

Integration of logistic regression and multilayer perceptron for intelligent single and dual axis solar tracking systems

Nadia AL-Rousan¹  | Nor Ashidi Mat Isa²  |

Mohammad Khairunaz Mat Desa²  | Hazem AL-Najjar¹ 

¹Department of Computer Engineering, School of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey

²School of Electrical and Electronic Engineering, University of Sains Malaysia, Nibong Tebal, Pulau Pinang, Malaysia

Correspondence

Nadia AL-Rousan, Department of Computer Engineering, School of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey.
Email: nadia.rousan@yahoo.com

Abstract

Intelligent solar tracking systems to track the trajectory of the sun across the sky has been actively studied and proposed nowadays. Several low performance intelligent solar tracking systems have been designed and implemented. Multilayer perceptron (MLP) is one of the common controllers that used to drive solar tracking systems. However, when the input data are complex for neural network, neural network would not well explain the relationship between these data. Thus, it performed worse than when the input data are simple. Using a premapping of relationship between samples of data as input to neural network along with the original input data could probably a strong guide to help neural network to reach the desired goal and predict the output variables faster and more accurate. It is found that using the output of logistic regression as input to neural network would faster the process of finding the predicted output by neural network. Thus, this study aims to propose new efficient and low complexity single and dual axis solar tracking systems by integrating supervised logistic regression (LR) and supervised MLP or cascade multilayer perceptron (CMLP). LR models are trained by using one of unsupervised clustering techniques (k-means, fuzzy c-means, and hierarchical clustering algorithms). The

proposed models were used to predict both tilt and orientation angles by two different data sets (month, day, and time variables data set) and (month, day, time, Isc, Voc, and power radiation variables data sets). The results revealed that the proposed MLP/CMLP-LR systems are able to increase the prediction rate and decrease the mean square error rate as compared to conventional models in both single and dual axis solar tracking systems. The new developed intelligent systems achieved less number of overall connections, less number of neurons, and less time complexity. The finding suggests that the proposed intelligent solar tracking systems has a great potential to be applied for real-world applications (i.e., solar heating systems, solar lightening systems, factories, and solar powered ventilation).

KEYWORDS

complexity, intelligent systems, logistic regression, multilayer perceptron, neural network, single and dual axis

1 | INTRODUCTION

Positioning the solar photovoltaic modules toward the position of the sun across the sky is very important issue nowadays. Solar tracking system is a device that can be implemented to follow the trajectory of the sun across the sky efficiently. Solar tracking system is used to minimize the angle of incidence between the incoming solar beam and the photovoltaic module surface. Therefore, it helps in increasing the amount of generated energy by using solar photovoltaic modules. Several driving and controlling methods have been used in solar tracking systems to track the sun globally. One of the most efficient techniques that played an important role in increasing the efficiency of solar tracking systems is using artificial neural network (ANN) controllers.^{1,2}

ANN is an important technology that currently used to improve the old driving methods of solar tracking systems. Numerous solar tracking systems that used ANN technology have been proposed and implemented to enhance the efficiency of solar energy generation, and to eliminate the current limitations. Intelligent solar tracking systems are promising technology to increase the advantages of solar tracking systems based on their characteristics compared to other technologies. Nowadays, several research are focusing on developing and proposing different solar tracking systems to track the sun efficiently. Numerous intelligent-based ANN systems have been developed to increase the performance of solar tracking systems.

Zaki et al. have implemented neural networks control system to predict the output voltage of the photovoltaic module by using both solar irradiation and five different temperature variables.³ Back-propagation algorithm based on gradient descent algorithm was used to

implement the proposed control system. A modified error function has been used to increase the performance of conventional back-propagation algorithm that has slow convergence. Three variables have been used as input attributes namely solar array temperature, solar irradiance, and maximum voltage reference value. Four hidden nodes constitute the hidden layer, while one output of maximum power point constitutes the output layer. The results that obtained from the proposed controller showed that the neural network with modified error function had better convergence rate. Moreover, only few iterations were needed to reach the maximum reference voltage compared to conventional neural network. However, the proposed controller was only tested by simulation, and the value of the maximum voltage was selected based on a look-up table. Using simulation test and look-up table did not cover the real environmental conditions. In addition, using 10 samples as a data set with high variation between temperatures has produced outlier cases, therefore, inaccurate performance.

Panait et al. have designed a simple dual-axis solar tracking system based on using neural network training. Reducing the cost and the complexity of the conventional neural network based solar tracking systems are the main aims of the proposed model.⁴ The proposed neural network contains four hidden nodes in a hidden layer. The proposed neural network used two tube-cased instruments to measure the solar radiation and spectral composition of light as inputs while the output is the next position of the solar prototype. A Lab-View simulator that used the capacitor and resistor variable trimmers to find the output response of the solar tracker has been used to test and evaluate the proposed model.

Al-Rousan et al. have developed new efficient single and dual axis solar tracking system controllers using adaptive neural fuzzy inference system (ANFIS).⁵ ANFIS controllers used to implement efficient single and dual-axis solar trackers that can increase the performance of solar trackers and predict the trajectory of the sun. Month, day, and time variables were used to predict the orientation angle in single-axis solar tracking system, and both tilt and orientation angles in dual-axis solar tracking system. Three, four, and five membership functions of Sugeno inference were tested and evaluated. The developed controllers were evaluated using MATLAB framework simulation. The results showed that using fuzzy logic model integrated with neural networks can be used sufficiently to control solar tracking systems better than using fuzzy logic and neural networks principles separately. Besides, using five membership functions obtained the optimum efficiency to predict tilt and orientation angles. However, although using ANFIS controllers obtained superior performance, but ANFIS model needs too long processing time since it consists of five hidden layers.

Cheikh et al. have used artificial neural network to increase the solar photovoltaic gained power.⁶ The inputs for the proposed neural network system are solar insolation, junction temperature, and dynamics of charging voltage, while the output is the output power of the photovoltaic module. A battery was used to represent the sun source in the proposed system. Multilayer perceptron (MLP) and a back-propagation MLP neural network with gradient-descent, sigmoid activation function, and mean square error (MSE) as performance function were used as a construction of the proposed model. The simulation results of the proposed model and the maximum power point (MPPT) tracking model were compared together. The results revealed that the proposed model could particularly affected on a complex and non-linear system. The results showed that the data that generated by simulation is not accurate in evaluating the proposed system.

Essefi et al. have proposed an intelligent controller based on neural network to track the MPPT under rapid changes of climatic conditions.⁷ A simulation study by using a DC load was implemented to validate the proposed controller system. The proposed neural network used the

photovoltaic array temperature and solar radiation as inputs to predict the optimum voltage that detect the MPPT. Linear neural network that consists of input, hidden, and output layers was used. One hundred hidden nodes constitute the hidden layers with identity activation function constructed the architecture of linear neural network. Sum squared error (SSE) was used as performance function. By using MATLAB/Simulink software, the simulated controller showed that the developed MPPT controller has fast convergence, efficient, robust, had negligible oscillations around the MPPT, and its possibility to implement easily. However, the proposed controller was implemented by using a linear neural network structure. In addition, 100 hidden neurons need long processing time and consume more energy.

Tahir et al. have proposed a new solar tracking system based on new neural network principle.⁸ A model reference neural network controller (MRNNC) with back-propagation learning algorithm was used to implement the proposed solar tracking system. The idea from the proposed MRNNC is to use input variables including time, current, and reference value to predict the voltage that represent the speed of the motor and voltage that represent the trajectory. The proposed neural network model consists input, hidden, and output layers. Thirteen hidden nodes were used to build the hidden layer with sigmoid activation function, while linear activation function was used for output layer. The MSE was used as a performance function. The proposed MRNNC is superior in dealing with dynamic problems. It could reduce the complexity without sacrificing performance efficiency as well as reduce the disadvantages of conventional systems. However, the proposed system was also implemented based on simulation using MATLAB/Simulink, and some chattering appeared on the ridges of the response signal.

Rabee et al. have developed three predictor systems based on using three different neural network architectures.⁹ Predicting the daily average solar radiation is the main idea of the proposed models. Actual average solar radiation obtained from five different locations were used as input variables to the proposed models. Multilayer feed forward neural networks have been used to implement the three proposed models. Gradient descent method was used to implement the first model. One hidden layer with 10 neurons with hyperbolic activation function for both hidden and output layers were used. On the other hand, Levenberg-Marquart algorithm with Gaussian activation functions in hidden layer and purelin activation function in output layer were used to implement the second and the third neural network models. One hidden layer with 10 neurons were used in the second network, while one hidden layer with 1460 neurons were used in the third network. The mean absolute percentage error (MAPE) was used as a performance function for the proposed models. The results showed that the third proposed model with large number of neurons is highly nonlinear, attempts to follow all the points around the trend, and able to cope with nonlinear data. Moreover, actual data from five different locations were used as training data which give the work more value because it could represent the real-time environment. However, the solution that proposed to solve the maintain generalization of the forecaster, and overfitting problems was to use large number of neurons which will take too long processing time and will consume more energy in reality.

Kayri et al. have proposed two single axis solar tracking systems based on intelligent neural network controller.¹⁰ The proposed neural network systems were implemented to increase the accuracy rate of the predicted values. Several variables including wind speed, wind direction, solar elevation angle, air temperature, relative humidity, and global radiation were used as input variables, while the output was the output power. Feed forward back propagation neural network algorithm was used to implement both of the proposed networks. Two different error

optimization methods were used namely Levenberg Marquardt and Bayesian Regularization. Both neural networks have 15 neurons in each hidden layer. MSE, root MSE, MAPE, relative absolute error, root relative square error, and correlation coefficient (R) were used to evaluate the performance of the proposed systems. The proposed systems were tested for both cloudy and sunny days separately, and the measurements for 3 different days were used to build the proposed systems. Testing the proposed models relived that using the error optimization method would increase the performance of the proposed neural network-based systems. In addition, hardware implementation gave the work more value. However, using data for 3 days only in cloudy weather, and 3 days only in sunny weather could not cover the whole conditions and weather changes.

Yang et al. have proposed artificial neural network vector control system by using adaptive dynamic programming for MPPT control and grid integration of a solar photovoltaic array through an LCL-filter based inverter.¹¹ The idea of the proposed model is to evaluate using neural network learning technique in photovoltaic systems. Feed forward back-propagation neural network with two hidden layers and six neurons in each layer was used to build the proposed controller. Marquart levenberg optimization algorithm, and hyperbolic activation functions were adopted to implement the proposed controller. The performance of the proposed system was evaluated by using both practical implementing in laboratory and MATLAB Simulink simulation. The idea of the proposed model is to predict the DC-DC convertor output. The proposed intelligent controller has improved the performance of the photovoltaic module systems. It was more reliable, and efficient to gain more output power, and it was able to get stable DC voltage although it was not able to cover all the current transient conditions, and it tested in laboratory.

Abdallah et al. have proposed a multilevel world comprehensive neural network model to predict optimum orientation and tilt angles for photovoltaic (PV) panels in all countries worldwide.¹² Meteorological data collected by the photovoltaic geographic information system (PVGIS) in 2019 are used to develop an efficient and accurate feed-forward neural network prediction model.

The idea of multilevel world comprehensive neural network model is to use two stages of neural network to predict the optimum tilt and orientation angles that obtain the maximum solar energy.

The first stage is to use longitude and latitude as input variables to predict orientation and tilt angles. While the second stage is to use the output of the first stage (orientation and tilt angles) along with longitude and latitude to predict three cases of annual radiation. Different topologies of the proposed predictor were tested and evaluated to select the optimum architecture. Two feed forward neural networks with one hidden layer and 350 hidden nodes of Tanh activation function were adopted for both first and second stages. The data set that used were divided into 90% for training and validation, and 10% for testing.

