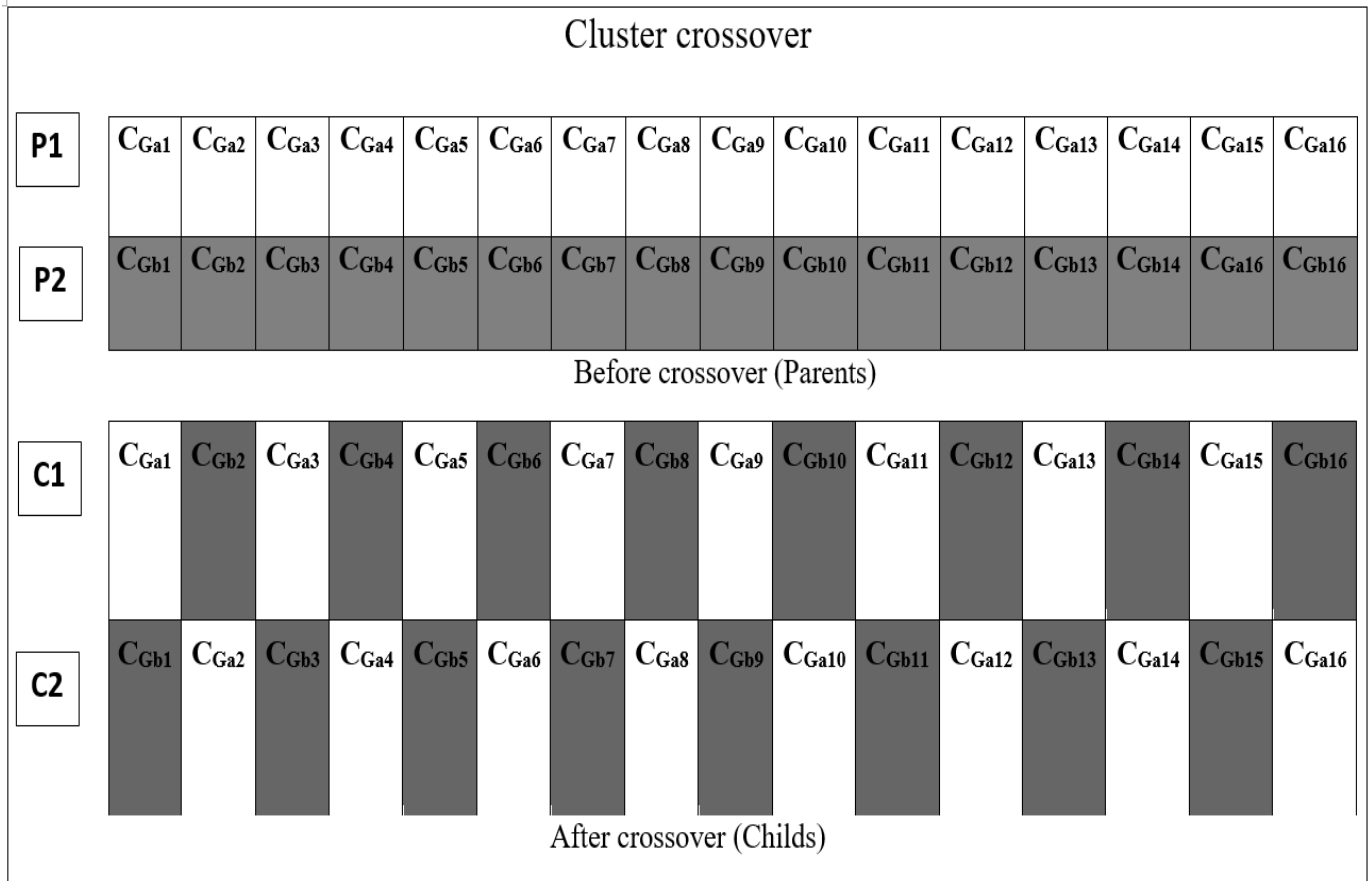


Figure 3.11 16- period cluster crossover

4. Mutation operator

The main objective of the mutation process is to make it immune to fall in the local optimum solution through expanding the diversity in the initial population to ensure reaching the entire search space by the Genetic algorithm. A mutation process comes after crossover operation with a certain probability usually low to avoid the premature solution, values around 0.1 or 0.01 are common, because of the high probability rate that could destroy the fit individual and turn the GAs to the random search. The mutation method is a process applied to the offspring itself like replacing indexes (gene) with others. The mutation probability and generations assigned (0.1, 500) respectively. In our proposed model, we multiply the gene by a constant, after tests we get the best solution with 2 as expressed in the following formula.

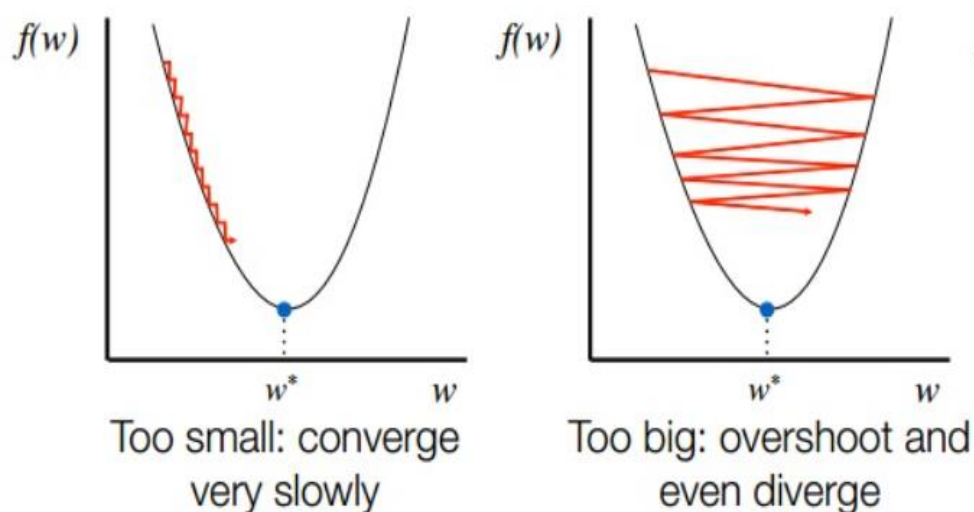
Mutated Kid:
$$mKid = mParent / 2. \tag{3.15}$$

Where mParent is the selected mutation parent.

Mutation Constant Multiplied	RMSE Training
0.125	0.059943
0.25	0.057565
2	0.041923
4	0.052564
6	0.064221
8	0.059452
10	0.059215

Table 3.1 The Mutation Constant Multiplied

In the above table 3.1, we conducted a test where we took the mutation constant multiplied to complete the best group of the GAs functions. We tested for a range of values from 0.125 to 10, the minimum RMSE value for training is represented by the value 2, so we consider it the best mutation constant.



(a) With too small mutation probability

(b) With too big mutation probability

Figure 3.12 The Affected Mutation Rate

As shown in figure 3.12 with the high mutation probability, the risk of missing the desired solution is very high that can cause overshoot and diverge even more, nonetheless with low mutation probability can cause a very slow convergence. (Meena, D.C. and Devanshu, A., 2017).

Hybrid Model Pseudo Code:

Algorithm 1: The general procedure that was used in building the Recurrent models

Input:

Load the electricity datasets (line 1 & 2 dataset)
 Input - input time series (monthly, seasonal, monthly electricity consumption).
 Target - feedback time series.

Output:

Prediction electricity consumptions and MSE, RMSE matrices

Data Preprocessing:

// Normalization and Smoothing the loaded Dataset
 range = [min(data); max(data)];

mindata = min(data);
 mindata = mindata(ones(size(data,1),1), :);
 data1 = (data-mindata);

maxdata = max(data);
 maxdata = maxdata(ones(size(data,1),1), :);
 data2 = (maxdata-mindata);
 data = data1 ./ data2;

partition normalized dataset (Training and Testing data)

Neural network Initialization:

Initialize NARX (Select architecture type).

Choose a Training Function, input Delays, feedback delays; the number of Hidden Units, *activation function g(.)*, cost-function, Number of training and testing time steps .

Prepare training data shifted based on the delay time step.

Configure NARX based prepared data.

Output: P number of models, Result Object;

Start optimized approach to enhance the NARX weights

Step 1: Initialize Parameters.

Choose Population size, number of generations, and GA Operator's rate.
 Preprocessing data

Step 2: Apply the features of GAs and Optimized approach

Initialize Population ←

Population Initialization by K Mean Clustering

```

pop = kmeans (popSize, NVARs);
pop1 = randn (popSize, NVARs);
rng (1); %Necessary for both Kmean and Fuzzy
ss=size(pop1); %Necessary for both Kmean and Fuzzy
[idx, c] =kmeans (pop1, ss (1)); % Kmean clustering
Pop=c; % Kmean Clustering
end

```

Evaluate GAs Fitness Function to find the Optimal Solution:

Fitness Function (Population, Population Size, Weights, Input, Target) ←

- **Apply Uniform Selection**
 - **Apply Cluster Crossover Operation**
- ```

for k=1: nXoverKids
 parent1 = pop(xParents(index), :);
 score1 = scores(xParents(index));
 index = index + 1;
 parent2 = pop(xParents(index), :);
 score2 = scores(xParents(index));
 index = index + 1;

```

```

% Secturring %
Len = length(parent1);
parent1 = num2hex(parent1);
parent2 = num2hex(parent2);
numberOfcluster = 16;
for i = 1: Len
 for j = 1: 2: numberOfcluster
 tmp = parent1(i, j);
 parent1(i, j) = parent2(i, j+1);
 parent2(i, j+1) = tmp;
 end
end
parent1 = hex2num(parent1);
parent2 = hex2num(parent2);

```

- **Apply Uniform Mutation Function**
- **Generate New Genotype.**
- **Evaluate Fitness Until Maximum Iteration reached.**
- **Assign the Best Individual (Optimal Weights) to the RNN.**

```

bestIdx = scores == min(scores);
best= pop (bestIdx, :);
% initNet = init(net);
net = setwb (net, best. ');
end

```

### Step 3: Set the Optimal Weight for NARX Optimized.

#### Step 4: Performance

- **Compute RMSE and Forecasted Outcome using Training Data.**

```

for i = 1: popSize
 % Assign each individual in the population to the RNN.
 % initNet = init(net);
 net = setwb (net, pop (i, :));

 % Calculate the actual values
 pred = net (Input, Xi, Ai);

 % Calculate the root-mean-square error (RMSE).

```

```
scores(i) = sqrt(mean((cell2mat(Target) - cell2mat(pred)). ^2));
end
```

- **Testing Phase**  
Compute RMSE and Forecasted Outcome using Training Data.

### 3.5 Summary

For electricity consumption workload time series forecasting, in this chapter, we implemented three hybrid models (RNNs-ES, NARX-ES, NARX-OA-Kmean approach), and described them in detail. Besides, we selected the best combination of genetic algorithm functions (initialization technique, Uniform Selection function, Uniform Mutation function, and arithmetic Crossover function) that has reduced the cost function (RMSE). We also introduced and illustrated the performance function RMSE that we used.

# Chapter 4

## Experiments and Results

## 4. Result and Discussion

### 4.1 Electricity Dataset

The research population would be every individual or company that uses the electricity service (or the municipality). Electricity consumption is measured periodically by the local municipality in Palestine. In this study, the process of preparing the collected electricity consumption datasets of Tulkarm is introduced. The results are the electricity consumption forecast. We compare the result for each NN's technique with our optimized hybrid models. Then, we explain and describe accurately how the results were obtained for the applied models with numerical and graphical results. The network targets are the actual hourly, daily; monthly and yearly for Tulkarm electricity consumptions from 2018-2020.

The network output is the prediction period for the next time series. The main goal of these experiments is to find a model with a closer result to real electricity consumption. These findings, which will be expressed as the Root Mean Square Error "RMSE" factor, is a good measure of accuracy. Data which is the target time-steps divided into three target time-steps randomly using cross-validation, the first one with 70% of Input-Target pairs from the total data for training, 15% for the second part (validation) used to measure network generalization, and 15% for the testing part used to measure the network performance during and after training.

Data set samples to range from 365 days for the yearly forecasting, 120 days for the seasonal forecasting, 30 days for the monthly forecasting and 1500 data points measured every 30 minutes during July, Table 4.1 illustrate all the dataset used.

| Period    | Forecasting type | Electricity resources | Target time steps (input vector) | Train<br>70% | Test<br>30% |
|-----------|------------------|-----------------------|----------------------------------|--------------|-------------|
| 2018-2020 | Monthly          | Line 1+2              | 90 value                         | 63           | 27          |
|           | Seasonally       | Line 1+ 2             | 363 value                        | 258          | 105         |
|           | Yearly           | Line 1+ 2             | 1095 value                       | 773          | 322         |
| 2020      | Per half hour    | Current intensity     | 1500 value                       | 1050         | 450         |

Table 4.1 Dataset Description

## 4.2 Experiment Procedure

This data is applied in our proposed model in Matlab R2019b under Windows 10 with processor i7, to ensure the performance of the proposed approach. Before simulating the data, it was processed and normalized to improve the accuracy of the result according to Equations (3.3) and (3.4).

In this study, we performed the Levenberg-Marquardt training algorithm with recurrent (RNN and NARX) and hybrid recurrent genetic algorithms on dataset samples. The Levenberg-Marquardt training algorithm stops when the improvement of the generalization stops, which is indicated by increasing RMSE. The Levenberg-Marquardt function needs more memory and less time.

The general procedure we used to acquire results constitute of:

□ Data:

- Load the electricity resources from 2018-2020 (Line 1&2, current intensity, One month, summer and winter season, One Year)
- Pre-processing data (smoothing and normalized).

- Training and testing normalized data.
- Recurrent time series models (RNN, NARX)
- Initialize RNN, NARX model (choose model type).
  - Define the execution process (number of neurons, input delays, target, number of hidden units, number of time-steps, feedback delays, and train function).
  - Data preparation (shifted time series)
  - Configure time series model (RNN, NARX) based on time series electricity data.
- Begin Optimization strategy using an evolutionary algorithm (Genetic algorithm with clustering approach).
- I. Phase 1: initialize k-mean clustering with defining parameters such as the number of generations, population size, and the rate of GA operators'. Moreover, the dataset should be pre-processed.
  - II. Phase 2: proceed with the genetic algorithms basic operation:
    - Define population
    - Finding the optimal solution by estimating the fitness function
    - Replicate
      - Estimate uniform selection
      - Perform cluster approach (16-period cluster).
      - implement an arithmetic crossover function.
      - Apply uniform mutation function
      - Generate the next population.
      - Evaluate fitness until reaching the maximum iteration.
    - Define the optimal weights for NARX (NARX optimized).

### □ performance

- Using training data to find the outcome of forecasting.
- Find Root Mean Square Error (RMSE).

### □ Testing Phase

- Using testing data to find the outcome of forecasting.
- Find Root Mean Square Error (RMSE).

In this section, we introduce the parameter definition to choose the best one. First, as explained in the previous chapter, we determine the best RMSE when the mutation constant = 2.

### ● Population Size:

In this step, we introduce the population size under the experiment run with various neurons (2,4,8,10) where each neuron runs with different crossover and mutation probabilities, generation =500. The experiment population size was set based on different types of time series, from 20 - 400, as shown in table 4.2, the best RMSE values for training and testing are represented by sample 7 (pop size = 140), and sample 12 (pop size=240).

Table 4.2 Selection of Population Size based on Recurrent Hybrid Model Results.

| RNNs -OA model<br>Generation =500,<br>number of neuron=2,4,8,10 |                 | Electricity L 1+2 (2018-2020) |              | Summer L 1+2 (2018-2020) |              |
|-----------------------------------------------------------------|-----------------|-------------------------------|--------------|--------------------------|--------------|
| Samples                                                         | Population size | RMSE Training                 | RMSE Testing | RMSE Training            | RMSE Testing |
| Sample 1                                                        | 20              | 0.052365                      | 0.093099     | 0.14345                  | 0.092947     |
| Sample 2                                                        | 40              | 0.055047                      | 0.19942      | 0.15067                  | 0.092911     |
| Sample 3                                                        | 60              | 0.058913                      | 0.61203      | 0.13836                  | 0.48959      |
| Sample 4                                                        | 80              | 0.046053                      | 0.12593      | 0.15277                  | 0.14128      |
| Sample 5                                                        | 100             | 0.056167                      | 0.14065      | 0.14702                  | 0.13917      |
| Sample 6                                                        | 120             | 0.059357                      | 0.16679      | 0.15277                  | 0.32482      |

|           |     |          |          |         |          |
|-----------|-----|----------|----------|---------|----------|
| Sample 7  | 140 | 0.060972 | 0.15909  | 0.14454 | 0.1017   |
| Sample 8  | 160 | 0.052988 | 0.32338  | 0.15277 | 0.14652  |
| Sample 9  | 180 | 0.057601 | 0.37169  | 0.14776 | 0.1014   |
| Sample 10 | 200 | 0.058953 | 0.10943  | 0.1446  | 0.098192 |
| Sample 11 | 220 | 0.040711 | 0.3672   | 0.14461 | 0.85354  |
| Sample 12 | 240 | 0.044017 | 0.090686 | 0.14646 | 0.11689  |
| Sample 13 | 260 | 0.053391 | 0.17501  | 0.1482  | 0.21299  |
| sample14  | 280 | 0.044429 | 0.12026  | 0.1436  | 0.13184  |
| Sample 15 | 300 | 0.046867 | 0.12018  | 0.13974 | 0.2943   |
| Sample 16 | 320 | 0.049091 | 0.2625   | 0.13953 | 0.10418  |
| Sample 17 | 340 | 0.051558 | 0.091879 | 0.14334 | 0.14024  |
| Sample 18 | 360 | 0.055812 | 0.092482 | 0.14026 | 0.22629  |
| Sample 19 | 380 | 0.051208 | 0.088989 | 0.14466 | 0.14267  |
| Sample 20 | 400 | 0.051722 | 0.1409   | 0.13713 | 0.12634  |

- **Generation:**

The next step is to find the best number of generations, the number of neurons =2, population size = 240 which was obtained in the previous step (sample 12). The best RMSE training and testing was at generation = 50 as shown in table 4.3. Moreover, when the number of neurons = 4 and 6, the best RMSE training and testing was also at generation = 50, then after this value any increase in the number of generations won't have any significance in RMSE as shown in table 4.4.

|                             |                                                           |
|-----------------------------|-----------------------------------------------------------|
| RNNs -OA model<br>Neuron =2 | Electricity L 1+2 (2018-2020)<br>Pop size=240 , #neuron=2 |
|-----------------------------|-----------------------------------------------------------|

| Samples  | Generation | RMSE Training | RMSE Testing |
|----------|------------|---------------|--------------|
| Sample1  | 20         | 0.057714      | 0.32759      |
| Sample 2 | 50         | 0.063422      | 0.092559     |
| Sample 3 | 80         | 0.063422      | 0.092559     |
| Sample 4 | 110        | 0.063422      | 0.092559     |
| Sample 5 | 140        | 0.063422      | 0.092559     |
| Sample 6 | 170        | 0.063422      | 0.092559     |
| Sample 7 | 200        | 0.063422      | 0.092559     |

Table 4.3 Generation Selection based on Recurrent Hybrid Model Results (Electricity L1+2)

When the population size = 140 which was obtained in the previous step (sample 7), the best RMSE training and testing were also at generation = 50, where the number of neurons = 2, 4,

We conducted 7 experiments from Generation 20 to 200, Pop size =240 and neuron =2. As shown in the above table 4.3, we get the best RMSE Training and Testing (RMSE Training =0.063422, RMSE Testing =0.092559) when Generation =50. So we consider sample 2 to represent the best generation.

| RNNs -OA model<br>Neuron =(4,6) |            | Electricity L 1+2 (2018-2020)<br>Pop size=240 |              |
|---------------------------------|------------|-----------------------------------------------|--------------|
| Samples                         | Generation | RMSE Training                                 | RMSE Testing |
| Sample 1<br>Neuron =4           | 20         | 0.056563                                      | 0.22799      |
| Sample 2<br>Neuron =4           | 50         | 0.056563                                      | 0.22799      |
| Sample 3<br>Neuron =4           | 80         | 0.056563                                      | 0.22799      |
| Sample 4<br>Neuron =6           | 20         | 0.061025                                      | 0.17268      |
| Sample 5<br>Neuron =6           | 50         | 0.054765                                      | 0.13278      |

Table 4.4 Generation Selection based on Recurrent Hybrid Model Results (Electricity L1+2 Dataset)

The experiment runs with (4,6) neurons, Pop size =240, and Mutation constant =2 as shown in table 4.4, from all samples applied in the RNNs-OA model, the minimum RMSE values for training and testing is represented by sample 2 when neuron =4 and sample 5 when neuron =6. So we consider sample 2 and sample 5 to represent the best generation.

Table 4.5 Generation Selection based on Recurrent Hybrid Model Result (Summer L 1+2 Dataset)

| RNNs -OA model | Summer electricity L 1+2 (2018-2020)<br>Pop size=140 , #neuron=2,4,6 |               |              |
|----------------|----------------------------------------------------------------------|---------------|--------------|
|                | Generation                                                           | RMSE Training | RMSE Testing |
| Sample1        | 20                                                                   | 0.15854       | 0.16008      |
| Sample 2       | 50                                                                   | 0.092947      | 0.1446       |
| Sample 3       | 80                                                                   | 0.11841       | 0.17558      |
| Sample 4       | 110                                                                  | 0.11841       | 0.17558      |
| Sample 5       | 140                                                                  | 0.11841       | 0.17558      |
| Sample 6       | 170                                                                  | 0.11841       | 0.17558      |
| Sample 7       | 200                                                                  | 0.11841       | 0.17558      |

The experiment runs on summer electricity dataset with (2,4,6) neurons, Pop size =240, and Mutation constant =2 as shown in table 4.5, from all samples applied in the RNNs-OA model, the minimum RMSE values for training and testing is represented by sample 2 when neurons = (2,4,6). So we consider sample 2 to represent the best generation.

- **Crossover and mutation probabilities:**

Here the mutation and crossover probabilities were obtained when the number of neurons = (6,8), the best RMSE training and testing was when the crossover and mutation probabilities pairs = {0.8, 0.1} at sample = 1 for the summer season, and what applies for the summer season applies on the winter and one month. Also, the best RMSE training and testing were when the crossover and mutation probabilities pairs = {0.5, 0.4} at sample = 4 for the yearly forecasting shown in table 4.6.

| RNNs –OA model       |                          |                         | Electricity L 1+2<br>(2018-2020) |                 | Summer L 1+2 (2018-<br>2020) |                 |
|----------------------|--------------------------|-------------------------|----------------------------------|-----------------|------------------------------|-----------------|
| Sample<br>neuron=6,8 | Crossover<br>probability | Mutation<br>probability | RMSE<br>Training                 | RMSE<br>Testing | RMSE<br>Training             | RMSE<br>Testing |
| Sample 1             | 0.8                      | 0.1                     | 0.039835                         | 0.09308         | 0.08676                      | 0.14047         |
| Sample 2             | 0.7                      | 0.2                     | 0.04100                          | 0.11676         | 0.11506                      | 0.15565         |
| Sample 3             | 0.6                      | 0.3                     | 0.05118                          | 0.099963        | 0.11519                      | 0.16073         |
| Sample 4             | 0.5                      | 0.4                     | 0.04936                          | 0.08722         | 0.13912                      | 0.14691         |

Table 4.6 Crossover and Mutation Probabilities Selection Based On Recurrent Hybrid Model Result

This experiment included determining the best crossover and mutation probabilities pair, which will represent the number of the crossover and mutation kids in the next population, where the experiment runs with (6,8) neurons, population size = 240, number of generations = 50, elite probability = 0.1, and the best mutation constant=2. Then, from all samples applied in the RNNs-OA model, the minimum RMSE value for training and testing is represented by sample 1 for the electricity L (1+2) dataset, and sample 4 has a minimum RMSE value for testing as shown in table 4.6. So we consider sample 1 to represent the best crossover and mutation probabilities pair for yearly forecasting and sample 4 to represent the best crossover and mutation probabilities pair for seasonal forecasting.

| Dataset/Electricity Line 1+2(2018-2020), Model: NARX-OA<br>Generation =50 ,population size=240 ,mutation constant =2 |                                                                            |                                                                           |                                                                            |                                                                           |                                                                            |                                                                           |                                                                            |                                                                           |
|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Number of<br>neurons                                                                                                 | <b>RMSE<br/>Training</b><br>Crossover<br>prob:0.8<br>Mutation<br>prob: 0.1 | <b>RMSE<br/>Testing</b><br>Crossover<br>prob:0.8<br>Mutation<br>prob: 0.1 | <b>RMSE<br/>Training</b><br>Crossover<br>prob:0.7<br>Mutation<br>prob: 0.2 | <b>RMSE<br/>Testing</b><br>Crossover<br>prob:0.7<br>Mutation<br>prob: 0.2 | <b>RMSE<br/>Training</b><br>Crossover<br>prob:0.6<br>Mutation<br>prob: 0.3 | <b>RMSE<br/>Testing</b><br>Crossover<br>prob:0.6<br>Mutation<br>prob: 0.3 | <b>RMSE<br/>Training</b><br>Crossover<br>prob:0.5<br>Mutation<br>prob: 0.4 | <b>RMSE<br/>Testing</b><br>Crossover<br>prob:0.5<br>Mutation<br>prob: 0.4 |
| 2                                                                                                                    | 0.049546                                                                   | 0.15019                                                                   | 0.04724                                                                    | 0.24956                                                                   | 0.04560                                                                    | 0.098993                                                                  | 0.04219                                                                    | 0.24472                                                                   |

|     |          |         |         |         |         |          |         |          |
|-----|----------|---------|---------|---------|---------|----------|---------|----------|
| 4   | 0.044466 | 0.18989 | 0.05029 | 0.10026 | 0.04263 | 0.10439  | 0.04455 | 0.10692  |
| 6   | 0.039835 | 0.09308 | 0.04100 | 0.11676 | 0.05118 | 0.099963 | 0.04466 | 0.092527 |
| 8   | 0.047734 | 0.08956 | 0.04627 | 0.10382 | 0.04249 | 0.089796 | 0.04936 | 0.08722  |
| 10  | 0.046723 | 0.20581 | 0.04621 | 0.13632 | 0.05082 | 0.17866  | 0.04295 | 0.09437  |
| 120 | 0.04331  | 0.11594 | 0.04376 | 0.09416 | 0.04568 | 0.15377  | 0.04568 | 0.15377  |
| 14  | 0.04879  | 0.09162 | 0.05866 | 0.12287 | 0.04781 | 0.14613  | 0.04781 | 0.14613  |

Table 4.7 Crossover and Mutation Probabilities Selection Based On Recurrent Hybrid Model Results  
(Electricity Line 1+2)

In the above table 4.7, for generation =50, population size=240 and mutation constant =2 for the electricity line 1+2 dataset, we started changing the number of neurons from (2 - 14) neurons by adding 2 neurons at a time, to select the best number of the neurons that has the minimum RMSE for each dataset. After comparing the result of the recurrent hybrid model for RMSE training and testing for crossover and mutation probabilities, we found that the best result was when the crossover and mutation probabilities pairs = {0.5, 0.4} for neuron = 8 and RMSE training and testing are 0.04936 and 0.08722 respectively. On the other hand, when changing the population size to 140 for the same dataset we found that the best result was when the crossover and mutation probabilities pairs = {0.8, 0.1} also for neuron = 8 and RMSE training and testing are 0.14047 and 0.08676 respectively shown in table 4.8. So we consider when Pop size=240 to represent the best crossover and mutation probabilities pair.

| Dataset/Summer Electricity Line 1+2(2018-2020), Model: NARX-OA<br>Generation =50 ,population size=140 ,mutation constant =2 |                                                           |                                                          |                                                           |                                                          |                                                           |                                                          |                                                           |                                                          |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|----------------------------------------------------------|-----------------------------------------------------------|----------------------------------------------------------|-----------------------------------------------------------|----------------------------------------------------------|-----------------------------------------------------------|----------------------------------------------------------|
| Number of neurons                                                                                                           | RMSE Training<br>Crossover prob:0.8<br>Mutation prob: 0.1 | RMSE Testing<br>Crossover prob:0.8<br>Mutation prob: 0.1 | RMSE Training<br>Crossover prob:0.7<br>Mutation prob: 0.2 | RMSE Testing<br>Crossover prob:0.7<br>Mutation prob: 0.2 | RMSE Training<br>Crossover prob:0.6<br>Mutation prob: 0.3 | RMSE Testing<br>Crossover prob:0.6<br>Mutation prob: 0.3 | RMSE Training<br>Crossover prob:0.5<br>Mutation prob: 0.4 | RMSE Testing<br>Crossover prob:0.5<br>Mutation prob: 0.4 |
| 2                                                                                                                           | 0.43593                                                   | 0.14056                                                  | 0.13493                                                   | 0.20974                                                  | 0.13755                                                   | 0.19766                                                  | 0.13801                                                   | 0.12272                                                  |
| 4                                                                                                                           | 0.14295                                                   | 0.15006                                                  | 0.1382                                                    | 0.13046                                                  | 0.13685                                                   | 0.08676                                                  | 0.13685                                                   | 0.09963                                                  |
| 6                                                                                                                           | 0.13983                                                   | 0.09957                                                  | 0.14055                                                   | 0.084827                                                 | 0.14138                                                   | 0.10367                                                  | 0.14249                                                   | 0.13174                                                  |
| 8                                                                                                                           | 0.14047                                                   | 0.08676                                                  | 0.14175                                                   | 0.10581                                                  | 0.14018                                                   | 0.10423                                                  | 0.14014                                                   | 0.11468                                                  |
| 10                                                                                                                          | 0.15769                                                   | 0.14103                                                  | 0.13829                                                   | 0.11634                                                  | 0.14145                                                   | 0.11574                                                  | 0.13898                                                   | 0.12336                                                  |
| 12                                                                                                                          | 0.1409                                                    | 0.09033                                                  | 0.13759                                                   | 0.13759                                                  | 0.13759                                                   | 0.14335                                                  | 0.14092                                                   | 0.1067                                                   |
| 14                                                                                                                          | 0.14309                                                   | 0.10257                                                  | 0.14126                                                   | 0.097081                                                 | 0.1486                                                    | 0.09536                                                  | 0.14157                                                   | 0.09644                                                  |

Table 4.8 Crossover and Mutation Probabilities Selection Based On Recurrent Hybrid Model Results  
(Summer Electricity Line 1+2)

As explained in Table 4.9, we determined the general characteristics of the optimized algorithm (OA) which has been used for all our proposed models, with different types of time-series datasets.

| Parameter                        | June month | Summer Electricity Line 1+2 (2018-2020) | Electricity Line 1+2 (2018-2020) |
|----------------------------------|------------|-----------------------------------------|----------------------------------|
| The fittest Number of generation | 7          | 50                                      | 50                               |
| Population Size                  | 15         | 140                                     | 240                              |
| The fittest neurons number       | 6          | 6,8                                     | 6,8                              |

| Measure of Performance                    | RMSE                            | RMSE                            | RMSE                            |
|-------------------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Selection Function                        | Uniform                         | Uniform                         | Uniform                         |
| Mutation Function                         | Uniform (modified)              | Uniform (modified)              | Uniform (modified)              |
| Crossover function                        | Arithmetic crossover (modified) | Arithmetic crossover (modified) | Arithmetic crossover (modified) |
| Mutation Constant                         | 2                               | 2                               | 2                               |
| Crossover and Mutation Probabilities Pair | (0.8,0.1)                       | (0.8,0.1)                       | (0.5,0.4)                       |

Table 4.9 General Characteristics of the RNNs-OA hybrid model for the dataset

### 4.3 Results

#### 4.3.1 Multilayer -Perceptron Model (MLP):

- **One-Month Forecasting**

For the monthly prediction, we chose the months of January and June from 2018 to 2020 as these months are supposed to have the maximum electricity consumption in a year in Palestine. For each month, the daily readings represent the mean value of every 30 minutes of electricity consumption from two lines, then the monthly result from 2018 to 2020 was combined by taking the average to be applied for MLP, RNN, NARX, and the proposed hybrid RNN models (RNN-ES, NARX-ES, NARX-OA-K mean), taking into account the number of executions starting from 2 neurons up to 14 neurons as shown in the tables from Table 4.10 to Table 4.22.

| Neuron | MSE training            | MSE Testing             |
|--------|-------------------------|-------------------------|
| 2      | $3.894 \times 10^{-3}$  | $5.0524 \times 10^{-3}$ |
| 4      | $3.5020 \times 10^{-3}$ | $4.4578 \times 10^{-3}$ |
| 6      | $1.240 \times 10^{-3}$  | $1.741 \times 10^{-2}$  |
| 8      | $4.362 \times 10^{-3}$  | $3.366 \times 10^{-3}$  |

|    |                       |                       |
|----|-----------------------|-----------------------|
| 10 | $3.527 \cdot 10^{-3}$ | $4.533 \cdot 10^{-3}$ |
| 12 | $4.527 \cdot 10^{-3}$ | $4.801 \cdot 10^{-3}$ |
| 14 | $3.957 \cdot 10^{-3}$ | $1.053 \cdot 10^{-2}$ |

Table 4.10: Result of June month MSE for MLP

As illustrated in Table 4.10 the proposed MLP model has the best results when executing 6 neurons with the minimum MSE training ( $\text{MSE}_{\text{train}} = 1.240 \cdot 10^{-3}$ ) and with increasing the number of hidden layers to 8 neurons, which was the lowest MSE testing ( $\text{MSE}_{\text{testing}} = 3.366 \cdot 10^{-3}$ ) as shown in Figure 4.1. So we decide to consider the best MSE when the number of hidden layers = 8. The results demonstrate that the number of neurons affects the MSE results. When the number of neurons isn't sufficient the MSE error increases and causes under-training, but when the number of neurons further increases beyond the critical point that leads to over-training which is due to the instability of NN learning.