The developed predictive neural model reflected very high accuracy in predicting the PV panels' optimal tilt and orientation angles worldwide.

Şahin, has proposed artificial neural network model to predict the optimum tilt angle of fixed solar panels in Turkey.¹³ A multilayer feed forward artificial neural network model was used to develop the proposed predictor. A total of 126 input variables divided into seven groups namely, number of months, monthly sunshine duration in Turkey, monthly sunshine duration in provinces, latitude of Turkey, longitude of Turkey, latitude degrees of provinces, and longitude degrees of provinces were used to predict 84 output variables of optimum tilt angles in all provinces of Turkey.

The data set used in developing the proposed model were collected from solar panels. The idea is to adjust the solar panels manually according to monthly average optimum tilt angles that predicted by artificial neural network.

Multivariable regression analysis was made by means of PASW Statistics 18 program. The results revealed that the solar panel systems adjusted every month are 34% more efficient than the solar panel systems adjusted according to a fixed angle yearly.

Kim et al. have proposed a solar panel tilt angle optimization model using five different machine learning algorithms.¹⁴ The idea is to use machine learning models to predict the optimum tilt angles that can maximize the converted energy of PV systems.

LR, least absolute shrinkage and selection operator (LASSO), random forest (RF), support vector machine (SVM), and gradient boosting (GB) algorithms were assessed to develop the proposed model in Daegu city, South Korea.

A total of 173,568 records of solar power generation data were collected from fixed 22 PV modules in Daegu city from January 2016 to March 2018. This collected data consists of several features such as module capacity, installation location, module azimuth angle, and panel angle.

Twenty-two features of meteorological data were collected through Meteorological Agency's Open Weather Portal (i.e., temperature, precipitation, wind speed, humidity, and sunshine, etc.).

The record of Solar Power Generation data, along with the collected meteorological data and other variables related to Sun Position data (i.e., declination, hour angle, zenith angle, elevation angle, azimuth angle, the average daily beam radiation, and daily diffuse radiation on a tilted surface.) were used as input variables to predict the solar panels tilt angles.

Correlation analysis was used to find the optimum input variables among all the 31 collected variables. It is found that 14 of these variables can be adopted to develop the prediction models namely, ratio of beam radiation, ratio of diffuse radiation, humidity, cloud amount, temperature, visibility, wind speed, wind direction, cloud height, microdust, precipitation, snow, and aerosol.

Training and testing the selected machine learning models proved that gradient boosting performed better compared to other learning algorithms. Besides, adjusting solar panels based on the predicted results could increase the amount of solar energy generated by PV modules compared to fixed panels.

AL-Rousan et al. have developed intelligent models to predict optimum orientation and tilt angles of solar PV module.² Real-time data collected from PV modules installed in Irbid city, Jordan were used to predict tilt and orientation angles using linear regression, MLP and CMLP intelligent models.

The idea is to investigate and evaluate several solar variables to select the most effective and dominant variables on adjusting both dual and single-axis solar PV modules. Correlation analysis was used to select the most appropriate variables to develop the proposed models.

MATLAB mathematical framework was used to evaluate the proposed models, and one, two, and three hidden layers with 15 hidden neurons were tested in both MLP and CMLP models.

The correlation analysis results revealed that month, day, and time are the most effective variables for horizontal single-axis and dual-axis solar tracking systems. Using these variables in MLP and CMLP produced high performance while linear regression predicted less than 70% of the given data in most cases.

Table 1 shows a comparison between different research that used artificial neural network principle to build intelligent solar tracking systems.

TABLE 1 Comparison between intelligent solar tracking systems controllers

Reference	Input-Output variables	Axis	Characteristics	Advantages	Disadvantages
Zaki et al. ³	Input: solar irradiance, five temperatures, V_{max} from look-up table. Output: MPPT	Simulation	<ul style="list-style-type: none"> Back-propagation training algorithm One hidden (4 neurons) Few iterations (not specified)	<ul style="list-style-type: none"> Better convergence rate Few iterations were needed to reach the reference value	<ul style="list-style-type: none"> Using look-up table to find the output voltage High variation between temperature values Few samples used Tested by simulation
Panait et al. ⁴	Input: solar radiation, and spectral composition of light Output: next position of the solar prototype	Dual	<ul style="list-style-type: none"> 4 hidden nodes in the core Sigmoid activation function	<ul style="list-style-type: none"> Cheap Not complex Efficiency up to 100% in summer and 40% in winter	Tested by simulation using Lab-View
AL-Rousan et al. ⁵	Inputs: Month, day and time Output: tilt and orientation angle	Single and dual axis	<ul style="list-style-type: none"> Five layers of neural network integrated with fuzzy logic rules (ANFIS) 3, 4, and 5 membership functions Sugeno inference MSE performance function 	The ANFIS controller has shown better results than the conventional neural network controller	MATLAB mathematical framework was used to evaluate the proposed models
Cheikh et al. ⁶	Inputs: solar insolation, the junction temperature, and the dynamics of the charging voltage Output: output power of the photovoltaic	Simulation	Back-propagation multilayer perceptron neural network with gradient-descent algorithm MSE performance function Sigmoid activation function	High gained power Low error rate of 0.063%	No practical data Based on simulation study

(Continues)

TABLE 1 (Continued)

Reference	Input-Output variables	Axis	Characteristics	Advantages	Disadvantages
Essefi et al. ⁷	Inputs: the photovoltaic array temperature, and the solar radiation Output: optimum voltage which corresponds to MPP	Simulation	<ul style="list-style-type: none"> Linear Neural Network One hidden layer 100 hidden nodes in the hidden layer SSE performance function Purelin activation function	<ul style="list-style-type: none"> Efficient Fast response time Negligible oscillations around the maximum power point. Easy implementation	<ul style="list-style-type: none"> Used linear neural network 100 neurons in hidden layer need long process time and energy Based on simulation study
Tahir et al. ⁸	Inputs: time, current, reference value Outputs: voltage that represent the speed of the motor, and voltage that represent the trajectory	Simulation	Feed forward model reference neural network With Back-propagation learning algorithm One hidden layer (13 neurons) MSE performance function Linear activation function in output layer Sigmoid activation function in hidden layer	Used with complex and nonlinear systems Reduces the disadvantages of conventional controllers	Based on simulation study Some 'chattering' appears on the ridges of the response signal.
Bou-Rabee et al. ⁹	Inputs: actual average solar radiation from five different locations	Simulation	<ul style="list-style-type: none"> Three multilayer feed-forward ANNs one hidden layer with 10 neurons (Gradient Descent Method) hyperbolic in hidden and output layers one hidden layer with 10 neurons (Levenberg-Marquardt algorithm) 	<ul style="list-style-type: none"> Able to cope with nonlinear data. Actual data were used for training 	<ul style="list-style-type: none"> (1) need maintain the generalization of the forecaster and to overfitting (2) achieve the target after 16 epochs (3) highly nonlinear and attempts to follow all the points around the trend

TABLE 1 (Continued)

Reference	Input-Output variables	Axis	Characteristics	Advantages	Disadvantages
	Output: the daily average solar radiation		<ul style="list-style-type: none"> one hidden layer with 1460 neurons (Levenberg-Marquardt algorithm) MAPE performance function 2 and 3 gaussian in hidden, purelin in output layer 		Used large number of neurons (1460) Need long process time and consume more energy.
Kayri et al. ¹⁰	Input: wind speed, wind direction, solar elevation angle, air temperature, relative humidity, and global radiation	Implemented practically	<p>2 feed forward neural networks: Levenberg Marquardt</p> <p>Bayesian Regularization</p> <p>15 neurons in each hidden layer.</p> <p>MSE, root MSE, MAPE, RAE, RRSE, and correlation coefficient (R).</p>	Proposed systems tested for both cloudy and sunny days practically using the error optimization method increased the performance	using data for three days only in cloudy weather, and three days only in sunny weather
	Output: output power				
Sun et al. ¹¹	Input: grid ac voltage, grid ac current, output voltage of the PV array, output current of the PV array Output: DC-DC convertor output	Simulation	<ul style="list-style-type: none"> Feedforward back propagation neural network with Marquart levenberg 6 neurons with 2 hidden layers 	<ul style="list-style-type: none"> Could maximize the output power More reliable and efficient extraction of solar power More stable DC-link voltage 	<ul style="list-style-type: none"> Tested in laboratory Not able to cover all transient conditions

(Continues)

TABLE 1 (Continued)

Reference	Input-Output variables	Axis	Characteristics	Advantages	Disadvantages
Abdallah et al.	Stage1: Input: Longitude and latitude, Output: Tilt and Orientation angles Stage2: Input: tilt, orientation, longitude, and latitude, Output: three outputs of annual radiation	Simulation	Hyperbolic activation function multilevel world comprehensive feed forward neural network model Two neural networks with one hidden layer and 350 neurons for hidden layer Levenberg Marquart Algorithm	Could obtain very high accuracy and efficient results worldwide.	No practical data Two stages of neural network with 350 neurons for each hidden layer used.
Şahin	Input: 126 input variables divided into 7 groups namely, number of months, monthly sunshine duration in Turkey, monthly sunshine duration in provinces, latitude of Turkey, longitude of Turkey, latitude degrees of provinces, and longitude degrees of provinces Output: 84 output variables of optimum tilt angles in all provinces of Turkey	PASW Statistics 18 program	<ul style="list-style-type: none"> Multilayer feed forward artificial neural network Levenberg-Marquardt algorithm 	Solar panels could perform 34% more efficient than the fixed solar panel systems	Huge amount of input and output variables Analyzed statistically using PAWS No detailed information about the structure used Needs long processing time
Kim et al.	Input: ratio of beam radiation, ratio of diffuse radiation, humidity, cloud amount, temperature, visibility, wind speed, wind direction, cloud height, micro dust, precipitation, snow, and aerosol.	Single axis	Linear regression (LR), least absolute shrinkage and selection operator (LASSO), random forest (RF), support	Examined 32 solar variables and found 14 variables as optimum variables to be used.	Huge amount of unnecessary variables were used Repeated every month after adjusting the solar panels manually

TABLE 1 (Continued)

Reference	Input-Output variables	Axis	Characteristics	Advantages	Disadvantages
	Output: tilt angle of PV panels.		vector machine (SVM), and gradient boosting (GB)	Could increase the amount of solar energy generated by PV modules compared to fixed panels.	No detailed information about the implemented models
AL-Rousan et al.	Input: month, day, and time.	Single and dual axis	<ul style="list-style-type: none"> Linear Regression, MLP, CMLP 	Selected the optimum variables to develop the proposed models	MATLAB mathematical framework was used to evaluate the proposed models
	Output: orientation angle in single-axis PV, and tilt and orientation angles in dual-axis PV.			MLP and CMLP produced high performance results	Linear regression predicted less than 70% of data

As shown in Table 1, several research have focused on proposing, developing, and building simulated solar tracking systems. Several intelligent controllers have been used to drive solar tracking systems. Neural network controllers are widely used to predict several criteria such as solar radiation in a specific region, maximum power point, trajectory of the photovoltaic modules systems, or duty cycle.

Besides, intelligent controllers are used to control the solar photovoltaic motions by predicting the next suitable position of solar panels toward the position of the sun. Intelligent driver systems are proposed to solve the problems of other driving methods by using the training process which can faster the process to find the next trajectory of the sun across the sky, therefore, decrease the amount of consumed energy. In addition, it can also increase the performance of solar tracking systems by minimizing the error rate and increasing the prediction rate. However, designing adaptive intelligent solar driving controllers that can be used globally is very difficult due to the difficulty to obtain a high prediction rate, and also a very low error rate with low time complexity.

On the other hand, the collected data and the solar variables that used to train and test the implemented solar tracking systems play a major role to propose an efficient, simple, adaptive, and high performance intelligent based solar tracking system controllers that can predict the trajectory of the sun across the sky accurately.

As shown in literature review, several research are carried out by developing the conventional neural network by changing the selected variables,^{2,13} modifying the error optimization function,³ using a reference value to the neural network inputs,^{3,8} finding new generation of neural network,^{4,10,12} or increasing the number of neurons in the hidden layers.^{7,9} Artificial neural network proved that it is sufficient to be used as solar tracking systems controller, and effective in complex and nonlinear systems.^{2,5} Although solar tracking systems based artificial neural network are better than conventional solar tracking systems, but these systems suffer from many problems including slow convergence to reach the desired goal, high cost, high complexity, and outlier cases when a variation in data is used. Using intelligent controllers could improve the efficiency of solar systems, minimize energy loss, reduce time response, and eliminated fluctuations around the outputs. Based on a literature review, several solutions were proposed to improve the efficiency of solar tracking controllers. Modifying the error optimization function was used to increase the performance of conventional feed forward back-propagation algorithm that has slow convergence.

In contrast, changing the architecture of the proposed solar tracking systems, adding new devices, and changing the selected variables cannot guarantee solving the disadvantages of the conventional neural networks controllers, and obtaining a high efficiency solar tracking system.

Moreover, several research have used the conventional neural network to implement different types of intelligent controllers.^{2,5,6,9} However, to use neural network principle, the input data are fed directly to neural network to train them. The performance of a neural network model fully depends on the input data nature. In some cases, increasing the complexity of input data could lead to produce low performance models. This is because when the input data are complex for neural network, neural network would not well explain the relationship between these data. Thus, it performed worse than when the input data are simple. Therefore, using high nonlinear and complex intelligent controllers in driving solar tracking systems will cause many problems. It will slow down the process of predicting tilt and orientation angles, and increase the complexity of the solar trackers, thus, increase the consumed energy.

To solve this problem, many works increased the number of hidden layers in the network or the number of neurons in the hidden layers which obtain highly nonlinear neural network

which could improve the generalization of the forecaster, and reduce the overfitting problems, but, it would increase the time complexity of the intelligent neural network based controller systems compared to conventional systems.

However, the main target of using artificial neural network to drive and control solar tracking systems is to simplify the current process of predicting the next trajectory of the sun and faster the process of moving solar tracking systems. Thus, the current intelligent solar tracking systems based neural network should be improved by minimizing the high level of nonlinearity (i.e., number of hidden layers and number of neurons) to decrease cost, processing time, and energy loss.

One of the best methods to decrease the high level of nonlinearity in neural network without effecting on its performance is to modify the error optimization function using a reference value as an input to neural network along with original input data. This reference value could premap the relationship between the data samples and guide neural network to reach the desired goal and predict the output variables faster and more accurate without the needing to increase the high level of nonlinearity.

Moreover, it is found that logistic regression (LR) classifier is more robust to find the relationship between samples of data compared to other statistical models.^{15,16} This is due to its characteristics that LR classifier is appropriate for developing a probabilistic framework.^{17,18} This is because it can easily adjust the classification threshold to find confidence classes.¹⁸ LR classifier can easily incorporate new training data into the model as well.¹⁹ However, LR can deal with nonlinear relationship similar to linear relationship between dependent and independent variables.²⁰ This characteristic is very important when unknown relationships exist between dependent and independent variables. These advantages lead to select LR classifier to enhance the performance of conventional neural network principles.

Solving the problems of the current solar tracking systems based on intelligent artificial neural networks is the main motivation globally. Thus, this study aims to solve the problems of using conventional neural networks principle by developing new single and dual axis solar tracking systems. The idea is to use a reference value as input variable along with original input data. This reference value could premap the relationship between the data samples. Premapping the relationship between the data samples could guide the neural network to faster predict the desired goal without the needing to increase the high level of nonlinearity. This can be done by integrating both LR model and conventional neural network principles. The output of both binomial logistic regression (BLR) and multinomial logistic regression (MLR) models are adopted as reference values to premap the data of both conventional MLP and CMLP neural networks respectively. In addition, the outputs of Both BLR and MLR are trained by using one of unsupervised clustering algorithms namely k-means, fuzzy c-means, and hierarchical clustering algorithms. Integration of MLP and CMLP principles with a trained LR model could produce new generation of intelligent solar tracking systems, achieve the target of this study, and improve to the field of intelligent solar tracking systems by minimizing the high level of nonlinearity and decreasing the complexity of used MLP controllers.

Besides, to the best of the authors' knowledge, there are no reported studies that developed the performance of MLP and CMLP by using a reference value that could premap the relationship between the data samples and guide the neural network to faster predict the desired goal without the needing to increase the high level of nonlinearity. On the other hand, this article will introduce a MLP structure with a single hidden layer that acts and performs better than MLP network with three hidden layers, also it aims to compare between the proposed models in this article and the most recent published articles in the field. This would show the capability and the robustness of the

proposed models compared to other published articles, besides, it will present the time complexity analysis between recent published algorithms besides to other parameters and principles. Developing the proposed models, and solving the problems of conventional neural network using the reference value would contribute to the fields of solar energy and intelligent solar tracking systems, and it will make this study excellent reference to the audience in the field.

The contribution in this paper is to (i) develop a new structure of neural networks where a single hidden layer performs better than three hidden layers in conventional multilayer perceptron (MLP), (ii) develop new generation of intelligent solar tracking systems where the level of nonlinearity is minimized and the time complexity and estimation time are decreased, (iii) developed the performance of MLP and CMLP by using a reference value that could premap the relationship between the data samples and guide the neural network to faster find the relationship between variables, and then faster predict the desired goal, (iv) present both MLP and CMLP models in details, which would make this article as a good reference to those who will use such models, (v) compare between the proposed models in this article and the most recent published articles in the field which will highlight the main drawbacks in the published intelligent models used to predict tilt and orientation angles, and (vi) this is the first article that focuses on minimizing the time complexity of MLP and CMLP models by adding a well-designed reference value rather than increasing the number of hidden layers and hidden nodes.

Integrating MLP/CMLP with LR models to minimize the linearity of the adopted neural networks is a novel and superior idea and can enhance the efficiency of the used applications, besides, it can decrease the estimation time to obtain the results.

2 | PRELIMINARIES AND DEFINITIONS

This section explains the main mechanisms and preliminaries that used to build the proposed solar tracking systems. This section is divided into three parts namely clustering algorithms, LR models, and artificial neural network. The first part focuses on explaining the clustering algorithms including k-means clustering, fuzzy c-means clustering, and hierarchal clustering algorithms. The second part presents the LR models including BLR and MLR, while the third part discusses the artificial neural network including both MLP and cascade multilayer perceptron (CMLP).

2.1 | Clustering algorithms

Clustering is the most important unsupervised learning technique that used to find the structure of a collection of data to group the data that have the similar characteristics together in one cluster.²¹ The clustering process aims to increase the similarity in intra-cluster and to decrease the similarity between the clusters. Moreover, the clustering technique is divided into mainly three groups namely exclusive clustering, overlapping clustering, and hierarchical clustering. In exclusive clustering, the data point should belong to one cluster such as K-means clustering algorithm. In overlapping clustering, the data point can belong to multiple clusters in the same time such as Fuzzy C-Means.²² In hierarchical clustering, the data point is organized as a tree which allows each cluster to have sub clusters such as hierarchical clustering. In this study, three learning techniques will be tested to implement the developed solar tracking systems including k-means clustering, hierarchical clustering and fuzzy c-means clustering. These clustering algorithms are the most common used clustering algorithms in different fields because of their efficiency compared to other algorithms.^{21,23–27}

2.1.1 | K-means clustering

K-means is a clustering algorithm that introduced in 1967 by MacQueen.²⁸ The procedure to cluster the data is based on defining a certain number of clusters (K). The algorithm will define k centroids for each cluster randomly from the data set to use them as a head for each cluster to assign a data point to each one of cluster heads. The next step is to take each point and associate it to the nearest centroid using the Euclidean distance equation (d) between each centroids (c) and each data point (x) as follow:²⁸

$$d(x, c) = \sqrt{\sum_{i=1}^K |x_i - c_i|^2} \quad (1)$$

Euclidean distance equation is the length of a line segment between two points. It can be calculated from the Cartesian coordinates of the points using the Pythagorean Theorem, and it is normally called the Pythagorean distance.

Once all the data is attached to one of the K -centroids, the first step of the algorithm is completed. At this point, to improve the performance of the clusters, new centroids are recalculated by using Equation (1) as follows:⁷

$$c_j = \frac{c_{j-1} + \sum_{m=1}^M (x_m)}{M} \quad (2)$$

where c_{j-1} and c_j are the old and new mean values for the cluster head, respectively, x_m is the data point of m th index data that attached to cluster c_{j-1} , and M is the number of data attached to cluster c_{j-1} . The steps will be repeated until no more changes are appeared. In other words, no points will move from one cluster to another. Finally, the execution of the algorithm will stop if the maximum iteration is reached or squared error function is minimized. Therefore, the objective function (J) in this case is a squared error function which defined as follows.²²

$$J = \sum_{i=1}^K \sum_{m=1}^M d^2(x_{i,m}, c_i) \quad (3)$$

where c_i is the cluster head, $x_{i,m}$ is the data point m in cluster i , M , and K are the total number of data point in cluster i , and total number of cluster heads, respectively, where, the squared error function for a point x with respect to its cluster center c is the distance between P and c squared; that is, $(x_i - c_i)^2 + (x_y - c_y)^2$.

2.1.2 | Hierarchical clustering

The first version of hierarchical clustering is introduced in 1967 by Johnson.²² The procedure to cluster the data is based on building a tree of clusters. Strategies for hierarchical clustering is divided mainly into two types including the bottom up approach and the top down approach. In bottom up approach, each data starts in its own cluster, and every pairs of clusters are merged together as one moves up the hierarchy to create a tree. In top down approach, all the data is considered in one cluster, and splitting is used to build a tree. Moreover, to build the clusters using hierarchical clustering algorithm, the number of clusters should be defined first

(i.e., K). The algorithm starts by assigning each item to its own cluster, and then for each pair of clusters in data set, the closest pair of clusters is found and merged into a single cluster using Euclidean distance between two clusters.²⁹ The distance between clusters is calculated using the average-linked clustering equation which is the mean distance between elements of each cluster as follows:³⁰

$$D(C_e, C_f) = \frac{1}{F * E} \sum_{a \in C_e, b \in C_f} d(a, b) \quad (4)$$

where $d(\cdot)$ is the Euclidean distance function, C_e and C_f are clusters which contain the members, F and E are the number of members in clusters C_e and C_f , respectively, a and b are the members of clusters C_e and C_f respectively. Finally, the process of merging the clusters based on average-linked distance equation will be repeated until all data are clustered into a single cluster that contains all samples of data.