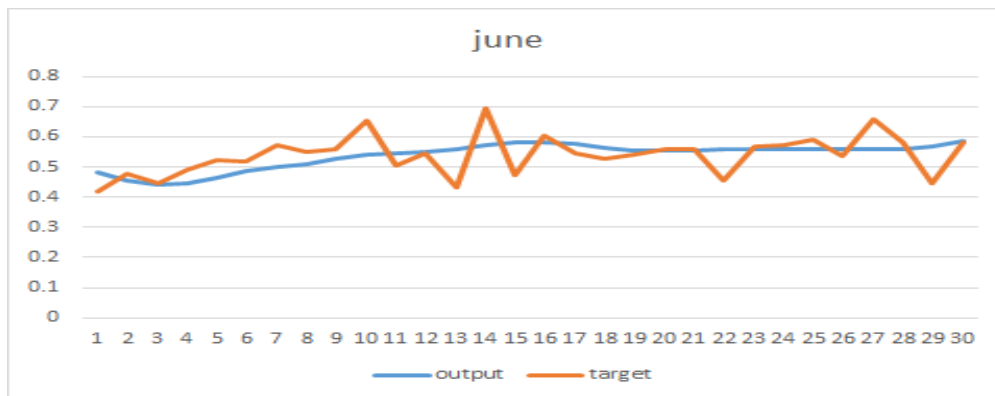


Figure 4.1: Real and predicted June month (2018-2020) using MLP

As illustrated in Figure 4.1, the target and output values are not close to each other. Thus, we deduce that the MLP model is not a suitable approach for forecasting one month's electricity consumption. And this indicates that the MLP method will give inaccurate results for

predictions.

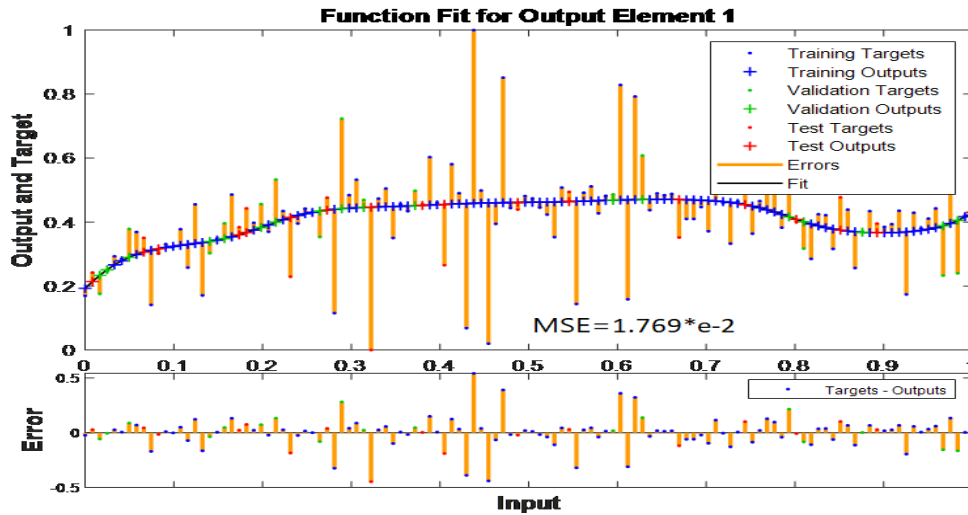


Figure 4.2 Real and predicted electricity consumption values using MLP, and the error value

As shown in Figure 4.2, their real and predicted values are not close to each other due to the weakness of the prediction model.

- **One Season Forecasting:**

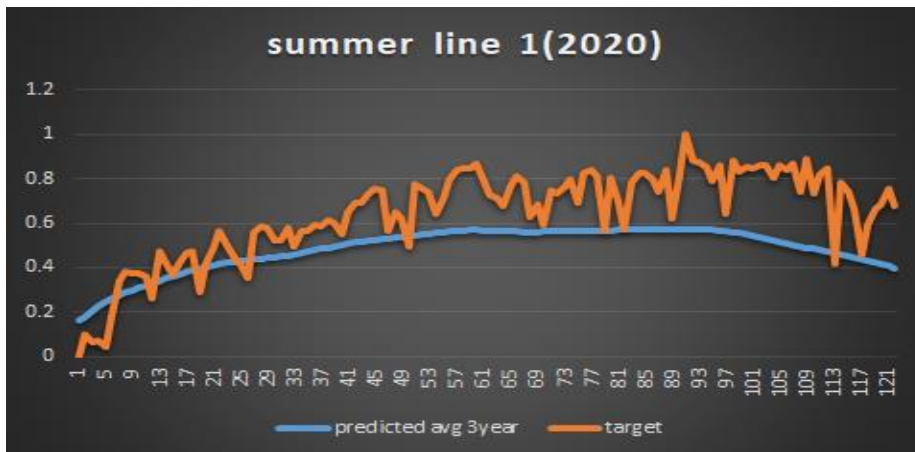


Figure 4.3: Real and Predicted Summer Line 1 based on the average of three previous years using

MLP

As illustrated in Figure 4.3, the target and output values are not close to each other. Thus, we deduce that the MLP model is not a suitable approach for forecasting one-season electricity

consumption. And this indicates that the MLP method will give inaccurate results for predictions.

- **One-Year Forecasting:**

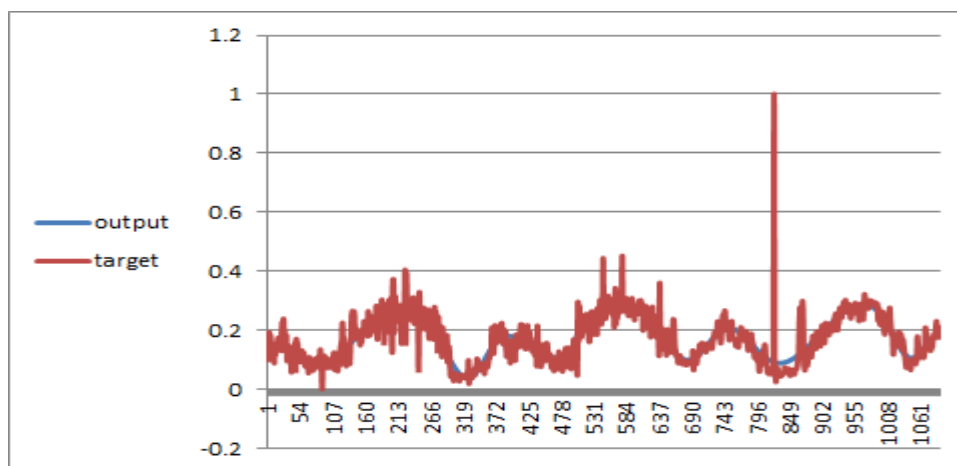


Figure 4.4: Real and predicted line 1+2 using MLP

As illustrated in Figures 4.4, the target and output values are not close to each other. Thus, we deduce that the MLP model is not a suitable approach for forecasting one-year electricity consumption. And this indicates that the MLP method will give inaccurate results for predictions.

## 4.3.2 Recurrent Models

### 4.3.2.1 NARX Model

- **One-Month Forecasting (June Month)**

| Neurons | MSE training         | MSE Testing           |
|---------|----------------------|-----------------------|
| 2       | $2.003 \cdot e^{-3}$ | $8.419 \cdot e^{-3}$  |
| 4       | $1.364 \cdot e^{-3}$ | $5.604 \cdot e^{-3}$  |
| 6       | $1.487 \cdot e^{-2}$ | $5.6000 \cdot e^{-3}$ |
| 8       | $3.563 \cdot e^{-3}$ | $1.683 \cdot e^{-3}$  |

|    |                       |                       |
|----|-----------------------|-----------------------|
| 10 | $2.717 \cdot 10^{-3}$ | $8.171 \cdot 10^{-3}$ |
| 12 | $4.902 \cdot 10^{-3}$ | $6.520 \cdot 10^{-3}$ |

Table 4.11: Results of June month MSE for NARX

In this experiment, when applying the NARX model, we started with 2 neurons and kept adding 2 each time, the results show that the model has best results when executing 4 neurons with the minimum MSE training =  $1.364 \cdot 10^{-3}$  and with increasing the number of hidden layers to 8 neurons, which has the lowest MSE testing =  $1.683 \cdot 10^{-3}$  as shown in table 4.11. So we decide to consider the best MSE when the number of hidden layers =8.

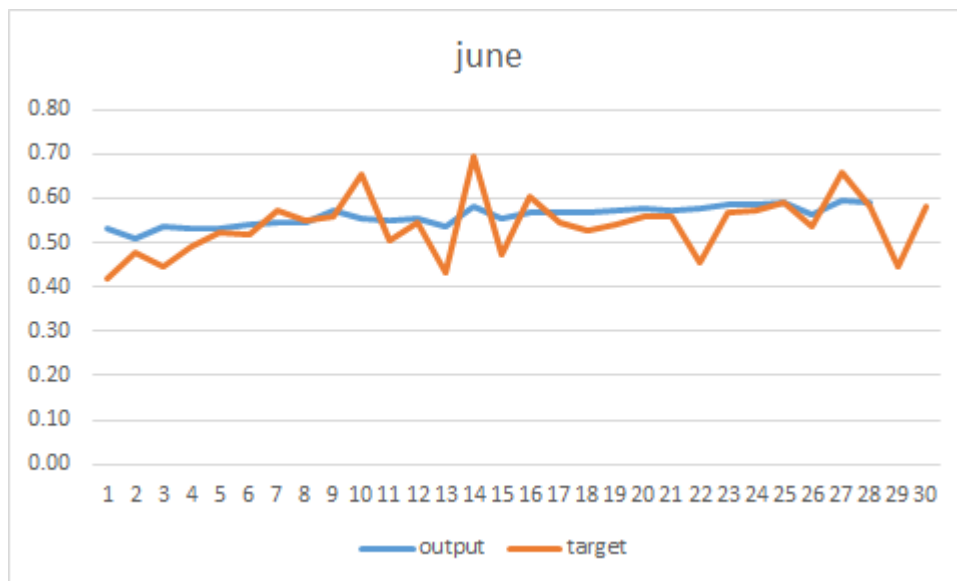


Figure 4.5.: Real and predicted June month using NARX

As illustrated in figure 4.5 as well as in MLP, the target and output values are not close to each other. So, this model is also not suitable for forecasting electricity consumption, as it does not give accurate results for predictions. Neither the season and year predictions are accurate, as shown in Figures 4.6 and 4.7.

- **One Season Forecasting:**

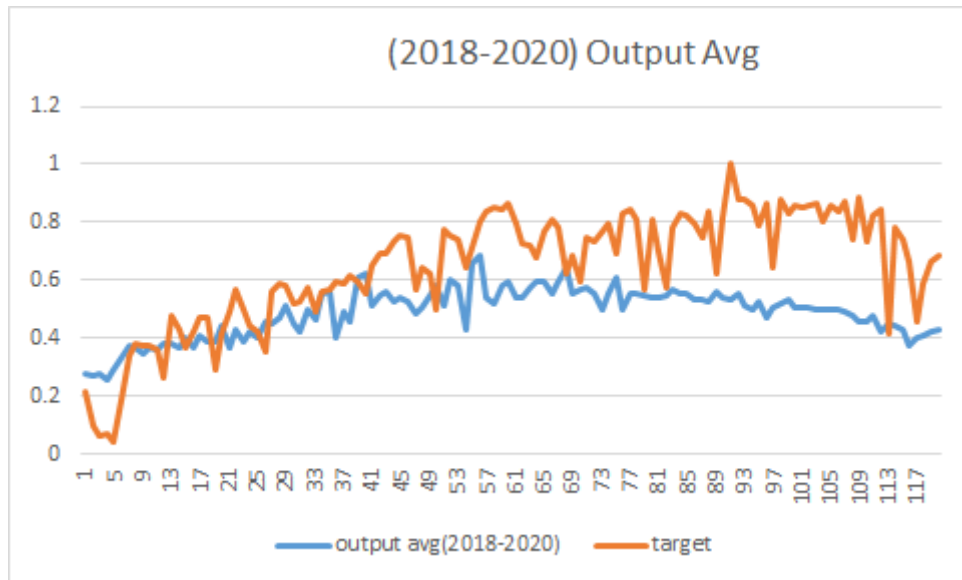


Figure 4.6: Real and Predicted Summer Line 1 of Average Three Years

As illustrated in Figure 4.6, the target and output values are not close to each other. Thus, we deduce that the NARX model is not a suitable approach for forecasting one-season electricity consumption. And this indicates that the NARX method will give inaccurate results for predictions.

- **One Year Forecasting:**

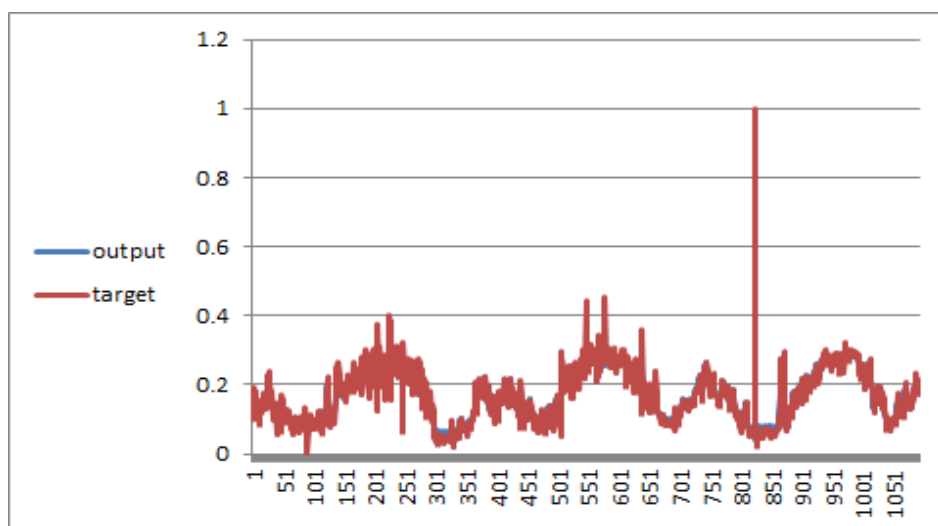


Figure 4.7: Real and Predicted Line 1+2 using NARX Model

As illustrated in Figure 4.7, the target and output values are not close to each other. Thus, we deduce that the NARX model is not a suitable approach for forecasting one-year electricity consumption. And this indicates that the NARX method will give inaccurate results for predictions.

#### 4.3.2.2 RNN Model

- **One-Month Forecasting:**

| Neurons | Epochs | Train data | Test data | Validation data | MSE                    |
|---------|--------|------------|-----------|-----------------|------------------------|
| 2       | 1000   | 70%        | 15%       | 15%             | 0.0135                 |
| 4       | 1000   | 70%        | 15%       | 15%             | 0.00669                |
| 6       | 185    | 70%        | 15%       | 15%             | $2.6 \times 10^{-9}$   |
| 8       | 137    | 70%        | 15%       | 15%             | $5.68 \times 10^{-14}$ |
| 10      | 20     | 70%        | 15%       | 15%             | $5.82 \times 10^{-24}$ |

Table 4.12 Model Training Result for Energy Prediction at June (One-Month)

In the RNN experiment as shown in table 4.12, we started with 2 neurons and 1000 iterations then added 2 gradually, as we reached 10 neurons we obtained the minimum MSE with  $5.82 \times 10^{-24}$  and minimum iterations equal to 20. Moreover, the real and predicted values for the one-month prediction were identical, which is perfect for a small amount of data as seen in figure 4.8.

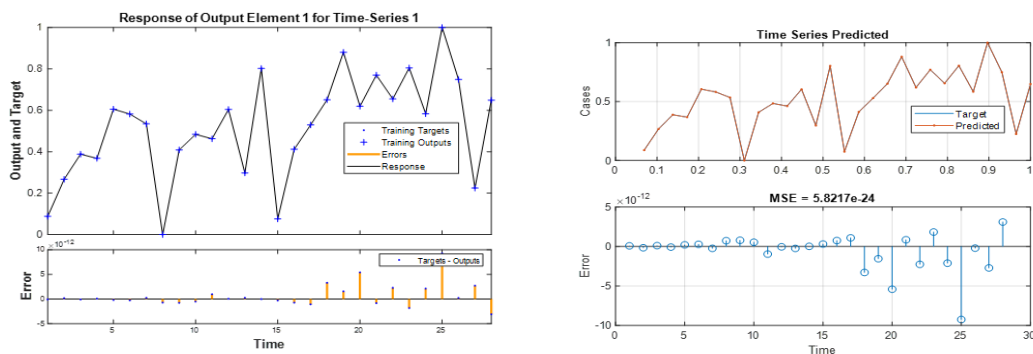


Figure 4.8: Real and Predicted Value June Month using RNN

As illustrated in Figure 4.8, the target and output values are very close to each other (identical). Thus, we deduce that the RNN model is a suitable approach for forecasting one-month electricity consumption. And this indicates that the RNN method will give accurate results for predictions.

- **One Season Forecasting:**

| Neurons | Epochs | Train data | Test data | Validation data | MSE      |
|---------|--------|------------|-----------|-----------------|----------|
| 2       | 1000   | 70%        | 15%       | 15%             | 0.014119 |
| 4       | 1000   | 70%        | 15%       | 15%             | 0.017785 |
| 6       | 1000   | 70%        | 15%       | 15%             | 0.018403 |
| 8       | 1000   | 70%        | 15%       | 15%             | 0.017502 |
| 10      | 1000   | 70%        | 15%       | 15%             | 0.016861 |
| 12      | 1000   | 70%        | 15%       | 15%             | 0.018205 |
| 14      | 276    | 70%        | 15%       | 15%             | 0.01878  |

Table 4.13: Model Training Result for Energy Prediction at Summer Line 1 Using RNN

On the other hand, when experimenting for the summer season and one-year prediction, we obtained minimum MSE with 0.014119 when we applied 2 neurons. As noticed from table 4.13 when the number of hidden layers increases by 2 neurons, the MSE gets worse. Thus the model is considered not effective when running on a big dataset as shown in figures 4.9 and 4.10.

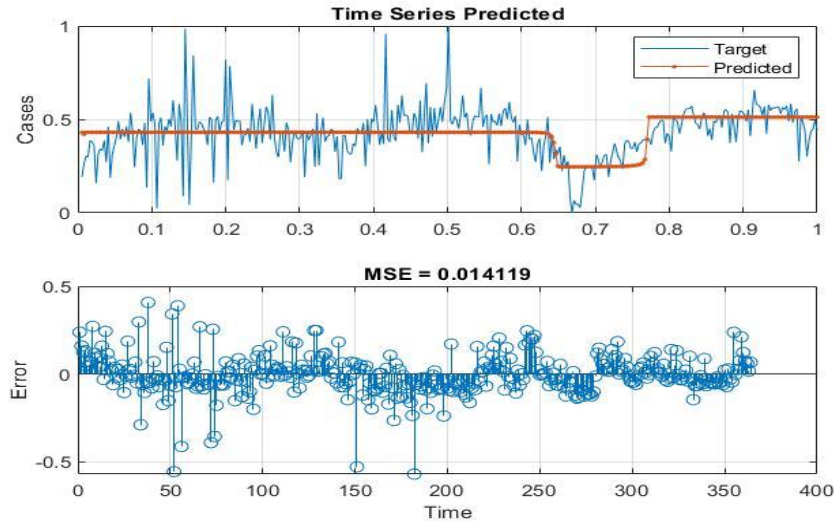


Figure 4.9: The Best Prediction Result of the Training Process for One-Season Using RNN

As illustrated in Figures 4.9, the target and output values are not close to each other. Thus, we deduce that the RNN model is not a suitable approach for forecasting one-season electricity consumption. And this indicates that the RNN method will give inaccurate results for predictions.

- **One Year Forecasting:**

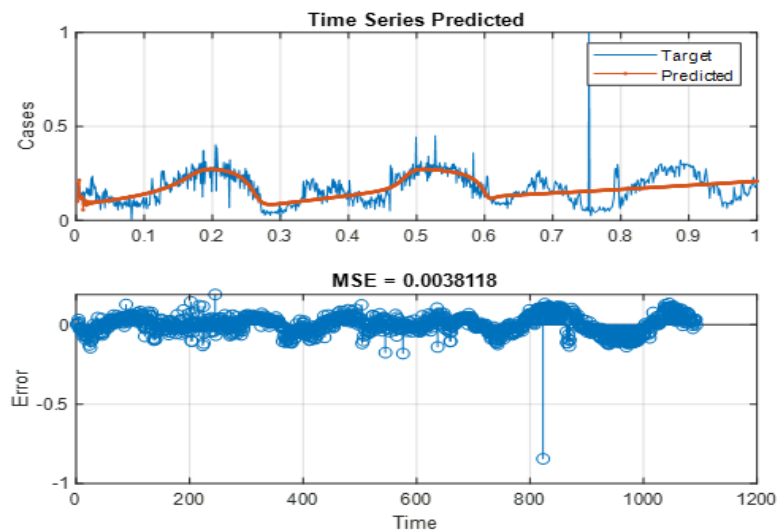


Figure 4.10 Prediction Result of the Training Process for One Year

AS illustrated in Figures 4.10, the target and output values are not close to each other. Thus, we deduce that the NARX model is not a suitable approach for forecasting one-year electricity consumption. And this indicates that the NARX method will give inaccurate results for predictions.

### 4.3.3 Hybrid Models

In this experiment, we divided the dataset into two types; short time series as a one-month dataset and long time series as one season, one year, and (half per hour) current intensity for 3000 sample datasets. Then we applied these data on three proposed hybrid models which are (RNN-OA, NARX-OA, NARX-OA-Kmean clustering).

#### 4.3.3.1 RNN-OA

- **One Month Prediction:**

In this model, we conducted two types of datasets (short time series and long time series datasets). First, as noticed in table 4.14 for the short data set (one-month forecasting), The RMSE training is at its lowest (RMSE training = 0.15323) when the hidden layer equals 8. But, RMSE testing at the same hidden layer is the worst (RMSE testing = 0.34952). Thus, we decide to choose the hidden layer when equal to 6 as it has the lowest RMSE testing (RMSE testing = 0.23779) and also it is RMSE training (RMSE training = 0.16924) which is close to the lowest RMSE when neuron equals 8.

| Neuron | RMSE training | RMSE testing |
|--------|---------------|--------------|
| 2      | 0.1893        | 0.2816       |
| 4      | 0.17728       | 0.30063      |

|    |         |         |
|----|---------|---------|
| 6  | 0.16924 | 0.23779 |
| 8  | 0.15323 | 0.34952 |
| 10 | 0.16237 | 0.31126 |
| 12 | 0.18552 | 0.35015 |
| 14 | 0.17925 | 0.32555 |

Table 4.14 Model Training Result for Energy Prediction (June Month)

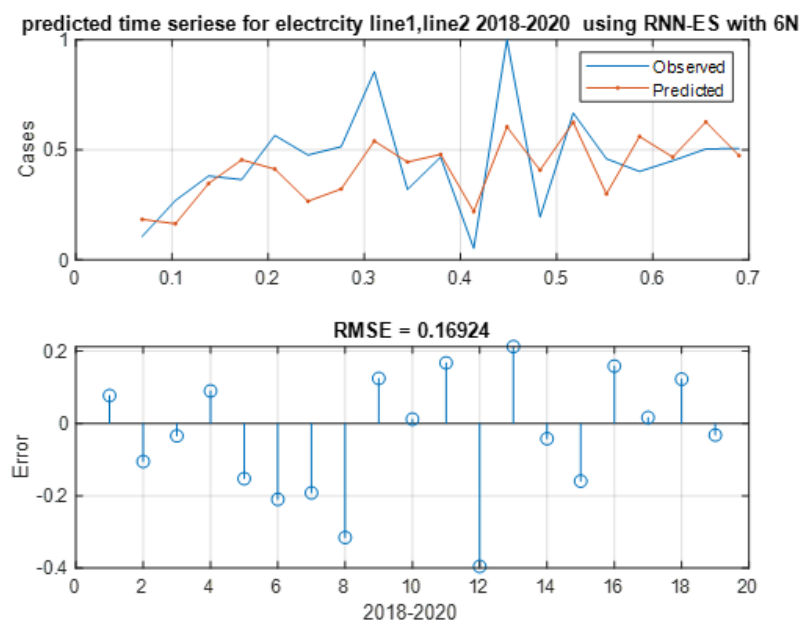


Figure 4.11 Prediction Result of the Training Process for One Month

Figure 4.11 shows the observed and predicted outputs deviate and do not match, which means that this model is weak for small amounts of data.

- **Summer and winter Seasons Prediction:**

| Neurons | RMS Training | RMS Testing |
|---------|--------------|-------------|
| 2       | 0.24601      | 0.30859     |

|    |         |         |
|----|---------|---------|
| 4  | 0.16465 | 0.20681 |
| 6  | 0.32056 | 0.5425  |
| 8  | 0.53022 | 0.66112 |
| 10 | 0.47995 | 0.66075 |
| 12 | 0.50165 | 0.61515 |
| 14 | 0.53492 | 0.49256 |

Table 4.15 Model Training Result for Energy Prediction (Summer Season)

As shown in table 4.15, the minimum RMSE was when we applied 4 neurons, where RMSE training= 0.20681 and RMSE testing = 0.16465, then after this value, it gets worse, which indicates that the best number of hidden layers is 4.

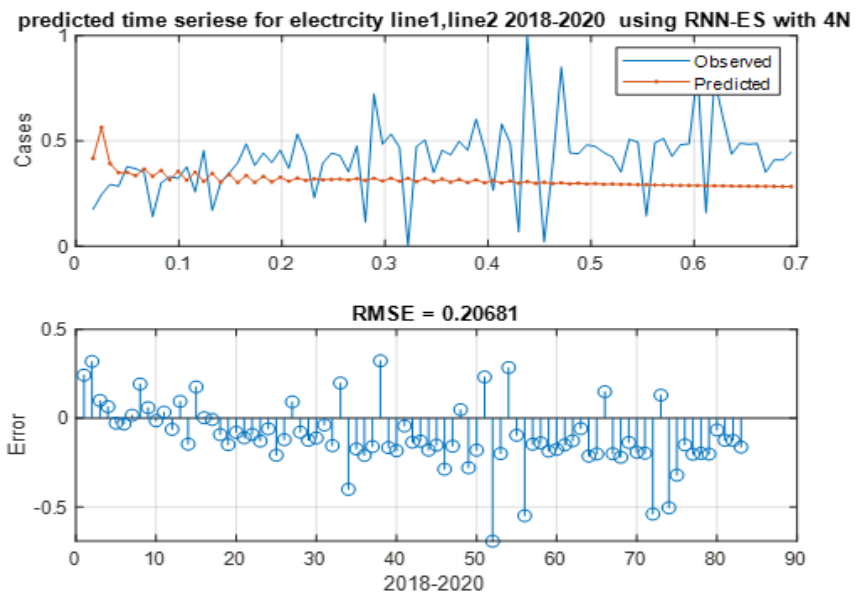


Figure 4.12 Prediction Result of the Training Process for Summer Season

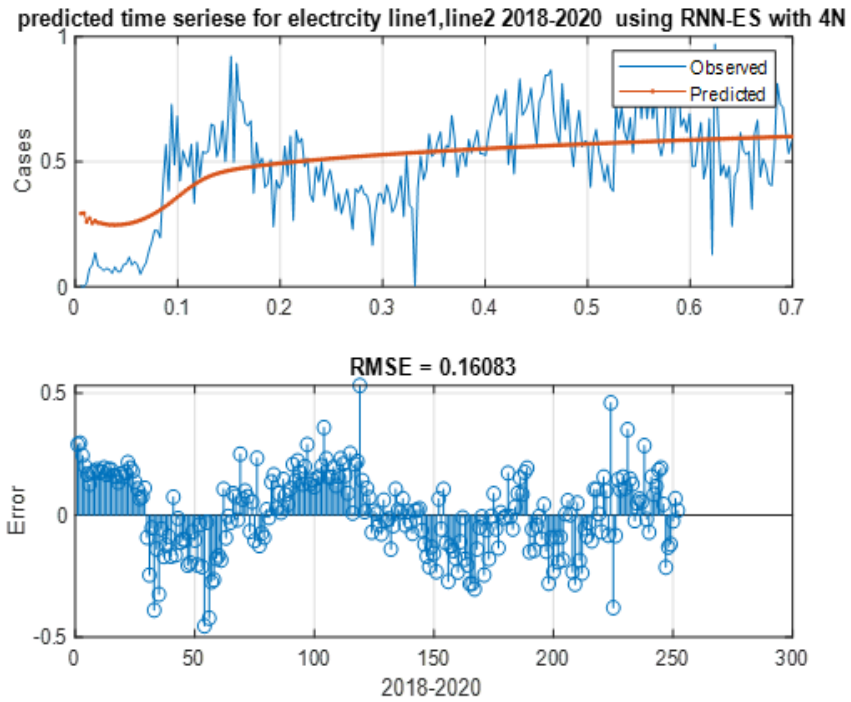


Figure 4.13 Prediction Result of the Training Process for Winter Season

In the above two figures 4.12 and 4.13, that demonstrate the results for summer and winter seasons predictions, where the predicted values appear as a straight line and the real data greatly deviates and are not identical. This means that this model could not be used to reflect accurate results.

- **One Year Prediction:**

| Neurons | RMSE Training | RMSE Testing |
|---------|---------------|--------------|
| 2       | 0.28861       | 0.297987     |
| 4       | 0.4791        | 0.29822      |
| 6       | 0.27408       | 0.31596      |
| 8       | 0.28452       | 0.28128      |

|    |        |         |
|----|--------|---------|
| 10 | 0.2781 | 0.28749 |
| 12 | 0.2926 | 0.30350 |
| 14 | 0.2726 | 0.31205 |

Table 4.16 Model Training Result for Energy Prediction

In the one-year prediction model, we started with 8 neurons as shown in table 4.16 with the minimum RMSE training and testing (0.28452, 0.28128) respectively. After that value any increase in the number of neurons causes the RMSE to give bad results.

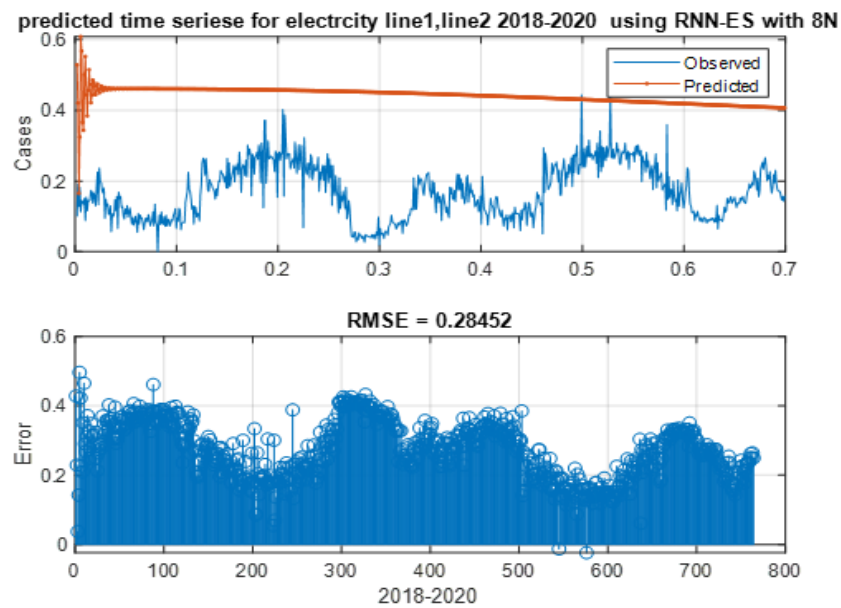


Figure 4.14: Prediction Result of the Training Process for One Year

As illustrated in figure 4.14 the real and predicted values do not match, as the predicted values represent a straight line, while all the real values are not identical, which means that this model could not be used to predict big data as it does not provide accurate results.

### 4.3.3.2 NARX-OA

Five sets of experiments were conducted to evaluate the proposed hybrid models based on changing the number of hidden layers from 2 to 14. All sets of experiments used nine types of parameter settings which determine the general characteristic of our proposed hybrid models as illustrated in Table 4.9.