2.1.3 | Fuzzy C-means clustering

Fuzzy C-means is a clustering algorithm that developed by Dunn,³¹ and improved by Bezdek.³² The aim of using Fuzzy C-means clustering is to allow each data to belong to two or more clusters each time by assigning different degrees of membership to each data point. Data point in C-means clustering is assigned membership to each cluster center, therefore, data point may belong to more than one cluster center. The algorithm starts by assigning primary centroids C to the predefined number of clusters based on the user definition. Then, the degree of membership (u) of all feature vectors in all the clusters will be computed to be used in updating the cluster head. The membership can be computed by using the following formula:³²

$$u_{ij} = \frac{1}{\left(\sum_{k=1}^C \frac{d(x_i, c_j)}{d(x_i, c_k)} \right)^{\frac{2}{m-1}}} \quad (5)$$

where $d(\cdot)$ is the Euclidean distance, c_j is the D-dimension center of the cluster of j th index, c_k is cluster centers of k th index, x_i is the i th of D-dimensional measured data, u_{ij} is the degree of membership of x_i in the cluster j , m is any real number greater than 1. Fuzzy c-means clustering algorithm will use the degree of membership of x_i in the cluster j to calculate the new cluster head for cluster j , and the new cluster head is calculated by finding the centroid of the cluster j using the following formula:³²

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m * x_i}{\sum_{i=1}^N u_{ij}^m} \quad (6)$$

where c_i is the D-dimension center of the cluster of j th index, x_i is the membership of cluster of index j th, N is the number of membership of cluster of index j th.

The algorithm will stop executing when the condition of the termination is reached. The termination condition is reached when the distance between degree of membership function in two consecutive steps (i.e., s and $s + 1$) is less than ε ($d(u_{ij}, \hat{u}_{ij}) < \varepsilon$). The ε is a

termination condition between 0 and 1. If the condition is not met, the algorithm will calculate the new clusters and the new degree of membership function and the process will be repeated until the condition is met. Fuzzy c-means clustering algorithm is based on minimization of the objective function (J). The objective function is the sum of weighed distances in Euclidean metrics from the data samples to the cluster head in each cluster as shown in the following formula:³²

$$J = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m d^2(x_i, c_j) \quad (7)$$

The clustering algorithm is used to divide the data into predefined number of groups, therefore, find the cluster number of specific data point. The process should be repeated using all data points to find a solution.

2.1.4 | Clustering algorithms comparison

To compare between the clustering algorithms, seven metrics can be used including the data set size, the number of clusters, the algorithm speed, the parametric/nonparametric test, the hard/soft clustering, the clustering technique and the Big-O analysis for the complexity. Table 2 shows the comparison between the clustering algorithms.

As shown in Table 2, the K-means clustering is the fastest algorithm, while the hierarchical clustering is slowest because it uses merging/splitting to reach the top of the tree. However, there is no proof to show which algorithm have a high capability of clustering the data more efficiently.^{37–39}

2.2 | Logistic regression

Several researchers recommended the basic guidelines for statistical rules of building a statistical regression models.^{40–42} These rules can be summarized in Table 3.

LR is a model that can find the association between a categorical dependent variable and a set of independent variables.⁹ LR is a supervised learning technique that uses the independent variables (inputs) to predict the dependent variable (output). If the dependent variable has two categories, then it is called a BLR, otherwise if the number of categories is greater than two then it is called a MLR.⁴³ The LR aims to build a classifier model based on the trained data to predict the new data points of the system. To implement LR, the coefficients (B_i where $i \geq 0$) of the LR in Equation (8) should be computed first.

Equation (8) is defined as an estimated LR equation which formulated as follows:⁴⁰

$$f = \beta_0 + \sum_{i=1}^K \beta_i * x_i \quad (8)$$

where f is the MLR/BLR equation, β_0 is the constant, β_i is the coefficient of i th variable, x_i is the i th variable and K is the number of variables.

For BLR K is equal to two and for MLR, K is greater than 2. To calculate the coefficients of MLR and BLR, first set N , p and K as number of samples, number of variables, and number of

TABLE 2 Comparison between intelligent solar tracking systems controllers

Metric	K-means	Hierarchical	FCM	Note	Metric
Data set size	Large data set	Small data set	Large data set	K-means and FCM clustering algorithms can work under a huge data set with different execution time, ³³ where hierarchical clustering is used for small data set since the calculation and the complexity of the algorithm is extremely high, ³⁴	Data set size
Number of clusters	Predefined	Do not need to be specified	Predefined	FCM and K-means algorithms cannot execute without redefining the number of clusters of the data ³⁴ , where hierarchical algorithm do not need to specify the number of clusters on advance, at any time the algorithm can be stopped. ³⁵	Number of clusters
Algorithm speed	Fast	Slow	Intermediate	K-means is the fastest algorithm between the three used algorithms, where hierarchical clustering is the slowest algorithm, since the merge and split process will take a lot of time and memory. ³⁵	Algorithm speed
Parametric/nonparametric	Parametric	Nonparametric	Parametric	Hierarchical clustering is considered as a non-parametric unsupervised learning because no density functions is considered in the algorithm and only natural groupings (clusters) in a data set is found. ³⁶	Parametric/nonparametric

TABLE 3 The statistical rule for building most suitable regression models

Number of dependent variables	Nature of independent variables	Nature of dependent variable(s)	Test
1	1 or more interval and/or 1 or more categorical	Categorical (multilevels)	Multinomial logistic Regression
1	1 or more interval and/or 1 or more categorical	Categorical (two- levels only)	Binomial logistic Regression
1	1 interval	Interval and normal	Linear Regression

clusters respectively. β is the coefficients vector with size equal to $(K - 1) * p$ and the last cluster K is considered as a reference for other clusters. X is denoted as vector contains the variables x_i (Equation 8) with size equal to K and the first element is equal to 1 to represent the constant in the Equation (8) (i.e., β_0). Moreover, g_i and y_i contain number of clusters (from 1 to K) without reference variable and number of clusters (from 1 to $K - 1$) with reference variable, respectively. $p(x_i, \beta)$ is the event probability that calculated using log odds of event success which is a linear function of the predictors where, the coefficient (b_1) is the amount the logit (log odds) changes with a unit change in x as follows.⁴⁴

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{i=1}^K \beta_i * x_i \quad (9)$$

equivalently to

$$p(x_i, \beta) = \frac{e^f}{1 + e^f} = \frac{e^{\beta_0 + \sum_{i=1}^K \beta_i * x_i}}{1 + e^{\beta_0 + \sum_{i=1}^K \beta_i * x_i}} \quad (10)$$

where β is a vector of LR coefficients, and x_i is the vector of i th variable as follows:

$$\beta = \begin{pmatrix} \beta_{10} \\ \beta_{11} \\ \vdots \\ \beta_{1p} \\ \beta_{21} \\ \vdots \\ \beta_{2p} \\ \vdots \\ \beta_{(K-1)0} \\ \vdots \\ \beta_{(K-1)p} \end{pmatrix}, X = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_{K-1} \end{pmatrix}$$

Then to find the optimal values for, the maximum log likelihood function in Equation (11) should be maximized. To maximize the function Newton–Raphson will be used. Newton–Raphson

is a way to quickly find a good approximation for the root of a real-valued function $f(x) = 0$ as shown in the following equation.⁴⁴

$$\ell(\beta) = \sum_{i=1}^N \log p_{g_i}(x_i, \beta) \quad (11)$$

As a result of applying Newton–Raphson method, the coefficients vector (β) of MLR and BLR are calculated in Equations (12) and (13), respectively.^{45,46}

$$\beta^{new} = \beta^{old} + (X^T \tilde{X})^{-1} X^T (y - p) \quad (12)$$

$$\beta^{new} = \beta^{old} + (\tilde{X}^T W \tilde{X})^{-1} \tilde{X}^T (y - p) \quad (13)$$

where β^{new} and β^{old} are the coefficients vector of the current iteration and previous iteration respectively. \tilde{X} is a diagonal matrix with X -vector, y is the concatenated indicator vector of dimension $N * (K - 1)$ which explained as follows.⁴⁶

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{k-1} \end{pmatrix}, y_k = \begin{pmatrix} I(g_1 = k) \\ I(g_2 = k) \\ \vdots \\ I(g_N = k) \end{pmatrix} \quad (14)$$

where $I(\cdot)$ is indicator function equal to 1 if the argument is true and otherwise 0. The p vector is defined as fitted probabilities vector with size equal to $N * (K - 1)$. To calculate p the following formula is used:

$$p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{K-1} \end{pmatrix}, p_k = \begin{pmatrix} p_k(x_1, \beta) \\ p_k(x_2, \beta) \\ \vdots \\ p_k(x_N, \beta) \end{pmatrix} \quad (15)$$

Finally, W is square matrix that calculated its elements using the probability in Equation (10), square matrix can be calculated as follows:

$$W = \begin{pmatrix} W_{11} & W_{12} & \dots & \dots & W_{1(K-1)} \\ W_{21} & W_{22} & \dots & \dots & W_{2(K-1)} \\ W_{31} & \dots & \dots & \dots & W_{3(K-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{(K-1)1} & W_{21} & \dots & \dots & W_{(K-1)(K-1)} \end{pmatrix} \quad (16)$$

where each element is calculated based on the condition as follows:⁴⁶

$$W_{km} = \begin{cases} p_k(x_i, \beta^{old})(1 - p_k(x_i, \beta^{old})), & \text{if } k = m \\ -p_k(x_i, \beta^{old})p_m(x_i, \beta^{old}), & \text{if } k \neq m \end{cases} \quad (17)$$

To calculate probability as a vector Equation (10) can be written as follows:³⁷

$$p_k(x_i, \beta) = \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}} \quad (18)$$

The MLR and BLR will repeat the process of finding optimal coefficients vector (β) until one of the termination conditions is reached including absolute convergence criterion for β , absolute convergence criterion for log-likelihood, or maximum allowable number of iterations.⁴⁵ After finding the optimal equation for the estimate logistic function (Equation 8) the training phase is completed.

Moreover, to classify the new data point to the predefined clusters, LR uses the probability formula (Equation 10) to calculate the probability between each cluster and the new data point and then the maximum probability that attached to the cluster number will be assigned to the new data point. The time that taken to classify the data set can be calculated by finding the time complexity of LR model by using big O notation as follows:⁴⁷

$$Complexity_{LR} = O(n \times v^2 \times c) \quad (19)$$

where n denotes the number of training samples, v denotes the number of attributes, and c denotes the number of output variables.

2.3 | Artificial neural network

computing system that learned to do a task using a number of simple highly interconnected processing elements (called neurons) which process the information through connections (called weights) between layers.⁴⁸ The main and simplest basic element in the neural network is called neuron as shown in Figure 1. The perceptron is the simplest network possible that uses single neuron and use a computational model to process one or more inputs to produce a single output. The inputs of each neuron should contain the input (i.e., x_n), the weight (i.e., w_n), and sometimes an addition connection (i.e., bias) is added to improve the performance of the neuron.⁴⁷ To find the output of the neuron each input will be multiplied with the connected weight and then the summation for all terms is considered as the output of the neuron. The result of the summation will be forwarded to the activation function $f(\cdot)$ to calculate the output of the neuron, usually the activation function is nonlinear function. Moreover, neural networks use different activation functions. The hyperbolic tangent transfer function has been selected to calculate the output of the hidden layers nodes. The hyperbolic tangent transfer function can produce outputs in scale of $[-1 +1]$ because it is a continuous function and can produce output for every x value as follows.⁴⁹

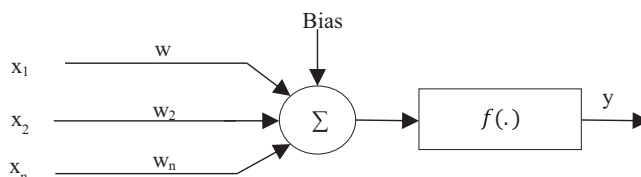


FIGURE 1 The single perceptron of artificial neural network

$$f(z) = \frac{2}{1 + e^{-2z}} - 1 \quad (20)$$

where z is the inputs of the neuron.

On the other hand, identity transfer function has been used to find the output of the output layer node. Identity transfer function is a linear activation function where the output of the functions will not be confined between any range as represented in Equation (21):⁴⁹

$$f(z) = z \quad (21)$$

2.3.1 | MLP and CMLP neural network

Neural network can be classified based on the signal flow which are feed forward and recurrent (cascade) networks. Other classification could be based on the existence of the hidden neurons (i.e., neurons which is not in the input or output layer) which are either MLP or single layer perceptron. In this study, two types of the networks will be considered including MLP and CMLP as shown in Figures 2 and 3, respectively. MLP is an artificial neural network wherein connections between the units do not form a cycle. The architecture of conventional MLP that consists of one hidden layer is shown in Figure 2. Based on Figure 2, the prediction output of the j th node in the hidden layer is calculated by⁵⁰

$$O_j = f\left(\sum_{i=1}^I x_i w_{ij} + b_j\right) \text{ for } 1 \leq i \leq I \quad (22)$$

where $f(\cdot)$ is the activation function, i is the number of nodes in the input layer, x_i and w_{ij} are inputs and weight from i th node in the input layer to the j th node in the hidden layer, and b_j is the bias for j th hidden node. while the output of k th node in the output layer can be calculated by⁵⁰

$$y_k = f\left(\sum_{j=1}^J O_j w_{jk} + b_k\right) \text{ for } 1 \leq k \leq K \quad (23)$$

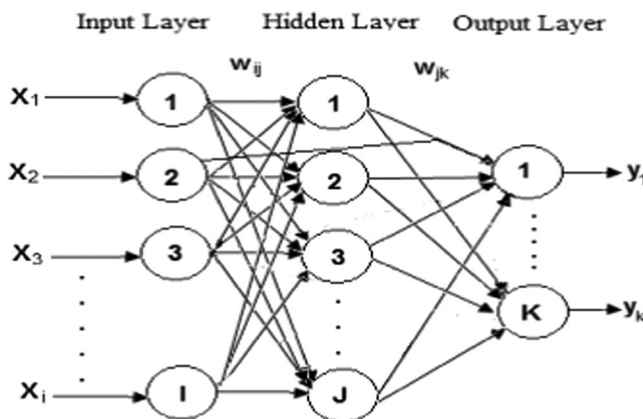


FIGURE 2 The architecture of multilayer perceptron network for one hidden layers

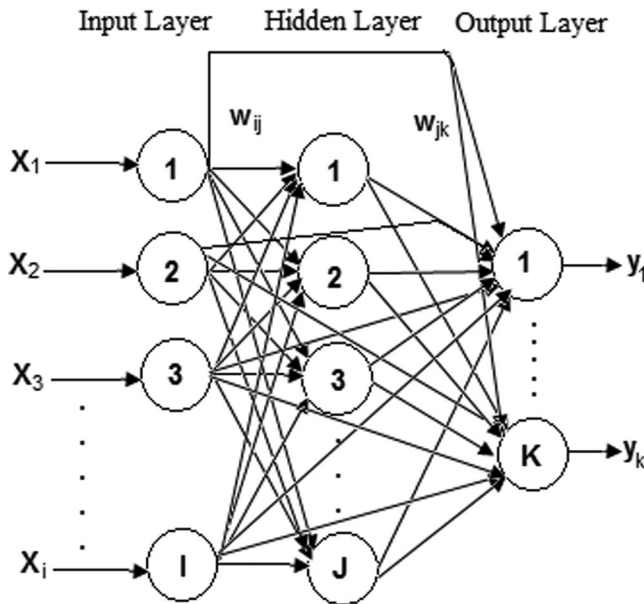


FIGURE 3 The architecture of cascade multilayer perceptron network for one hidden layers.

where y_k is the output of k th node in the output layer, K is the number of nodes in the output layer, w_{jk} is the weight from the j th node in the hidden layer to the k th hidden node in the output layer, and b_k is the bias for the k th output node.

On the other hand, CMLP network is an artificial neural network that has the same structure and the work methodology as MLP.^{51,52} In contrast, the CMLP uses a weight connection from the input directly to each layer, and from each layer to all successive layers.^{51,52} The architecture of CMLP that consists of one hidden layer is shown in Figure 3.

As shown in Figure 3, the input layer passes the input data directly to the output layer and all successive layers. The predicted output of k th node in the output can be calculated by:⁵⁰

$$y_k = f \left(\sum_{i=1}^I w_{ik} x_i + \sum_{j=1}^J O_j w_{jk} + b_k \right) \quad (24)$$

where y_k is the predicted output of k th node in the output layer, w_{ik} is the weight from the i th node in input layer to the k th node in output layer.

2.3.2 | Back propagation optimization algorithms

To calculate the predicted output of MLP and CMLP, the network should be learned first. To learn a network two major learning models are used including the online learning (instantaneous learning) and the offline learning (training or batch learning).⁴⁸ In online learning, the weights and biases will be updated upon presentation of each pattern, while in offline learning the weights and biases will be updated after complete set of patterns (after learning each neuron separately). In several research and mathematical tools, online learning is denoted as adaption learning and offline learning as training. Many developments on the conventional

MLP and CMLP have been suggested to speed up the learning process and to enhance the predicted outputs.^{52–54} Both learning methods should be implemented on the backward way that the network will start back propagate the results to update the weights and biases after calculating the outputs of the network. Two learning methods can be used to back propagate the results including gradient decent with momentum back propagation and Levenberg-Marquardt back propagation.

After calculating the output of the network by using Equation (22) for MLP and Equation (23) for CMLP, the error will be propagated backward to adjust the weights and biases using the first order-derivative for the error function as shown in Figure 4.

As shown in Figure 4, the direction of backward propagation is from output layer to input layer. While moving to input layer, weights and biases will be updated step by step (i.e., first from out to h2, then from h2 to h1 and finally from h1 to in). The error function that used in gradient decent model is defined as follows:⁵⁵

$$E(x, w) = \frac{1}{2} \sum_{p=1}^P \left[\sum_{k=1}^K (d_{kp} - o_{kp})^2 \right] \quad (25)$$

where $E(x, w)$ is the summation of the squared errors function for an input vector x and a weight vector w , w is the weight vector of the network that:

$w = [w_1, w_2, \dots, w_N]^T$ which consists all the weights of the network, x is the input vector of the network, d_{kp} is the desired output vector of the k th output and p th pattern, o_{kp} is the actual output vector of the k th output and p th pattern, P is the index of patterns that $1 \leq p \leq P$, and K is the index of network outputs $1 \leq k \leq K$. The gradient should be computed for each weight in vector w using a chain rule as follows:⁵⁵

$$\frac{\partial E(x, w)}{\partial w} = \left[\frac{\partial E}{\partial w_1} \frac{\partial E}{\partial w_2} \dots \frac{\partial E}{\partial w_n} \right]^T \quad (26)$$

Therefore, to calculate the new weights and biases between the j th neurons of the hidden layer and the i th neurons in the input layer the following equations are used.⁵⁶

$$w_{ji}(t) = w_{ji}(t-1) + \Delta w_{ji}(t) \quad (27)$$

$$b_j(t) = b_j(t-1) + \Delta b_j(t) \quad (28)$$

where $w_{ji}(t)$ and $b_j(t)$ are the current weight and bias, $w_{ji}(t-1)$ and $b_j(t-1)$ are the previous weight and bias, $\Delta w_{ji}(t)$ and $\Delta b_j(t)$ are the increment of weights and biases that can be calculated using the following formulas:⁵⁶

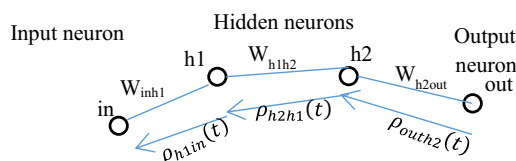


FIGURE 4 Gradient decent techniques [Color figure can be viewed at wileyonlinelibrary.com]

$$\Delta w_{ij}(t) = \eta_w \rho_j(t) x_i(t) + \alpha_w \Delta w_{ij}(t-1) \quad (29)$$

$$\Delta b_j(t) = \eta_b \rho_j(t) + \alpha_b \Delta b_j(t-1) \quad (30)$$

where $\alpha_w \Delta w_{ij}(t-1)$ and $\alpha_b \Delta b_j(t-1)$ are the momentum terms that determine the influence of the past parameter changes on the current direction of movement in the parameter space, η_w and η_b represent the learning rates, and $\rho_j(t)$ is the error signal of the j th neuron of the hidden and output layer which is back propagated in the network. $\rho_j(t)$ is calculated based on the position of the neuron as shown in Figure 4, for output layer neuron to h2 neuron the following equation is used:⁵⁵

$$\rho_j(t) = -(y_j(1 - y_j)(t_j - y_j)) \quad (31)$$

where y_j is the output of j th node in the output layer, t_j is the desired target of j th node. For other reasons, $\rho_j(t)$ is calculated using the following formula:⁵⁵

$$\rho_j(t) = -\left(y_j(1 - y_j) \sum_k w_{jk} \rho_k(t) \right) \quad (32)$$

where y_j is the output of j th node in the output layer, $\sum_k w_{jk} \rho_k(t)$ is the propagated results from hidden neuron to another hidden neuron or input neuron as shown in Figure 4, k contains the layers from output layer to the current layer j .

Finally, after this step the first iteration of the gradient descent with momentum technique is completed. The gradient descent with momentum technique uses the first order derivative of the error function and many works proved that the second-order derivative methods (i.e., Levenberg–Marquardt back propagation) have more ability to improve the convergence speed and enhance the results of the neural networks.^{48–61} As a result, Levenberg–Marquardt back propagation is used after applying gradient decent with momentum algorithm by using the weights and biases from the previous iteration to calculate the next weights and biases for the MLP and CMLP.

The standard Newton method defines a Hessian matrix as a second order derivative for the error function. Levenberg–Marquardt algorithm solves the standard Newton method by using a Gauss–Newton algorithm which is an approximation for Hessian matrix. The approximate Hessian matrix is calculated by multiplying the transpose of the Jacobian matrix with Jacobian matrix, where Jacobian matrix is the first order derivative of the error function. To find the optimal weights and biases using the Levenberg–Marquardt algorithm, the algorithm will monitor the change in the error function. If the change is rapid, the algorithm will use Gauss–Newton algorithm as follows:^{61,62}

$$W_{k+1} = W_k - (H + \mu I)^{-1} J_k e_k \quad (33)$$

where W_k and W_{k+1} are the weight matrix for the current and the next step iteration k th and $(k+1)$ th, respectively. μ is damping factor, I is identity matrix, e_k and J_k are the error vector of the k th iteration and Jacobian matrix for iteration k th, respectively. H is the Hessian matrix that calculated as:^{61,62}

$$H \approx J_k^T J_k \quad (34)$$

J_k is the Jacobian matrix for iteration k th that contains the first derivative of the error function, and T is the transpose for the J matrix.

Moreover, if the error change is not decreased the H matrix can be neglected and only the damping term (μI) is considered which will direct the algorithm to move one step closer to gradient decent algorithm. The new weight will be calculated as follows:^{51,52}

$$W_{k+1} = W_k - (\mu I)^{-1} J_k e_k \quad (35)$$

The last equation is descent gradient algorithm where gradient is equal to $J_k e_k$. Besides that, the biases will use the same equations from 32 to 34 but instead of using the weights' matrix, bias matrix will be used. After completing Levenberg-Marquardt algorithm calculations, the new weights and biases will be used to calculate the new outputs of the MLP and CMLP.

The time that taken to define and set the new weights to calculate the outputs of MLP and CMLP can be calculated by finding the time complexity of MLP and CMLP. The time complexity for single hidden layer network can be expressed by using big O notation as follows.^{63,64}

$$\text{Complexity}_{MLP} = O(d \times h) + O(h \times c) \quad (36)$$

$$\text{Complexity}_{CMLP} = O(d \times h) + O(d \times h \times c) \quad (37)$$

where d is the number of input variables, h denotes the number of hidden nodes in the hidden layer, and c denotes the number of output variables.