- **One Month Prediction:**

| Neurons | NARX-OA       |              | NARX-OA-K Mean clustering |              |
|---------|---------------|--------------|---------------------------|--------------|
|         | RMSE Training | RMSE Testing | RMSE Training             | RMSE Testing |
| 2       | 0.20413       | 0.21994      | 0.20319                   | 0.37696      |
| 4       | 0.19606       | 0.21175      | 0.19756                   | 0.20064      |
| 6       | 0.20419       | 0.22029      | 0.21164                   | 0.2292       |
| 8       | 0.22633       | 0.30549      | 0.21922                   | 0.21672      |
| 10      | 0.19394       | 0.31777      | 0.21365                   | 0.26488      |
| 12      | 0.20312       | 0.32155      | 0.20458                   | 0.24982      |
| 14      | 0.21820       | 0.32720      | 0.23254                   | 0.26382      |

Table 4.17 Model Training Result for Energy Prediction (June Month)

As can be seen from Table 4.17, the results indicate that RMSE testing is the minimum along with all experiments, where we obtained the best RMSE testing and training when neurons equal 4 for the NARX-OA-K Mean clustering model which is (0.19756, 0.20064) respectively. As we noticed from the above table, through comparing all experiment runs for June month forecasting, the NARX-OA-K Mean clustering model demonstrates the best. On the other hand, as noticed in the two figures 4.15 and 4.16 the real and predicted values do not match, so we conclude that the hybrid model does not fit when the amount of data is small (< 100 samples).

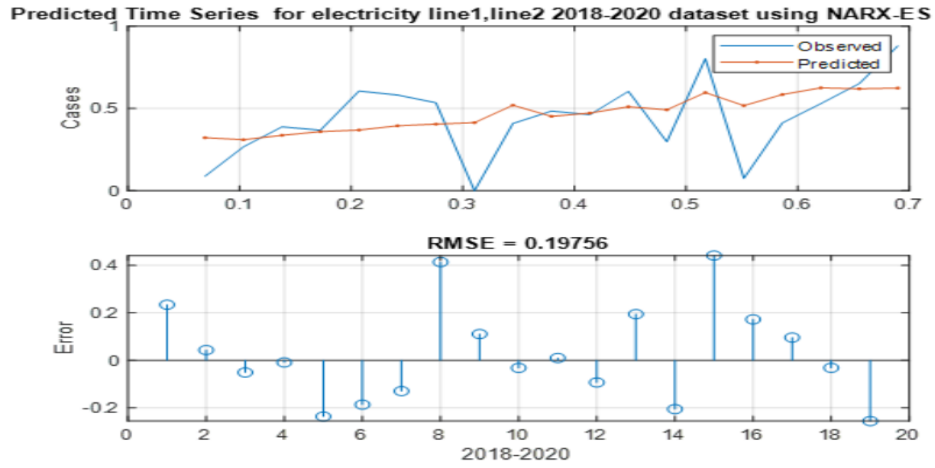


Figure 4.15: The Prediction Result of the Training Process for One Month using NARX-OA K Mean clustering

Figure 4.15 shows the observed and predicted outputs deviate and do not match, which means that the NARX-OA K Mean clustering model is weak for one-month forecasting.

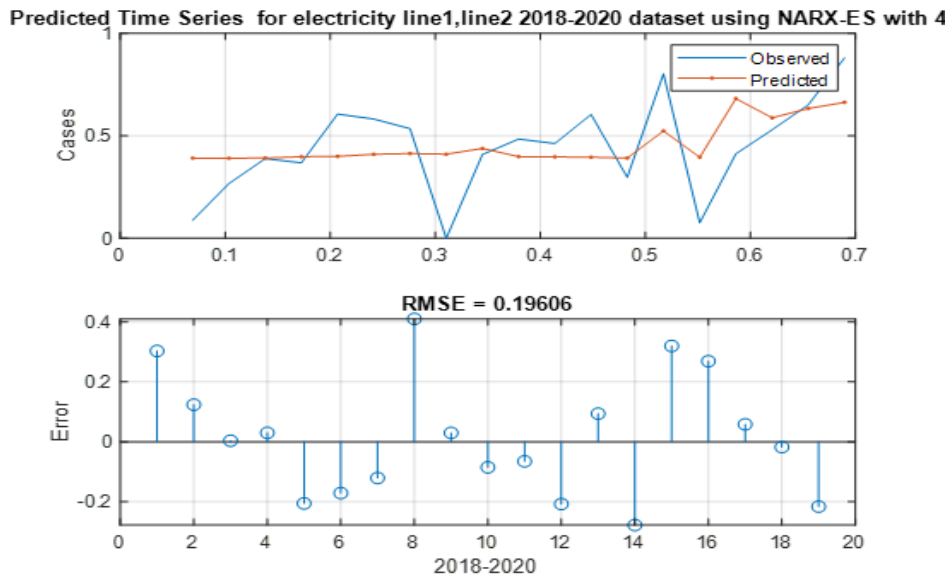


Figure 4.16 Prediction Result of the Training Process for One Month using NARX-OA

Figure 4.16 shows the observed and predicted outputs deviate and do not match, which means that the NARX-OA model is weak for one-month forecasting.

- **Summer Season Prediction:**

| Neurons | NARX-OA       |              | NARX_OA-Kmean clustering |              |
|---------|---------------|--------------|--------------------------|--------------|
|         | RMSE Training | RMSE Testing | RMSE Training            | RMSE Testing |
| 2       | 0.14883       | 0.1261       | 0.1024                   | 0.14748      |
| 4       | 0.14579       | 0.20279      | 0.09150                  | 0.10478      |
| 6       | 0.15044       | 0.16174      | 0.10455                  | 0.14784      |
| 8       | 0.14956       | 0.11283      | 0.097365                 | 0.16141      |
| 10      | 0.1455        | 0.1727       | 0.10921                  | 0.16073      |
| 12      | 0.15620       | 0.1822       | 0.11520                  | 0.16447      |
| 14      | 0.15942       | 0.2039       | 0.11528                  | 0.15246      |

Table 4.18 Model Training Result for Energy Prediction (Summer Season).

The result acquired for the second set experiment (summer season forecasting) explained the best performance recorded by the NARX-OA- K Mean clustering model as illustrated in table 4.18. In most cases, the performance of the NARX-OA-K Mean clustering model was better than the performance of other hybrid models as is shown in figure 4.18.

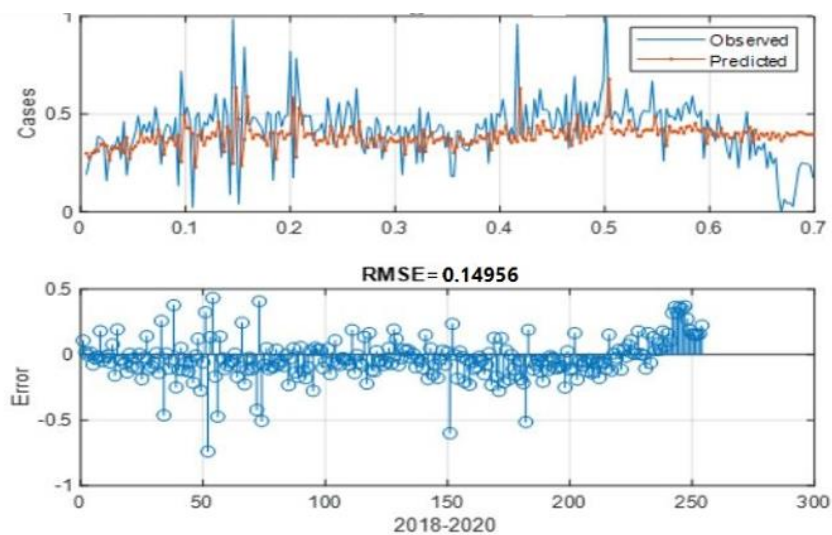


Figure 4.17 Prediction Result of the Training Process for One Season using NARX-OA

Figure 4.17 shows the observed and predicted outputs for one season forecasting applied by the NARX-OA model, with RMSE training = 0.14956.

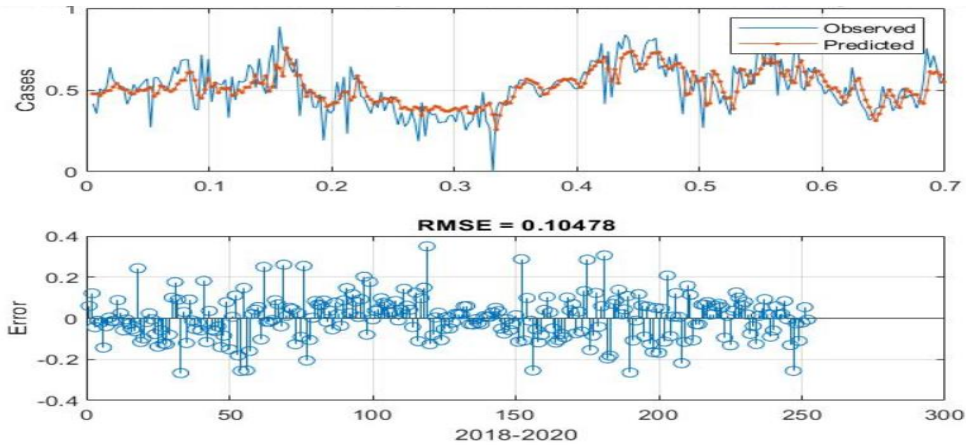


Figure 4.18 Prediction Result of the Training Process for Summer Season using NARX-OA-K Mean clustering model

Figure 4.18 shows the observed and predicted outputs for one season forecasting applied by the NARX-OA-K Mean clustering model, with RMSE training = 0.10478.

- **Winter Season Prediction:**

| Neurons | NARX-OA       |              | NARX_OA-Kmean clustering |              |
|---------|---------------|--------------|--------------------------|--------------|
|         | RMSE Training | RMSE Testing | RMSE Training            | RMSE Testing |
| 2       | 0.13769       | 0.2161       | 0.14421                  | 0.20187      |
| 4       | 0.13277       | 0.18998      | 0.17667                  | 0.17902      |
| 6       | 0.12786       | 0.29403      | 0.14061                  | 0.27492      |
| 8       | 0.12929       | 0.19716      | 0.1733                   | 0.20999      |
| 10      | 0.14818       | 0.24739      | 0.13461                  | 0.13029      |
| 12      | 0.14934       | 0.47915      | 0.17058                  | 0.22046      |
| 14      | 0.13726       | 0.38752      | 0.14411                  | 0.33997      |

Table 4.19: Model Training Result for Energy Prediction (Winter Season)

As can be seen from Table 4.19, the result of the third set experiment showed the lowest RMSE testing from 2 neurons until 14 neurons recorded by the NARX-OA-K Mean clustering model. This result indicates the efficiency of the NARX-OA-K Mean clustering model outperformed other hybrid models, which give the best one-season forecasting. As shown in Figure (4.19) which approves the accuracy of our model with the match of the predicate and target curves when executing 10 neurons.

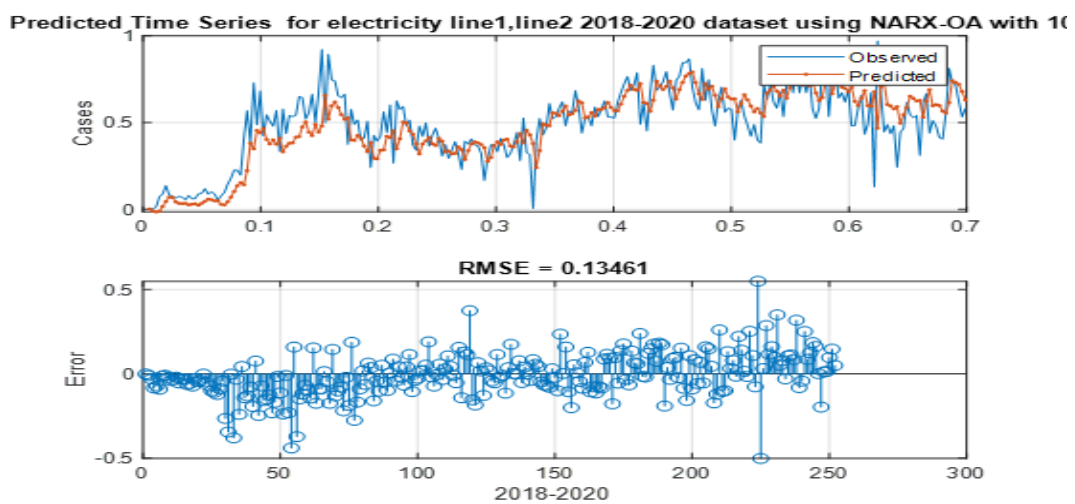


Figure 4.19: Prediction Result of the Training Process for One season (Winter Season) using NARX-OA-K Mean Clustering.

Figure 4.19 shows the observed and predicted outputs for one season forecasting applied by the NARX-OA-K Mean Clustering model, with RMSE training and testing (RMSE training=0.13461, RMSE Testing=0.13029).

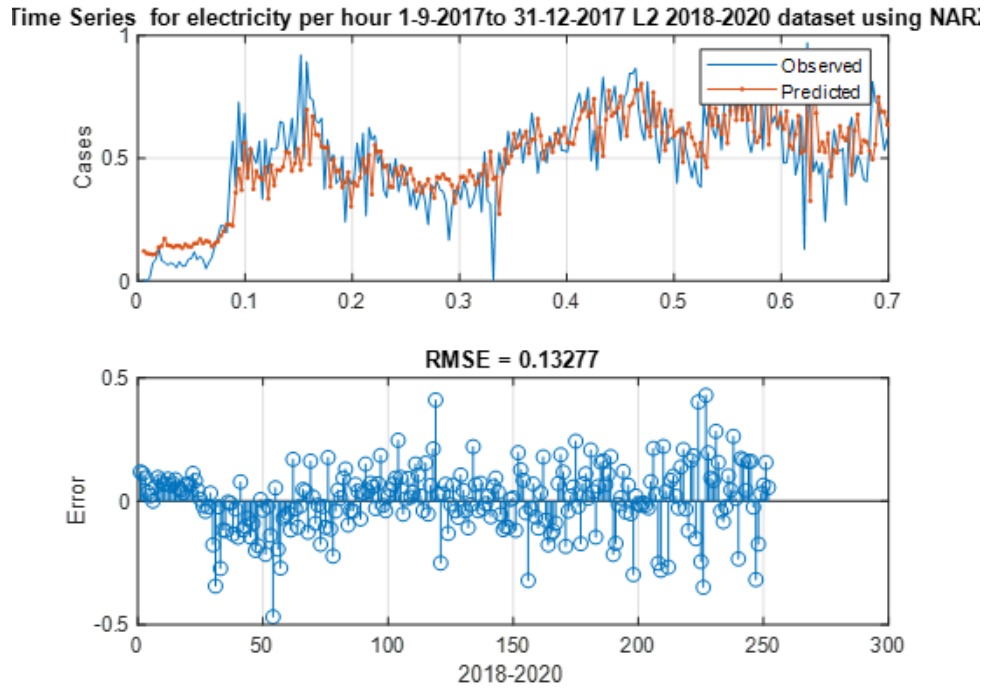


Figure 4.20: Prediction Result of the Training Process for One Season(Winter Season) using NARX-OA

Figure 4.20 shows the observed and predicted outputs for one season forecasting applied by the NARX-OA model, with RMSE training and testing (RMSE training=0.13277, RMSE Testing=0.18998).

- **One Year Prediction:**

| Neurons | NARX-OA       |              | NARX_OA-Kmean clustering |              |
|---------|---------------|--------------|--------------------------|--------------|
|         | RMSE Training | RMSE Testing | RMSE Training            | RMSE Testing |
| 2       | 0.042197      | 0.24472      | 0.047243                 | 0.23456      |
| 4       | 0.04455       | 0.12692      | 0.055896                 | 0.10048      |
| 6       | 0.044662      | 0.092527     | 0.043563                 | 0.090827     |
| 8       | 0.046277      | 0.10382      | 0.042951                 | 0.087569     |
| 10      | 0.04298       | 0.12437      | 0.063531                 | 0.091023     |
| 12      | 0.043312      | 0.11594      | 0.056738                 | 0.089953     |

|    |         |         |          |          |
|----|---------|---------|----------|----------|
| 14 | 0.04879 | 0.19162 | 0.582315 | 0.092156 |
|----|---------|---------|----------|----------|

Table 4.20: Model Training Result for Energy Prediction

According to table 4.20, when testing the fourth set experiment (one-year forecasting) with different hidden units started from 2 to 14 neurons, we observe that the result still indicates the efficiency of the NARX-OA-K Mean clustering model with the lowest RMSE testing and training (0.042951, 0.087569) at hidden layer equal 8. When using an improved approach with NARX-OA, we obtain better solutions along with all experiments as illustrated in table 4.20. As observed from the last three executions, adding additional neurons does not usually lead to a better error rate due to some factors like data quality, instability of NN learning, and over-fitting.