3 | METHODOLOGY

Using artificial intelligent controllers is desired to drive the motions of solar tracking systems. By using intelligent controllers in solar tracking systems, the performance of solar tracking systems could probably be increased. However, using intelligent drivers not guaranteed that the photovoltaic modules would perform better than traditional solar photovoltaic systems in all cases. Nevertheless, MLP network, and CMLP network principles are sufficient to be used as solar tracking systems controllers due to their effectiveness in complex and nonlinear systems which leads them to perform better than other intelligent classifiers in driving solar tracking systems.

In contrast, the current intelligent solar tracking systems suffer from many problems, which weaken the performance of the overall solar tracking systems. In addition, it would also slow down the process of predicting tilt and orientation angles, therefore increase the time complexity of the solar tracking systems. The main problems of the current solar tracking systems based on intelligent classifiers are slow convergence to reach the desired goal. This is because of the high complexity of the processes that adopted to find the desired goal. Consequently, high complexity leads to obtain a highly cost system. In addition, complex architecture of the intelligent classifiers would increase the complexity of the systems, and would increase the processing time, thus, increase the consumed energy.

Solving the problems of the current solar tracking systems based on intelligent controllers is the main motivation to develop driving solar tracking systems globally. The main target is to develop efficient intelligent solar tracking systems which would increase the performance of solar tracking controllers, produce high performance, simplify the current systems, minimize the cost of the current systems, and faster the training process.

Based on the characteristics of artificial neural network principles, MLP and CMLP can be efficiently used to drive solar tracking systems. However, the complexity of the solar tracking

controllers would be increased by increasing the number of hidden nodes in the hidden layers, and the number of hidden layers in the network. Increasing the complexity would increase the cost, processing time, and consumed energy.

As explained in the literature review, minimizing the high level of nonlinearity by decreasing the number of hidden layers and the number of hidden nodes in the hidden layers is not enough to generate that neural network can deal with data samples efficiently to produce an adaptive, efficient, low complex, low cost, and fast intelligent solar tracking systems.

Developing an intelligent, efficient, adaptive, and high performance MLP and CMLP principles to drive and control solar tracking systems is a very difficult and critical issue. However, finding such system is the main motivation globally. Finding such system would improve the field of renewable energy and would promote the use of solar energy in investment globally. Replacing fossil-fuels resources by the clean solar energy is highly demanded. This can only be applied if effectual solar tracking systems are created.

To achieve the desired target, and to develop the conventional MLP and CMLP principles, several techniques are proposed to increase the efficiency of conventional MLP and CMLP as explained earlier. Using a reference value as an input in addition to the input variables of conventional MLP and CMLP principles proved its capability to increase the performance of the solar tracking systems based on MLP and CMLP. This technique can deal with dynamic problems. In addition, it would also solve the disadvantages of the conventional systems and introduce a new generation of neural network principles.

Developing new neural network principles by combining and modifying the architecture, adding an additional input variable as reference value, and integrating the conventional neural network principles with other intelligent classifiers would optimize the performance of solar tracking systems, and introduce an effectual principle that can be used efficiently in driving solar tracking systems.

This section deals with the methodology to develop new single and dual axis solar tracking systems by integrating LR with MLP. Two neural networks principles of conventional MLP and CMLP are adopted to develop the proposed systems. In addition, two LR models including BLR and MLR are adopted to be integrated with MLP models. Both BLR and MLR are trained by using one of unsupervised clustering algorithms namely k-means, fuzzy c-means, and hierarchical clustering algorithms.

Both BLR and MLR will be trained using the best clustering algorithms. The output of the trained LR classifier is used as input data to train MLP and CMLP classifiers, along with original input data. The output of the trained LR represents the desired reference value that would increase the performance of MLP and CMLP principles. The integration of the trained LR and the trained neural network models are adopted as new solar tracking systems controllers as shown in Figure 5. By integrating LR with MLP, it is believed that the performance of the proposed intelligent neural network could be improved.

It is believed that the proposed solar tracking systems which based conventional MLP and CMLP neural network principles integrated with LR model would solve the complexity problem, faster the process to find the desired goal, increase the efficiency of solar tracking controllers, produce high performance, simplify the current systems, and minimize the cost of the current intelligent solar tracking systems. LR classifier has several advantages over other classifiers.^{65,66} As explained earlier, these advantages lead to select LR classifier to enhance the performance of conventional neural network principles as explained.

However, LR can deal with nonlinear relationship similar to linear relationship between dependent and independent variables.²³ This characteristic is very important when unknown relationships exist between dependent and independent variables.

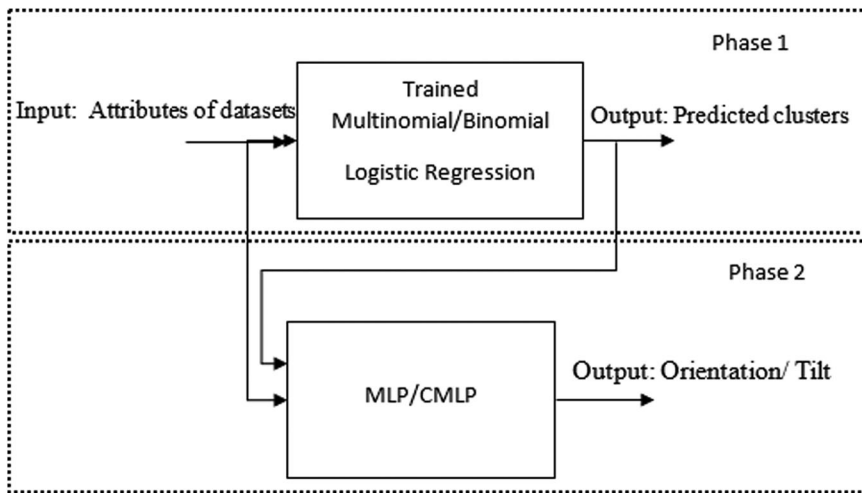


FIGURE 5 Block diagram of the developed single and dual axis solar tracking systems. CMLP, cascade multilayer perceptron; MLP, multilayer perceptron

The methodology of the proposed intelligent solar tracking systems is divided into three main phases namely selecting of the optimum clustering algorithm, training the LR which based on the selected optimum clustering algorithm, and integrating of the trained LR and a neural network to develop new single and dual axis solar tracking systems controllers. The three phases performed are explained in the following subsections.

3.1 | Phase one: Selection of the optimum clustering algorithm

This section deals with the phase of selecting the clustering algorithm that performs better compared to other selected algorithms. Three unsupervised clustering algorithms namely k-means, fuzzy c-means, and hierarchal clustering algorithms are tested to find their capability in clustering input data and investigate of the optimum clustering algorithm and optimum number of clusters that can be used to train the LR models. These clustering algorithms are selected because they are the mostly used clustering algorithms in the clustering field.

To test these clustering algorithms, a combination between one of the selected clustering algorithms and one of neural network architecture (i.e., MLP or CMLP) is adopted. The optimum number of clusters for each algorithm will be determined by testing different numbers of clusters from 2 to 10 clusters. The two clusters are the minimum number of clusters that can be used to cluster the collected data, while ten clusters are the maximum number of clusters that can be used to cluster the collected data without getting over fitting.⁴⁰ The clusters that obtained from the three clustering algorithms with number of clusters from two to ten are adopted as input variable to MLP in addition to the input variables of data set one (month, day, and time) or data set two (month, day, time, I_{sc} , V_{oc} , and power radiation variables). The block diagram that represents the processes involved in clustering phase is shown in Figure 6.

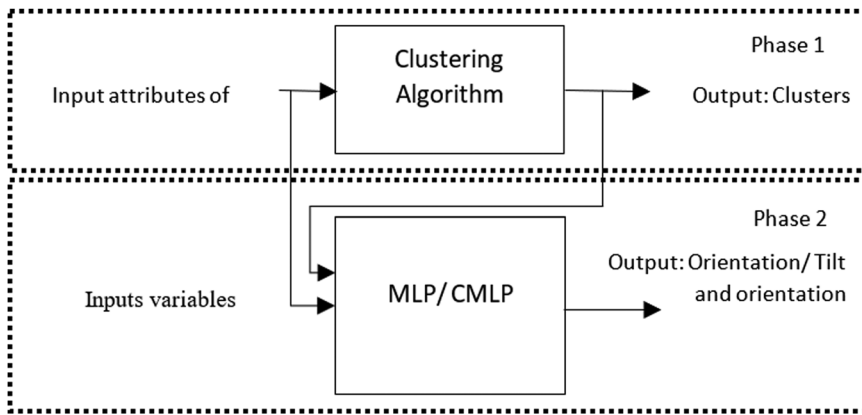


FIGURE 6 Block diagram of selecting the optimum clustering algorithm process. CMLP, cascade multilayer perceptron; MLP, multilayer perceptron

As shown in Figure 6, selecting the optimum clustering algorithm is divided into two phases. The first phase is to cluster the input attributes of data set. The second phase is to train the MLP using the obtained clusters from the first step as inputs along with the input attributes of data set one (month, day, and time) or data set two (month, day, time, I_{sc} , V_{oc} , and power radiation), while outputs are either orientation angle or tilt and orientation angles for single and dual axis solar tracking systems, respectively. The flowchart of clustering phase is shown in Figure 7.

Based on Figure 7, the first process involved is to select one of the three clustering algorithms k-means, fuzzy c-means, and hierarchical clustering. The next process is to split the input data including the clusters output from the previous process, the selected input attributes, and the target data into three sets. Seventy percent of data are used for training, 15% of data are used for validation, and 15% of data are used for testing. The developed models proposed to use a simple MLP or CMLP (i.e., with one hidden layer and optimum number of hidden nodes) to find the best clustering technique.

For training neural network, the general architecture for the proposed networks is initialized including the weights, biases, error target, and transfer functions. In the proposed network, weights and biases are initialized randomly, error target is predefined as 10^{-15} , and both hyperbolic tangent and identity transfer functions are used to calculate the output of the hidden layers nodes and the output layer nodes, respectively. The network is trained to tune error and biases to the optimum values.

The next process is to test and evaluate the performance of the trained network. Both MSE and prediction rate are performed criteria to evaluate MLP and CMLP principles. MSE should be minimized to optimize the developed neural networks. MSE can be calculated by using Equation (38).⁶⁶

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (38)$$

where N is the number of samples, \hat{y}_i is the predicted output values, and y_i is the original output values.

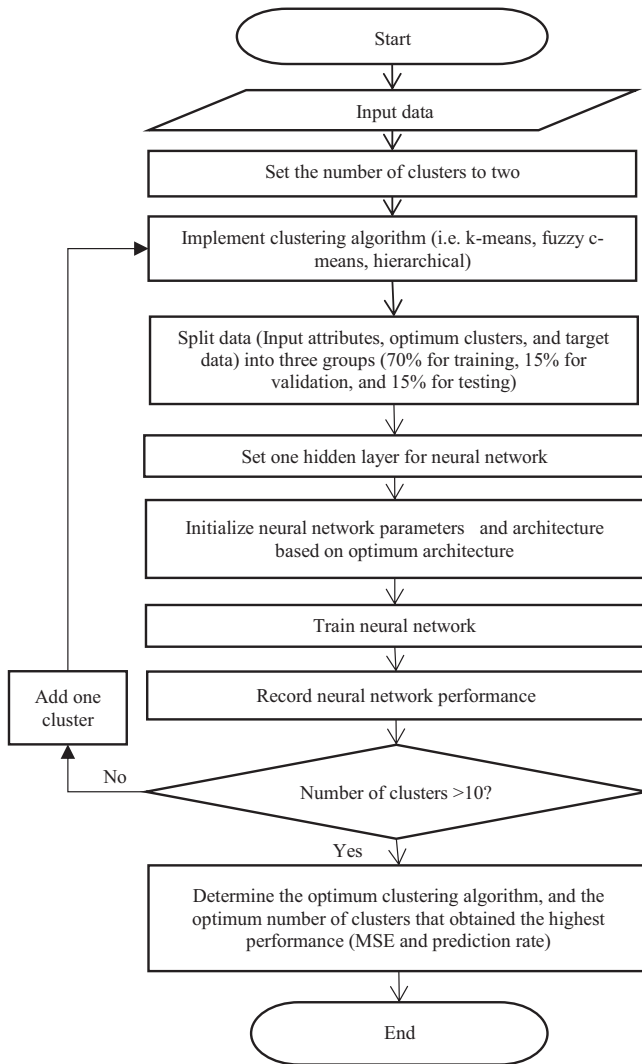


FIGURE 7 Flowchart of clustering phase

Prediction rate measures the percentage of data samples that can be predicted by the used predictor. The prediction rate can be calculated by using Equation (39).⁶⁷

$$\text{Prediction rate} = \left(1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \right) \times 100\% \quad (39)$$

where \bar{y}_i is the mean of output values.

Then, the optimum clustering algorithm, and the optimum number of clusters that achieved the highest performance are selected for both MLP and CMLP separately. Finally, at the end of this process the optimum clustering algorithm is used to train the LR classifier.

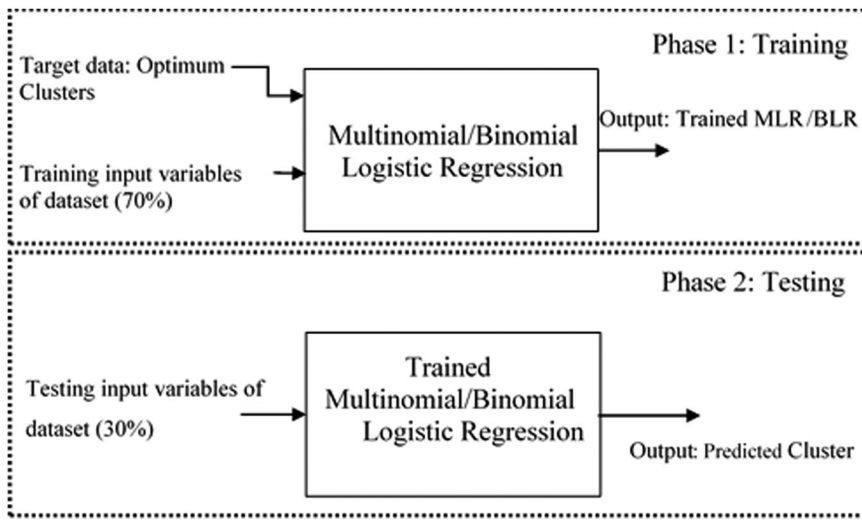


FIGURE 8 Block diagram of MLR/BLR. BLR, binomial logistic regression; MLR, multinomial logistic regression

3.2 | Phase two: Training of LR based on the optimum clustering algorithm

Both BLR and MLR are employed to classify the input attributes of single and dual axis solar tracking systems. The LR models are trained to be used as a predictor to predict the appropriate cluster for data attributes.

This section uses the selected clustering algorithm and optimum number of clusters to train the LR models. This process prepares the LR models to predict the appropriate and significant cluster classification for any sample of data. The idea of employing this stage is to faster the process of classifying a sample of data as well as help in improving the performance of MLP and CMLP. Both MLR and BLR classifiers are used separately. MLR and BLR models are selected as LR classifiers based on statistical rules as explained in Table 3.^{68–70} The block diagram to train MLR and BLR models is shown in Figure 8.

As shown in Figure 8, the MLR and BLR classifiers follow the same procedure. Two phases are involved. The first phase is to train the MLR and BLR by using 70% of the data samples of selected datasets as inputs, in addition to the clusters output from the clustering algorithm in clustering phase as a target data. The second phase is to test the trained MLR and BLR to predict the suitable clusters for the rest of 30% of the data samples of datasets accurately. This would faster the process of classifying any sample of data in the future. Testing the MLR and BLR is very important process to ensure and validate the performance of the trained MLR and BLR, and to evaluate its capability to classify new samples of data accurately. Based on the results of selecting the optimum number of clusters and the statistical rules that explained in Table 3, a BLR model is used when the optimum number of clusters that adopted to develop solar tracking system is two. In contrast, a MLR is used when the optimum number of clusters that adopted to develop solar tracking system is between three and ten. Figure 9 shows the flowchart of training and testing MLR and BLR.

As shown in Figure 9, the first process involved is to split the data samples into two sets. Seventy percent of data are used for training and 30% of data are used for testing. The second process involved

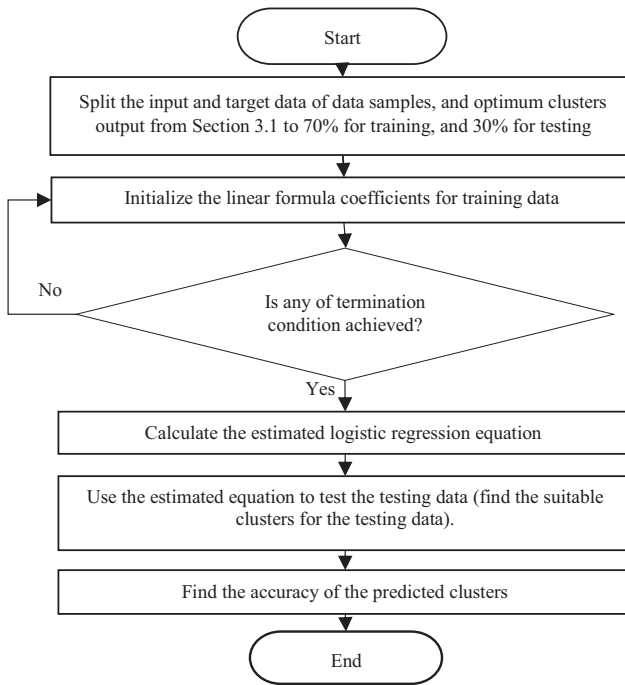


FIGURE 9 Flowchart of training and testing MLR and BLR. BLR, binomial logistic regression; MLR, multinomial logistic regression

is to initialize the linear formula coefficients to be used in training the data attributes. In this process, the linear formula coefficients for the training data are calculated for both MLR and BLR.

The next process is to check whether any of the termination conditions is fulfilled. The estimated LR equation is calculated when any of termination conditions is achieved. This equation is used to test the testing data and to find the predicted cluster for each sample of data. Then, the accuracy of MLR and BLR are calculated to ensure about the capability of MLR and BLR to predict the suitable cluster for any sample of data accurately.

Accuracy performance criteria is used to evaluate the capability of MLR and BLR to predict the suitable cluster for any sample of data accurately. The accuracy can be calculated by using Equation (40).⁷¹

$$\text{Accuracy(\%)} = \frac{\text{Total number of correct clusters}}{\text{Total number of data}} \times 100\% \quad (40)$$

The trained MLR and BLR are used to develop single and dual axis solar tracking systems as shown in the following section.

3.3 | Phase three: Integration of LR and MLP for single and dual axis solar tracking systems

This section explains the process of developing the conventional MLP and CMLP by integrating them with one of the trained MLR or BLR classifiers. The proposed models will be compared to the conventional MLP and CMLP neural networks-based systems. This will help in proving the

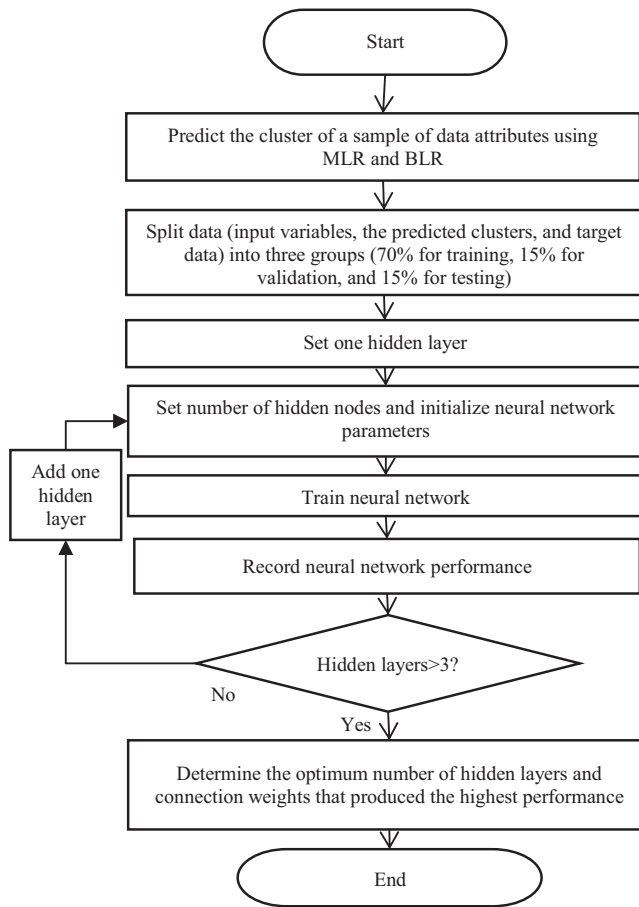


FIGURE 10 The flowchart of the developed single and dual axis solar trackers using MLP/CMLP. BLR, binomial logistic regression; CMLP, cascade multilayer perceptron; MLP, multilayer perceptron; MLR, multinomial logistic regression

effectiveness of integrating the trained LR to the conventional neural network. In addition, it would also evaluate the proposed technique in improving the performance of solar trackers controllers. The flowchart of the developed single and dual axis trackers is shown in Figure 10.

As shown in Figure 10, the first process involved is to use the MLR or BLR to predict the suitable cluster of a given sample of data. This cluster will be used as input to MLP and CMLP along with the input variables. The second process is to split the input data into training, validation, and testing sets. The input data include the predicted clusters output from the previous process, the input variables of the selected solar tracking models, and target data of orientation angle for single-axis solar tracker and both tilt and orientation angles for dual-axis solar tracker. Seventy percent of data are used for training, 15% of data are used for validation, and 15% of data are used for testing as well.

The third process is to define the general architecture of the neural network including the number of hidden layers and number of hidden nodes. One hidden layer is performed first with the optimum number of hidden nodes that that recommended by AL-Rousan et al.² Then, the weights and biases parameters are randomly initialized. The error target is defined as 10^{-15} . The hyperbolic tangent and identity transfer functions are used as transfer functions for the output of the hidden layers nodes and the output layer nodes, respectively.

The next process is to train the neural network, then to test the testing data to evaluate the performance and to ensure about the capability of the trained neural network. The performance results of the tested neural network are recorded. All the previous processes are repeated for two and three hidden layers. The optimum scenario that achieved the highest performance is adopted to develop single and dual-axis solar tracking systems separately.

4 | EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the results of integrating MLP networks with trained LR models to develop new single and dual axis solar tracking system controllers. To achieve the goals of this study, practical data will be used. The data were collected by using a real mechanical solar tracking prototype. A high efficiency multicrystalline photovoltaic module of model KC 120-1 was used to collect the data periodically. These data were recorded for several selected days based on weather conditions from 8:00 am to 5:00 pm. Real based data sets were recorded for several variables including time, day, month, I_{sc} , V_{oc} , power radiation, tilt angle, and orientation angle. Two data sets are adopted to predict the orientation in single axis solar tracking or both tilt and orientation angles in dual-axis solar tracking.⁷² Month, day, and time variables were selected as data set one, and month, day, time, I_{sc} , V_{oc} , and power radiation variables were selected as data set two. This section is divided into three subsections to present the results of the developed MLP and CMLP based systems.