Figure (4.21) shows that the predicted and target curves are more convergent with RMSE testing and training (RMSE testing = 0.087569, RMSE Training = 0.042951) when the one-year time series was applied using NARX-OA-K Mean clustering compared with figure (4.22) where the time series dataset was conducted using NARX-OA with RMSE testing and training (RMSE testing = 0.092527, RMSE training = 0.044662).

Predicted Time Series for electricity line1,line2 2018-2020 dataset using NARX-ES with 11

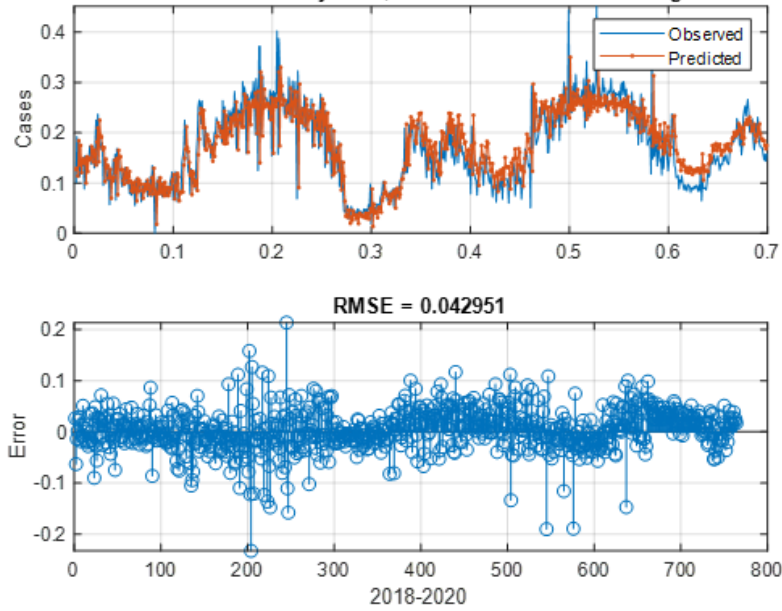


Figure 4.21 Prediction Result of Training Process for One Year using NARX-OA-K Mean Clustering

Figure 4.21 shows the observed and predicted outputs for one season forecasting applied by the NARX-OA-K Mean Clustering model, with RMSE training and testing (RMSE training =0.042951, RMSE Testing =0.087569).

Predicted Time Series for electricity line1,line2 2018-2020 dataset using NARX-ES with 6

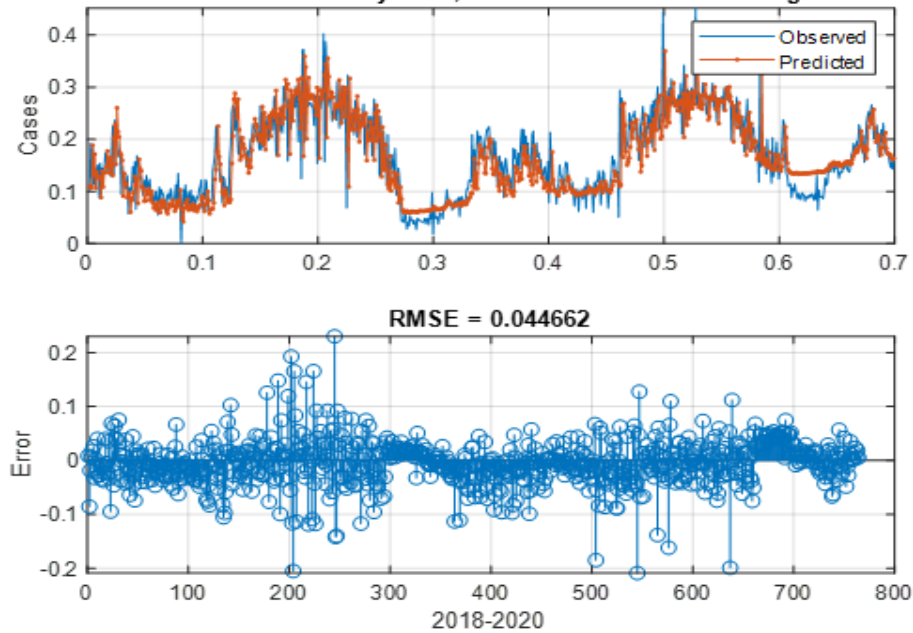


Figure 4.22: Prediction Result of the Training Process for One Year using NARX-OA

Figure 4.22 shows the observed and predicted outputs for one season forecasting applied by the NARX-OA model, with RMSE training and testing (RMSE training=0.044662, RMSE Testing=0.092527).

#### 4.3.4 Current Intensity prediction

The results from this set of the experiment still prove that the best performance is also obtained by the NARX-OA-K Mean Clustering model. In all test cases, the performance of NARX-OA-K Mean Clustering was better than other hybrid models (RNN-OA, NARX-OA) with RMSE testing and training at 6 hidden units (RMSE testing = 0.20779, RMSE training = 0.074283) as shown in table 4.21 and figure 4.23, the performance of the NARX-OA and RNN-OA are shown in figure 4.24 and 4.25.

| Hybrid models<br>Performance<br>Neurons | NARX-OA       |              | NARX-OA-K Mean Clustering |              |
|-----------------------------------------|---------------|--------------|---------------------------|--------------|
|                                         | RMSE Training | RMSE Testing | RMSE Training             | RMSE Testing |
| 2                                       | 0.074444      | 0.45193      | 0.10177                   | 0.27892      |
| 4                                       | 0.095092      | 0.38194      | 0.13284                   | 0.30484      |
| 6                                       | 0.091255      | 0.26773      | 0.074283                  | 0.20779      |
| 8                                       | 0.075519      | 0.30913      | 0.11018                   | 0.24197      |
| 10                                      | 0.076869      | 0.46141      | 0.11479                   | 0.28025      |
| 12                                      | 0.096818      | 0.27086      | 0.11035                   | 0.25384      |
| 14                                      | 0.095212      | 0.29215      | 0.10245                   | 0.24253      |

Table 4.21 Model Training Result for Current Intensity Prediction (Large Data)

According to table 4.21, when testing the fifth set experiment (Current Intensity Forecasting) with different hidden units started from 2 to 14 neurons, we observe that the result still indicates the efficiency of the NARX-OA-K Mean clustering model with the lowest RMSE testing and training(0.074283, 0.20779) at hidden layer equal 6.

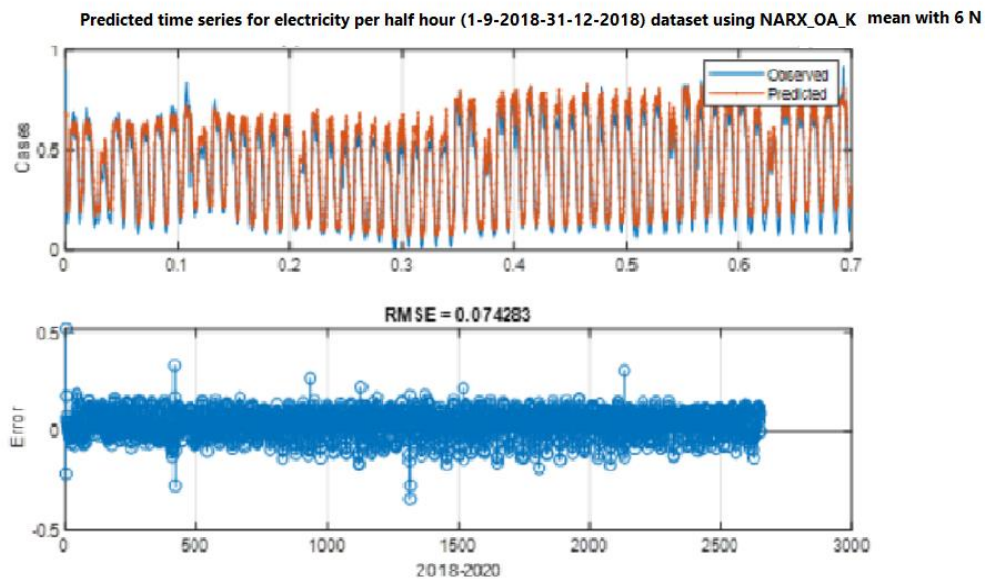


Figure 4.23 Prediction Result of Training Process for Current Intensity using NARX-OA-K Mean Clustering

Figure 4.23 shows that the predicted and target curves are more convergent with RMSE training RMSE training = 0.074283 when the current intensity time series was applied using NARX-OA-K Mean clustering compared with figure 4.24 where the time series dataset was conducted using NARX-OA with RMSE testing and training RMSE training = 0.091255.

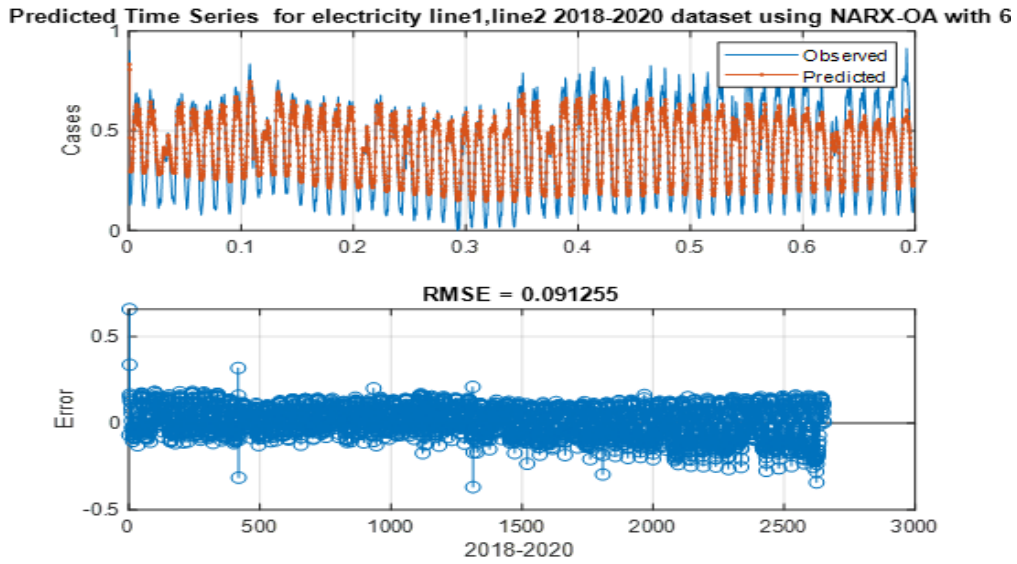


Figure 4.24 Prediction Result of the Training Process for Current Intensity using NARX-OA

Figure 4.24 shows the observed and predicted outputs for current intensity forecasting applied by the NARX-OA model, with RMSE training=0.092155.

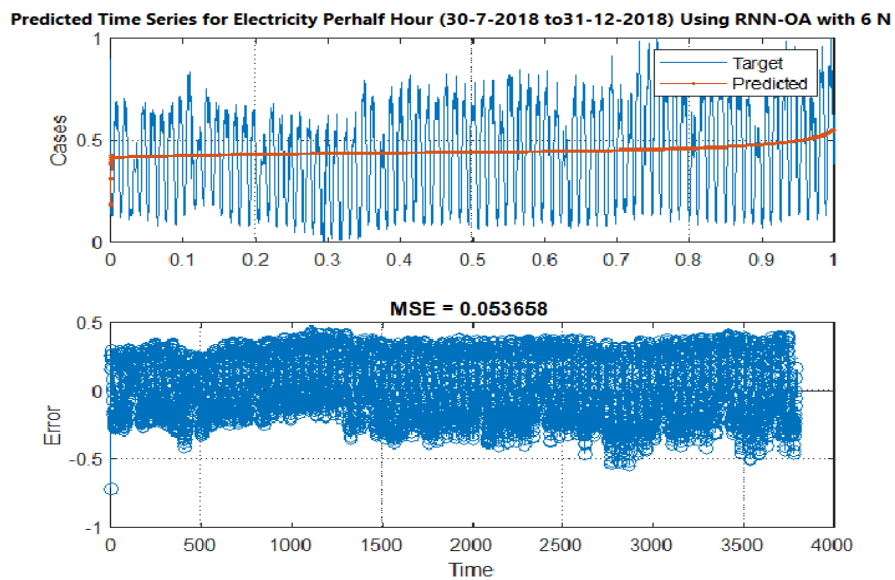


Figure 4.25 Prediction Result of the Training Process for Current Intensity using RNN-OA

Figure 4.25 shows the observed and predicted outputs deviate and do not match, which means that the RNN\_OA model is weak for current intensity forecasting.

#### 4.4 Trial Data Test

We also tested our modified Hybrid models through an applied new dataset (solar dataset) with 5 neurons, Population size=240, Generation=50, Mutation and Crossover probabilities (0.1,0.8) as shown in figures 4.26 and 4.27. Where we acquired the lowest RMSE testing and training (RMSE testing = 0.09053, RMSE training = 0.041923) using the NARX-OA-K Mean Cluster approach.

As we can see in the figures (4.26, 4.27, and 4.28), we applied our three proposed hybrid models (RNN-OA, NARX-OA, and NARX-OA- K-mean cluster model), we concluded that NARX-OA- K-mean cluster model gives more accurate results than the other two models that couldn't provide clear or valid results nor predict the electricity consumption, thus we cannot rely on those models, whereas in NARX-OA- K-mean cluster model the real and predicted data are matching and gives accurate results for large amounts of data (> 100 samples).

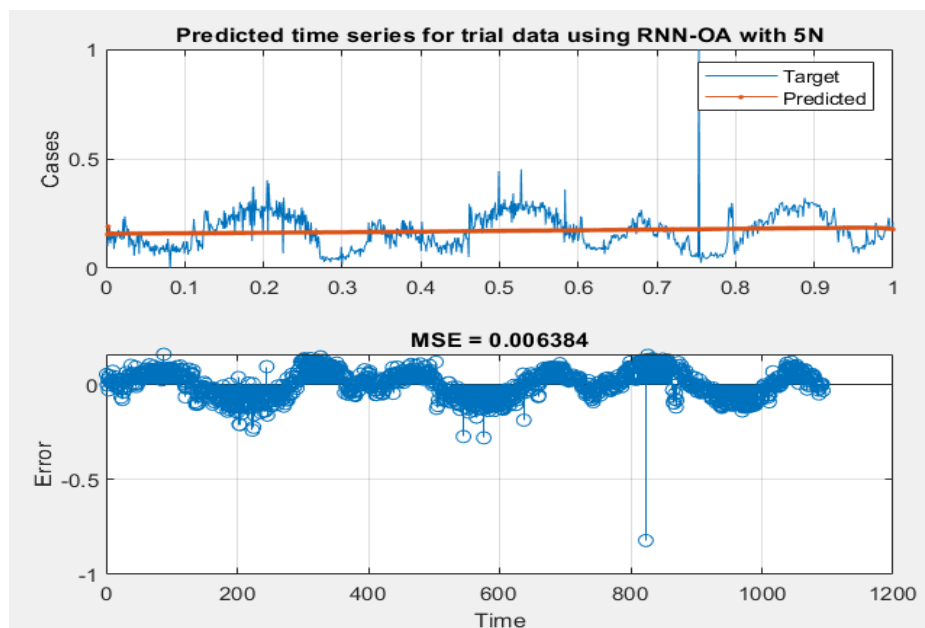


Figure 4.26 Prediction Result of Trial Dataset Training Process using Modified Model (RNN-OA)

As illustrated in Figure 4.26, The real and predicted values of the trial data aren't identical to each other. This indicates also that the RNN-OA model isn't an effective method that can be used in forecasting. Thus, using this method in prediction will predict an accurate result.

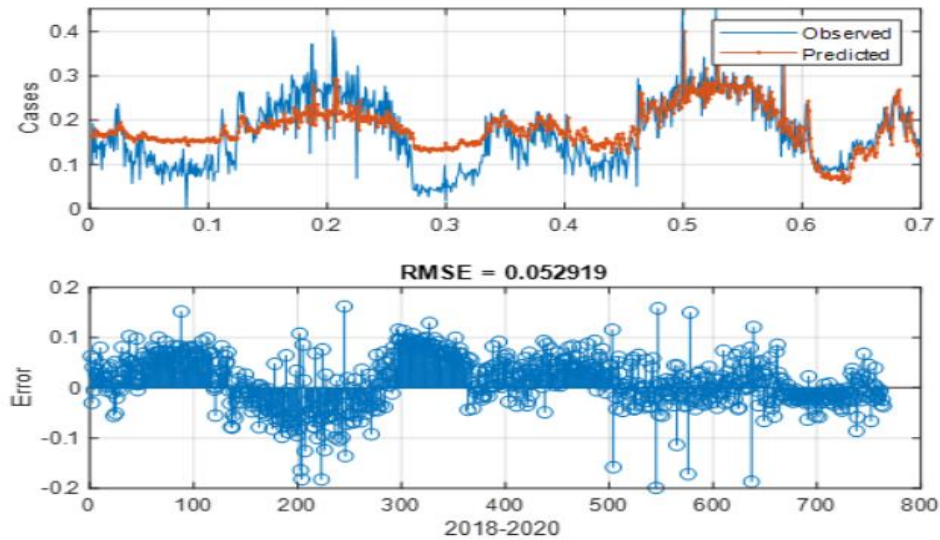


Figure 4.27 Prediction Result of the Training Process for Trial Dataset using NARX-OA

Figure 4.27 shows the observed and predicted outputs of the trial data do not match each other. These results can't be taken into consideration because using this method in prediction will predict wrong results.

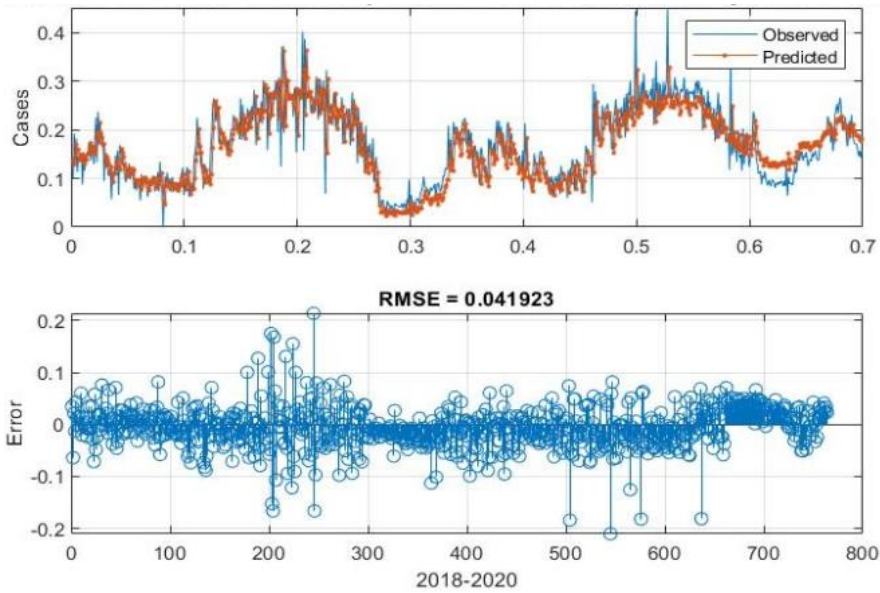


Figure 4.28 The Prediction Result of the Training Process for Trial Dataset using Modified Model  
(NARX-OA- K-mean cluster)

As illustrated in figure 4.28, the observed and predicted values of trial data are very close to each other. The NARX-OA- K-mean cluster model is an effective and strong model that can be used in time series forecasting.

#### 4.4 Comparison Between Conventional and Hybrid Models:

We took the average consumption from 2018 to 2020 for our proposed model using different enhanced techniques. As the dataset of electricity consumption is large and constantly changing, we processed this data by taking samples. Each processed data point is defined as the average of the data point added with its two successor points. We conducted four executions to train the processed and non-processed data. The results turned out to be better with processed data as shown in Table 4.22.

| Technique | Neurons | Train Data | Test Data | Validation Data | Mse (Winter Season) |
|-----------|---------|------------|-----------|-----------------|---------------------|
|-----------|---------|------------|-----------|-----------------|---------------------|

|                                       |    |     |     |     |            |
|---------------------------------------|----|-----|-----|-----|------------|
| RNN<br>With<br>Processed<br>data      | 7  | 70% | 15% | 15% | 0.0017007  |
|                                       | 13 | 70% | 15% | 15% | 0.0011271  |
|                                       | 15 | 70% | 15% | 15% | 0.00102625 |
|                                       | 30 | 70% | 15% | 15% | 0.00102155 |
| RNN<br>with non-<br>processed<br>data | 7  | 70% | 15% | 15% | 0.0044146  |
|                                       | 13 | 70% | 15% | 15% | 0.0035447  |
|                                       | 15 | 70% | 15% | 15% | 0.0050868  |
|                                       | 30 | 70% | 15% | 15% | 0.0041206  |

Table 4.22: Results of the Training Model using RNN

As shown in figure 4.29, the results of prediction using RNN enhanced with genetics (neuro genetic) were similar to the results when using RNN alone, because short-term data gives the same results even when using the enhanced RNN. This is due to the feedback between hidden neurons that memorizes the significant events for all the nodes where the impact of each time step is the same on the output.

On the other hand, when using large data (1500 data points taken every 30 minutes during July), the enhanced RNN with genetics shows high performance and accuracy with  $MSE = 7.9 \cdot 10^{-3}$  as shown in figure (4.29). However, when using the same large data with RNN the prediction was poor with  $MSE = 1 \cdot 10^{-2}$  as shown in figure (4.30) where the predicted and target curves are not matching.

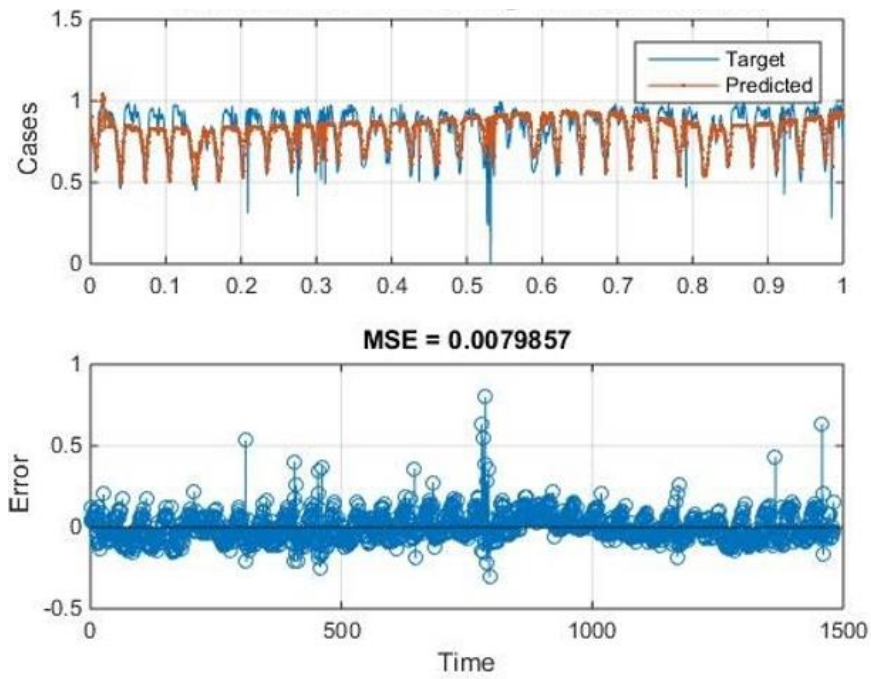


Figure 4.29 Large Data Prediction using RNN-OA Model

Figure 4.29 shows the observed and predicted outputs for current intensity forecasting (1500 data points taken every 30 minutes during July) applied by the RNN-OA model, with RMSE training=0.00798.

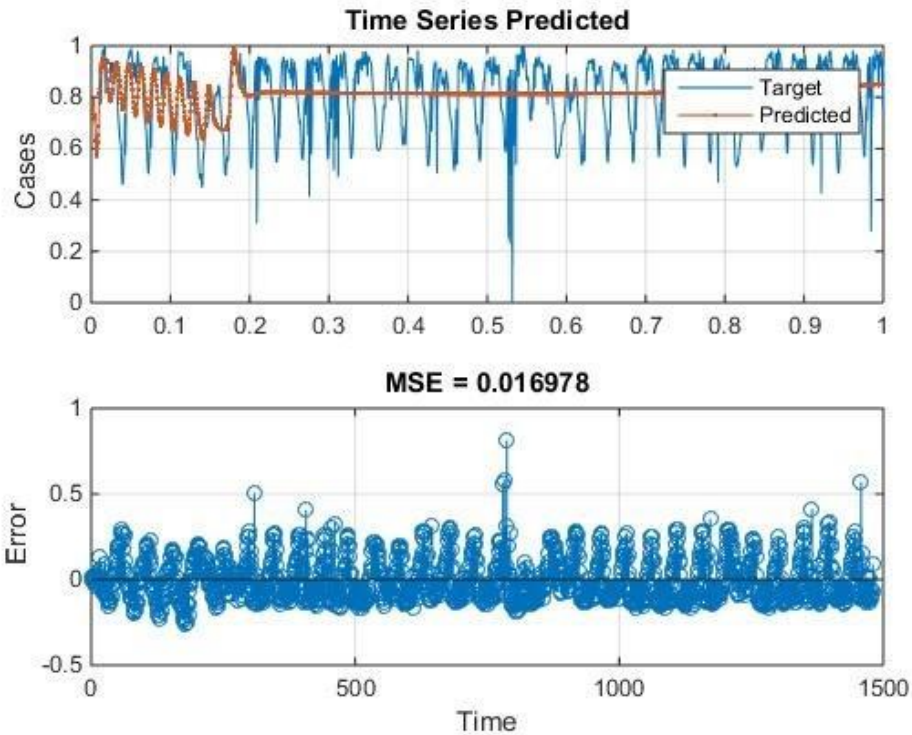


Figure 4.30 Large Data Prediction using RNN

Figure 4.30 shows the observed and predicted outputs for current intensity forecasting (1500 data points taken every 30 minutes during July) applied by the RNN model, with RMSE training=0.016978. The real and predicted values do not match each other. These results indicate that the RNN model is not effective with large data and can't be taken into consideration because using this method in prediction will predict wrong results.

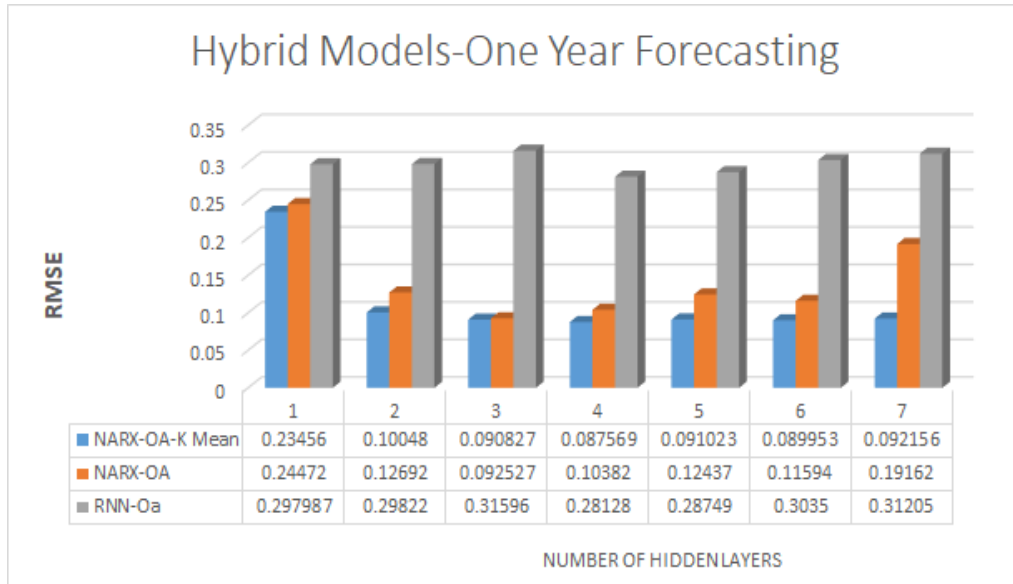


Figure 4.31 Comparison Between Hybrid Models for One-Year Forecasting

As illustrated in figure 4.31 when comparing between hybrid models based on RMSE tasting for 2 - 14 hidden layers, where NARX-OA-Kmean model showed the minimum and best values and results were more precise than the other NARX-OA and RNN-OA models. Thus, using the NARX-OA-Kmean model in prediction will give Tulkarm municipality a high knowledge with the amount of the electricity expected workload, which will help Tulkarm municipality to be able to manipulate and process the high electricity workload by effective ways, hence, it will ensure providing high-quality service for people, residential and commercial buildings, and Tulkarm municipality.

## 4.6 Challenges and Limitations

We encountered several challenges and obstacles during the collection of data used in the model such as:

1. The data collection process took place in the early period of the COVID-19 pandemic, thus making communication with the municipality hard.
2. Strict municipality policy regarding giving away any information.
3. The required data for the load forecasting algorithm was weakly reserved in the municipality as most of the data was collected from paper works.

# Chapter 5

## Conclusion and Future Work

## 5. Conclusion and Future Work

### 5.1 Conclusion

Electricity consumption prediction has a great part in the research and development of the energy industry, as the precise prediction of the future energy supply and demand has an important role in the sustainable economy of our country. However, it is not easy to adapt a specific or appropriate forecasting model, as many forecasting methods were developed, none can be generalized for all demand patterns. Therefore, artificial intelligence techniques have gained importance in reducing estimation errors and gave better results than conventional models.

This study proposes a model to predict the future electricity demand, by connecting time, place, and historical data. Palestinian society aims to reduce the dependencies the usage energy of electricity from the other side “Israel's occupation” so they turned to use solar cells energy and create models which provide the power companies and municipality the future information about electricity consumption. To organize their distributions in a computerized and organized manner to maintain stability in the delivery and distribution of electricity.

In this study, we presented three hybrid models to predict electricity consumption prediction for data samples from 2018 to 2020 in Palestine, through the Levenberg-Marquardt algorithm with RNN back-propagation to predict the monthly, seasonal, and yearly electricity consumption with the best accuracy.

This study implemented hybrid systems that depend on using the artificial neural networks (RNN-OA, NARX-OA, and NARX-OA-K mean cluster) integrated with the optimization algorithm (OA). Then, each model is separately applied, and depending on the RMSE values

we decide the best-proposed hybrid model that produces the best results for electricity prediction.

We found the RNN performs at its best at short-term forecasts (in our case the short term is up to 100 samples). Thus we combined RNN with genetics for long-term forecasts to achieve better results. Our proposed work gives electricity companies accurate information about future electricity consumption prediction so that it would help with developing work plans and the distribution of electricity loads along with appropriate computerized and structured guidelines.

The RNN and NARX Hybrid model were combined with a modified genetic algorithm, then we enhanced the NARX Hybrid model using the K Mean Clustering approach for the two phases of the modified genetic algorithm which are (Initialization step and arithmetic crossover).

## **5.2 Recommendations**

### **(a) For Tulkarm municipality:**

- Improve and upgrade the electrical network to satisfy the expected electricity demand in the future (electricity loaded forecasting).
- Generating more electricity using energy cells.
- Develop work plans and the distribution of electricity loaded

### **(b) For Palestinian Energy and Natural Resources Authority (PENRA)**

Working on uniting all the residential buildings, companies, commercial buildings, governmental establishments, etc. of Palestine within the electricity distribution companies and municipalities to make all the information of the electricity consumption available for any future planning process.

### **5.3 Future work**

Regarding future work, the RNN model can be developed by combining RNN with a Particle swarm optimization algorithm and a Neuro-Fuzzy system to enhance the performance of the researcher's model. In addition, many alterations to the different parameters of genetic algorithms can be done to enhance the performance of the model. Moreover, using a classification approach to be applied on electricity meter consumption to find electricity consumption for every region and district of the city. And also make this study to other electricity distribution companies, municipalities, and Palestine in general.

## References

- [1] Mocanu E, Nguyen PH, Gibescu M, Kling WL. (2016). Deep learning for estimating building energy consumption. *Sustain Energy Grids Networks* ;6:91–9
- [2] Rahman, A., Srikumar, V., & Smith, A. D. (2018). Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy*, 212, 372–385.
- [3] Hochreiter S, Schmidhuber J. (1997). Long short-term memory. *Neural Comput* ;9:1735–80.
- [4] Hamzaçebi, C., Es, H. A., & Çakmak, R. (2017). Forecasting of Turkey's monthly electricity demand by seasonal artificial neural network. *Neural Computing and Applications*.
- [5] Khwaja, A. S., Zhang, X., Anpalagan, A., & Venkatesh, B. (2017). Boosted neural networks for improved short-term electric load forecasting. *Electric Power Systems Research*, 143, 431–437.
- [6] Salahat, S., & Awad, M. (2017). Short-Term Forecasting of Electricity Consumption in Palestine Using Artificial Neural Networks. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 8(2), 11-21.
- [7] H. Gavin (2009), "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems, Department of Civil and Environmental Engineering Duke University.
- [8] Qasrawi, I., & Awad, M. (2015). Prediction of the power output of solar cells using neural networks: solar cells energy sector in Palestine. *Int J Comput Sci Secur IJCSS*, 9(6), 280.
- [9] Biswas, M. R., Robinson, M. D., & Fumo, N. (2016). Prediction of residential building energy consumption: A neural network approach. *Energy*, 117, 84-92.
- [10] Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016). Recurrent neural network grammars. *arXiv preprint arXiv:1602.07776*.
- [11] Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction.
- [12] Ugurlu, U., Oksuz, I., & Tas, O. (2018). Electricity price forecasting using recurrent neural networks. *Energies*, 11(5), 1255.

- [13] Chung, H., & Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, 10(10), 3765.
- [14] Heghedus, C., Segarra, S., Chakravorty, A. and Rong, C., 2019, July. Neural Network Architectures for Electricity Consumption Forecasting. In the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 776-783). IEEE.
- [15] Salami, M., Movahedi Sobhani, F. and Ghazizadeh, M.S., 2018. Short-term forecasting of electricity supply and demand by using the Wavelet-PSO-NNs-SO technique for searching in big data of Iran's Electricity Market. *Data*, 3(4), p.43.
- [16] Bâra, A. and Oprea, S.V., 2017. Electricity consumption and generation forecasting with artificial neural networks. *Advanced Applications for Artificial Neural Networks*; El-Shahat, A., Ed, pp.119-141.
- [17] Amber, K.P., Ahmad, R., Aslam, M.W., Kousar, A., Usman, M. and Khan, M.S., 2018. Intelligent techniques for forecasting electricity consumption of buildings. *Energy*, 157, pp.886-893.
- [18] Juaidi, A., Montoya, F.G., Ibrik, I.H. and Manzano-Agugliaro, F., 2016. An overview of renewable energy potential in Palestine. *Renewable and Sustainable Energy Reviews*, 65, pp.943-960.
- [19] Hassanat, A.B., Prasath, V.B., Abbadi, M.A., Abu-Qdari, S.A. and Faris, H., 2018. An improved genetic algorithm with a new initialization mechanism based on regression techniques. *Information*, 9(7), p.167. ‘
- [20] Dong, H., Li, T., Ding, R., Sun, J. (2018). A novel hybrid genetic algorithm with granular information for feature selection and optimization, *Applied Soft Computing*, 65, 33-46.
- [21] Meshkinfam, S., Ghassemi Tari, F. (2016). A genetic algorithm for generalized tardiness flowshop problems, *International Journal of Engineering Science and Technology*, 8(9), 219-228
- [22] Uysal, E.İ., Demircioğlu, G., Kale, G., Bostancı, E., Güzel, M.S. and Mohammed, S.N., 2019, October. Network Anomaly Detection System using Genetic Algorithm, Feature Selection, and Classification. In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1-5). IEEE.
- [23] Murthy, C. A., and MACHINE INTELLIGENCE UNIT. "Genetic Algorithms: Basic principles and applications." In 2012 2nd National Conference on Computational Intelligence and Signal Processing (CISP), pp. 22-22. IEEE, 2012.

- [24] Palestine News & Information Agency (WAFA). 2019. "Solar Energy Initiative". [http://info.wafa.ps/ar\\_page.aspx?id=9080](http://info.wafa.ps/ar_page.aspx?id=9080)
- [25] Malik, H., Srivastava, S., Sood, Y.R. and Ahmad, A. eds., 2018. Applications of Artificial Intelligence Techniques in Engineering: SIGMA 2018, Volume 1 (Vol. 698). Springer.
- [26] Moghaddam, A.H., Moghaddam, M.H. and Esfandyari, M., 2016. Stock market index prediction using artificial neural networks. *Journal of Economics, Finance and Administrative Science*, 21(41), pp.89-93.
- [27] J. Ahire, (2018, Nov 11). The Artificial Neural Networks Handbook: Part 4 [Blog post]. Retrieved from <https://medium.com/@jayeshbahire/the-artificial-neuralnetworks-handbook-part-4-d2087d1f583e>
- [28] Rinchon, J.P.M., Concha, N.C. and Calilung, M.G.V., 2017, December. Reinforced concrete ultimate bond strength model using hybrid neural network-genetic algorithm. In 2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-6). IEEE.
- [29] Haghiri, S., Daghighi, A. and Moharramzadeh, S., 2018. Optimum coagulant forecasting by modeling jar test experiments using ANNs. *Drinking Water Engineering and Science*, 11(1), pp.1-8.
- [30] Gharehbaghi, K. and McManus, K., 2019. TIS condition monitoring using ANN integration: an overview. *Journal of Engineering, Design and Technology*.
- [31] Aghbashlo, M., Hosseinpour, S. and Mujumdar, A.S., 2015. Application of artificial neural networks (ANNs) in drying technology: a comprehensive review. *Drying technology*, 33(12), pp.1397-1462.
- [32] Feng, J. and Lu, S., 2019, June. Performance analysis of various activation functions in artificial neural networks. In *Journal of Physics: Conference Series* (Vol. 1237, No. 2, p. 022030). IOP Publishing.
- [33] Bansal, A., Sharma, M. and Goel, S., 2017. Improved K-mean clustering algorithm for prediction analysis using classification technique in data mining. *International Journal of Computer Applications*, 157(6), pp.0975-8887.
- [34] Gupta, D., 2017. Fundamentals of deep learning–activation functions and when to use them. Dosegljivo:<https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activationfunctions-when-to-use-them>.

- [35] Tari, F.G. and Hashemi, Z., 2018. Prioritized K-mean clustering hybrid GA for discounted fixed charge transportation problems. *Computers & Industrial Engineering*, 126, pp.63-74.
- [36] Naeem, S. and Wumaier, A., 2018. Study and implementing K-mean clustering algorithm on English text and techniques to find the optimal value of K. *Int. J. Comput. Appl*, 182(31), pp.7-14.
- [37] Louzazni, M., Mosalam, H., Khouya, A. and Amechnoue, K., 2020. A non-linear auto-regressive exogenous method to forecast the photovoltaic power output. *Sustainable Energy Technologies and Assessments*, 38, p.100670.
- [38] Wei, M., Ye, M., Li, J.B., Wang, Q. and Xu, X., 2020. State of Charge Estimation of Lithium-Ion Batteries Using LSTM and NARX Neural Networks. *IEEE Access*, 8, pp.189236-189245.
- [39] Rahimi, Z., Shafri, H.Z.M. and Norman, M., 2018. A GNSS-based weather forecasting approach using Nonlinear Auto Regressive Approach with Exogenous Input (NARX). *Journal of Atmospheric and Solar-Terrestrial Physics*, 178, pp.74-84.
- [40] Boussaada, Z., Curea, O., Remaci, A., Camblong, H. and Mrabet Bellaaj, N., 2018. A nonlinear autoregressive exogenous (NARX) neural network model for the prediction of the daily direct solar radiation. *Energies*, 11(3), p.620.
- [41] Rahimi, Z., Shafri, H.Z.M. and Norman, M., 2018. A GNSS-based weather forecasting approach using Nonlinear Auto Regressive Approach with Exogenous Input (NARX). *Journal of Atmospheric and Solar-Terrestrial Physics*, 178, pp.74-84.
- [42] Banihabib, M.E. and Mousavi-Mirkalaei, P., 2019. Extended linear and non-linear auto-regressive models for forecasting the urban water consumption of a fast-growing city in an arid region. *Sustainable Cities and Society*, 48, p.101585.
- [43] Naderpour, H., Rafiean, A.H. and Fakharian, P., 2018. Compressive strength prediction of environmentally friendly concrete using artificial neural networks. *Journal of Building Engineering*, 16, pp.213-219.
- [44] Juaidi, Adel; Montoya, Francisco G.; Ibrik, Imad H.; Manzano-Agugliaro, Francisco (2016). *An overview of renewable energy potential in Palestine. Renewable and Sustainable Energy Reviews*, 65(), 943–960. doi:10.1016/j.rser.2016.07.052
- [45] Fathi Nassar, Yasser; Yassin Alsadi, Samer (2019). *Assessment of solar energy potential in Gaza Strip-Palestine. Sustainable Energy Technologies and Assessments*, 31(), 318–328. doi:10.1016/j.seta.2018.12.010

- [46] A. Azadeh; S.F. Ghaderi; S. Sohrabkhani (2008). *Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors.* , 49(8), 2272–2278. doi:10.1016/j.enconman.2008.01.035
- [47] Tang, Lei; Wang, Xifan; Wang, Xiuli; Shao, Chengcheng; Liu, Shiyu; Tian, Shijun (2018). *Long-term Electricity Consumption Forecasting Based on Expert Prediction and Fuzzy Bayesian Theory.* *Energy*, (), S0360544218320632–. doi:10.1016/j.energy.2018.10.073
- [48] Vu, Nguyen Hoang Minh; Khanh, Nguyen Truong Phuc; Cuong, Vo Viet; Binh, Phan Thi Thanh (2017). [IEEE 2017 International Conference on Electrical Engineering and Informatics (ICELTICS) - Banda Aceh (2017.10.18-2017.10.20)] 2017 International Conference on Electrical Engineering and Informatics (ICELTICS) - Forecast on vietnam electricity consumption to 2030. , (), 72–77. doi:10.1109/ICELTICS.2017.8253238
- [49] Zeid, M. and Thawaba, S., 2018. Planning under a colonial regime in Palestine: Counter planning/decolonizing the West Bank. *Land Use Policy*, 71, pp.11-23.
- [50] Marei, Ibrahim (2017). Developments in law and policy: the promotion of green energy in the electricity sector of Palestine. *Journal of Energy & Natural Resources Law*, 35(1), 47–67. doi:10.1080/02646811.2016.1216698
- [51] Hamzaçebi, C., Es, H.A. and Çakmak, R., 2019. Forecasting of Turkey’s monthly electricity demand by seasonal artificial neural network. *Neural Computing and Applications*, 31(7), pp.2217-2231.
- [52] Ozoh, P., Abd-Rahman, S., Labadin, J. and Apperley, M., 2014. A comparative analysis of techniques for forecasting electricity consumption. *International journal of computer applications*, 88(15).
- [53] Bianco V, Manca O, Nardini S (2009) Electricity consumption forecasting in Italy using linear regression models. *Energy* 34(9):1413–1421
- [54] Azadeh A, Saberi M, Seraj O (2010) An integrated fuzzy regression algorithm for energy consumption estimation with non-stationary data: a case study of Iran. *Energy* 35(6):2351–2366
- [55] A. Dedinec, S. Filiposka, A. Dedinec, L. Kocarev, (2016) “Deep belief network based electricity load forecasting: An analysis of Macedonian case,” *Energy*, vol. 115, pp. 1688-1700.
- [56] D. BRITZ, (2015). “Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs” From url: <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnn>

- [57] Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia computer science*, 132, 1351-1362.
- [58] Meena, D.C. and Devanshu, A., 2017, January. Genetic algorithm tuned PID controller for process control. In *2017 International Conference on Inventive Systems and Control (ICISC)* (pp. 1-6). IEEE.
- [59] Puig-Arnavat, M., & Bruno, J. C. (2015). Artificial neural networks for thermochemical conversion of biomass. In *Recent Advances in Thermo-Chemical Conversion of Biomass* (pp. 133-156). Elsevier.
- [60] Xu, H., Ma, C., Lian, J., Xu, K. and Chaima, E., 2018. Urban flooding risk assessment based on an integrated k-means cluster algorithm and improved entropy weight method in the region of Haikou, China. *Journal of hydrology*, 563, pp.975-986.
- [61] Alizadeh, M.J., Shahheydari, H., Kavianpour, M.R., Shamloo, H. and Barati, R., 2017. Prediction of longitudinal dispersion coefficient in natural rivers using a cluster-based Bayesian network. *Environmental Earth Sciences*, 76(2), p.86.
- [62] Lamini, C., Benhlima, S. and Elbekri, A., 2018. Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Computer Science*, 127, pp.180-189.
- [63] Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y. and Gao, Z., 2017. A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*, 256, pp.56-62.
- [64] Meena, D.C. and Devanshu, A., 2017, January. Genetic algorithm tuned PID controller for process control. In *2017 International Conference on Inventive Systems and Control (ICISC)* (pp. 1-6). IEEE.
- [65] Furqan, M., Hartono, H., Ongko, E. and Ikhsan, M., 2017. Performance of arithmetic crossover and heuristic crossover in genetic algorithms based on alpha parameters. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 19(1), pp.31-36.
- [66] Yang, W., Wang, J., Niu, T. and Du, P., 2020. A novel system for multi-step electricity price forecasting for electricity market management. *Applied Soft Computing*, 88, p.106029.
- [67] Trotter, P.A., McManus, M.C. and Maconachie, R., 2017. Electricity planning and implementation in sub-Saharan Africa: A systematic review. *Renewable and Sustainable Energy Reviews*, 74, pp.1189-1209.
- [68] Deb, C., Zhang, F., Yang, J., Lee, S.E. and Shah, K.W., 2017. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74, pp.902-924.

- [69] Mahmoud, M. M., and I. H. Ibrik. 2006. "Techno-economic Feasibility of Energy Supply of Remote Villages in Palestine by PV-Systems, Diesel Generators and Electric Grid." *Renewable and Sustainable Energy Reviews* 10 (2): 128–138.
- [70] Al Qadi, S., B. Sodagar, and A. Elnokaly. 2018. "Estimating the Heating Energy Consumption of the Residential Buildings in Hebron, Palestine." *Journal of Cleaner Production* 196: 1292–1305.
- [71] worlddata.info. 2019. ([www.worlddata.info/asia/palestine/index.php](http://www.worlddata.info/asia/palestine/index.php))
- [72] Tulkarm-Municipality. 2018. Report of Power Consumption During 2018. Tulkarm: Tulkarm-Municipality report for 2018.
- [73] Salah, Wael A.; Abuhelwa, Mai. 2020. *Energy status and practices for efficient energy management to reduce power interruptions: a case study on Tulkarm district in Palestine. International Journal of Sustainable Energy*, (), 1–15.
- [74] Fouquier, A., Robert, S., Suard, F., Stéphan, L. and Jay, A., 2013. State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews*, 23: 272-288.
- [75] Wei, N., Li, C., Peng, X., Li, Y. and Zeng, F., 2019c. Daily natural gas consumption forecasting via the application of a novel hybrid model. *Applied Energy*, 250: 358-368.

تعد الطاقة الكهربائية عاملاً حيوياً في تنمية و تطور الدول، حيث للطاقة الكهربائية أهمية كبيرة في حياتنا لما لها من مساهمة في المجال التكنولوجي و التعليمي و الصناعي و الطبي و كافة المجالات و لذلك اصبح الطلب على الطاقة الكهربائية في ازدياد مستمر. في هذه الأطروحة ، طبقنا عدة نماذج هجينة محسنة (RNN-OA, NARX-OA, NARX-OA-Kmean cluster) التي تدمج بين أنواع مختلفة من الشبكات العصبية NNs (وهي الشبكات العصبية المتكررة (RNNs) LSTM والشبكات العصبية الخارجية غير الخطية التلقائية (NARX)، مع خوارزمية تحسين ، تسمى إستراتيجية التطور (ES). هذه النماذج تم تطبيقها للتنبؤ بعبء استهلاك الطاقة الكهربائية ، والمشار إليه بـ line ١ و line ٢ ، وشدة التيار الكهربائي لمدينة طولكرم في فلسطين (AAUP). أظهرت النتائج التجريبية أن أداء النموذج الهجين (NARX-OA-Kmean cluster) كان أفضل من (RNN-OA) و (NARX-OA) في التنبؤ بعبء استهلاك الطاقة الكهربائية بشكل قصير المدى (يومي، شهري) و طويل المدى (فصلي و سنوي)، بينما نموذج NARX-OA كان أفضل من RNNs-OA في التنبؤ بعبء استهلاك الطاقة الكهربائية ، ونموذج RNNs كان أفضل من النماذج الأخرى في التنبؤ بعبء استهلاك الطاقة الكهربائية قصير المدى. هذه النتائج تم التعبير عنها بواسطة factor RMSE ، وذلك بعد ما تم تطبيق النماذج المقترحة على مجموعات البيانات المستخدمة (line 1&2 current intensity).