4.1 | Results of phase one: Selection of the optimum clustering algorithm

The main target of this phase is to select the optimum clustering algorithms and optimum number of clusters that could achieve the highest performance. This section is divided into two

TABLE 4 The prediction rate (%) and MSE results of MLP and the three clustering algorithms of single-axis solar tracking system using data set one

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	96.67	96.85	96.79	0.0032	0.0034	0.0028
3	96.55	96.75	97.59	0.0033	0.0025	0.0025
4	96.59	97.11	97.02	0.0030	0.0029	0.0071
5	96.91	96.71	96.48	0.0038	0.0028	0.0034
6	96.43	97.05	97.05	0.0052	0.0026	0.0028
7	97.66	96.93	96.20	0.0024	0.0027	0.0030
8	96.51	96.78	96.98	0.0025	0.0026	0.0029
9	96.19	97.43	96.16	0.0029	0.0028	0.0049
10	96.54	96.60	97.30	0.0026	0.0083	0.0029

TABLE 5 The overall prediction rate (%) and MSE results of MLP and the three clustering algorithms of single-axis solar tracking system using data set two

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	95.80	92.40	97.50	0.0013	0.0027	0.0007
3	94.80	94.60	94.70	0.0010	0.0012	0.0019
4	96.30	96.10	95.90	0.0007	0.0016	0.0007
5	96.48	97.40	97.95	0.0005	0.0006	0.0007
6	96.40	96.90	95.10	0.0009	0.0010	0.0011
7	97.10	95.80	97.00	0.0012	0.0010	0.0009
8	96.40	95.40	96.30	0.0006	0.0018	0.0011
9	95.90	97.60	96.80	0.0008	0.0010	0.0006
10	97.30	93.30	96.00	0.0006	0.0010	0.0009

TABLE 6 The prediction rate (%) and MSE results of MLP and the three clustering algorithms of dual-axis solar tracking system using data set one

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	96.80	96.96	96.05	0.0010	0.0014	0.0016
3	95.73	95.58	96.20	0.0017	0.0024	0.0017
4	94.50	95.50	96.01	0.0037	0.0023	0.0027
5	93.23	96.10	96.39	0.0028	0.0018	0.0014
6	94.41	93.95	96.34	0.0022	0.0033	0.0013
7	94.16	96.13	96.18	0.0026	0.0017	0.0014
8	95.71	96.02	96.22	0.0020	0.0015	0.0024
9	93.33	95.55	96.71	0.0038	0.0019	0.0010
10	95.29	95.64	96.21	0.0020	0.0024	0.0014

parts to present the proposed MLP and CMLP based solar tracking systems. The optimum clustering algorithms for both MLP and CMLP are presented in the following sections.

4.1.1 | Selection of the optimum clustering algorithm for MLP network

To develop new single and dual axis solar tracking systems, the first process is to evaluate the conventional MLP network with one of the three clustering algorithms. The most effective clustering algorithm that can improve the performance of the conventional MLP for both single and

TABLE 7 The overall prediction rate (%) and MSE results of MLP and the three clustering algorithms of dual-axis solar tracking system using data set two

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	94.86	96.70	94.90	0.0007	0.0008	0.0011
3	94.90	94.70	93.60	0.0014	0.0013	0.0009
4	93.50	96.15	94.30	0.0014	0.0007	0.0012
5	95.05	95.40	93.20	0.0014	0.0014	0.0019
6	94.06	94.50	96.60	0.0013	0.0011	0.0009
7	93.80	93.60	94.30	0.0015	0.0016	0.0016
8	94.60	96.20	95.80	0.0010	0.0010	0.0012
9	95.40	92.60	93.20	0.0013	0.0013	0.0012
10	94.03	95.20	93.02	0.0014	0.0010	0.0012

dual axis solar tracking systems will be selected. Then, the optimum number of clusters that achieved the highest performance are adopted to implement the proposed solar tracking systems.

The prediction rates and MSE of the three clustering algorithms with conventional MLP for single and dual axis systems are shown in Tables 4–7.

Tables 4 and 5 present the single-axis solar tracking system results for data set one and data set two respectively. While Tables 6 and 7 present the dual-axis solar tracking system results for data set one and data set two, respectively. As shown in Table 4, the results revealed that k-means clustering algorithm with seven clusters predicted the highest prediction rate and the lowest MSE. K-means clustering algorithm predicted 97.66% of data set, while fuzzy c-means and hierarchical clustering algorithms obtained the highest prediction rates of 97.43% and 97.30% when nine and ten clusters are used respectively. Moreover, the MSE of k-means clustering with seven clusters is 0.0024, while the optimum MSE values for fuzzy c-means and hierarchical clustering algorithms are 0.0025 when three clusters are used for both cases. The results showed that the three clustering algorithms achieved very close prediction rates and MSE results. Comparing the MSE values, using k-means clustering algorithm with seven clusters is the optimum clustering algorithm to develop a single-axis solar tracking system for data set one.

On the other hand, for data set two that tested on single-axis solar tracking system, the results in Table 5 show that hierarchal clustering algorithm predicted the highest prediction rate of 97.95%, when five clusters are used. Fuzzy c-means clustering algorithm with nine clusters predicted 97.60% of data set two, and k-means clustering algorithm using 10 clusters predicted the worst amount of data with prediction rate of 97.3%, while the MSE of k-means clustering algorithm with five clusters (i.e., 0.0005) is the lowest MSE as compared to fuzzy c-means and hierarchical clustering algorithms that achieved MSE of 0.0006 using five and nine clusters, respectively. Moreover, the prediction rate of k-means clustering algorithm with five clusters is 96.48%. The overall prediction rate and MSE results revealed that using k-means clustering algorithm with five clusters is better than using both fuzzy c-means and hierarchical clustering algorithms.

In contrary, Table 6 presents the results for dual-axis solar tracking system using data set one. The results revealed that fuzzy c-means clustering algorithm predicted the highest amount

of data with a prediction rate of 96.96% for two clusters. K-means clustering algorithm produced very close prediction rate of 96.80% with two clusters, and hierarchical clustering algorithm predicted the worst with prediction rate of 96.71%. However, the three clustering algorithms predicted close prediction rates. Table 6 shows that both k-means clustering and hierarchical clustering algorithms achieved the lowest MSE of 0.0010 when two and nine clusters are used, respectively. While fuzzy c-means clustering algorithm obtained higher MSE of 0.0014 when two clusters are used. Thus, the best combination to improve the performance of MLP is by using k-means clustering algorithm with conventional MLP. The optimum number of clusters that produced the lowest MSE and a high prediction rate is two clusters. Referring to the data distribution that obtained by using two clusters which is the optimum number of clusters, further analysis found that the best distribution is based on the half-month. Half-month is a unit of time typically used in astronomy.^{73–75} The first half of each month is clustered to cluster one, while the second half of each month is clustered to cluster two.

For dual-axis solar tracking system, Table 7 presents the results of using data set two to predict tilt and orientation angles. The results revealed that fuzzy c-means clustering algorithm predicted the highest prediction rate of 96.70%. In addition, hierarchical and k-means clustering algorithms predicted 96.60% and 95.40% of data set, respectively. On the other hand, the optimum MSE is 0.0007 that achieved when k-means and fuzzy c-means clustering algorithms are used with two and four clusters, respectively. Hierarchical clustering algorithm achieved the worst MSE of 0.0009 when six clusters are used. Fuzzy c-means and hierarchical clustering algorithms achieved very close prediction rate of 96.70% and 96.60%, respectively. According to the MSE and referring to the complexity of the three algorithms, it is found that using two clusters of k-means clustering algorithm is better than using fuzzy c-means and hierarchical clustering algorithms.

Referring to Table 2 and comparing both prediction rate and MSE values for the three algorithms and all number of clusters, the results revealed that the best combination to improve the performance of the neural network and then improve single and dual axis solar tracking systems is using K-means clustering algorithm with conventional MLP. The optimum

TABLE 8 The prediction rate (%) and mse results of cmlp and the three clustering algorithms of single-axis solar tracking system using data set one

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	97.03	96.83	96.62	0.0022	0.0027	0.0029
3	96.63	96.44	96.22	0.0030	0.0022	0.0023
4	97.60	96.01	96.72	0.0016	0.0031	0.0020
5	96.71	96.98	96.54	0.0025	0.0024	0.0023
6	96.81	96.15	97.05	0.0016	0.0019	0.0032
7	96.06	96.44	96.07	0.0021	0.0020	0.0032
8	97.07	96.55	97.11	0.0022	0.0027	0.0023
9	97.12	96.37	97.33	0.0023	0.0034	0.0017
10	96.78	96.60	97.26	0.0021	0.0040	0.0018

TABLE 9 The overall prediction rate (%) and MSE results of CMLP and the three clustering algorithms of single-axis solar tracking system using data set two

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	94.90	93.80	93.80	0.0027	0.0012	0.0048
3	93.30	94.38	95.40	0.0028	0.0013	0.0040
4	93.80	94.60	93.10	0.0038	0.0008	0.0037
5	96.45	96.10	95.40	0.0042	0.0005	0.0025
6	94.30	95.45	93.03	0.0017	0.0006	0.0043
7	94.40	95.70	94.50	0.0020	0.0010	0.0040
8	94.40	95.80	95.60	0.0024	0.0009	0.0021
9	93.75	95.60	90.00	0.0026	0.0006	0.0037
10	95.80	92.60	95.20	0.0003	0.0008	0.0024

TABLE 10 The overall prediction rate (%) and MSE results of CMLP and the three clustering algorithms of dual-axis solar tracking system using data set one

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	96.29	97.55	96.13	0.0016	0.0008	0.0021
3	96.13	96.42	97.28	0.0013	0.0010	0.0011
4	96.50	97.43	95.87	0.0022	0.0008	0.0015
5	97.12	97.39	96.57	0.0013	0.0009	0.0010
6	97.02	97.62	97.27	0.0009	0.0008	0.0011
7	97.10	97.49	96.99	0.0018	0.0008	0.0015
8	97.71	97.47	97.01	0.0017	0.0008	0.0008
9	96.86	97.57	96.63	0.0013	0.0010	0.0007
10	97.06	97.50	97.56	0.0005	0.0007	0.0012

clustering algorithm and the optimum number of clusters should produce the lowest MSE with high prediction rate.

4.1.2 | Selection of the optimum clustering algorithm for CMLP network

Similar to MLP network procedure, the performance of conventional CMLP network with the three clustering algorithms are evaluated. The most effective clustering algorithms that can improve the conventional CMLP performance are selected. Then, the optimum number of

TABLE 11 The overall prediction rate (%) and MSE results of CMLP and the three clustering algorithms of dual-axis solar tracking system using data set two

Number of clusters	Clustering algorithm					
	Prediction rate (%)			MSE		
	K-means	Fuzzy c-means	Hierarchical	K-means	Fuzzy c-means	Hierarchical
2	94.30	94.60	95.97	0.0013	0.0012	0.0008
3	95.01	94.70	93.70	0.0009	0.0011	0.0010
4	95.60	93.70	95.74	0.0006	0.0014	0.0007
5	94.61	96.59	95.35	0.0011	0.0008	0.0008
6	93.10	95.38	94.03	0.0014	0.0010	0.0013
7	92.40	95.20	95.50	0.0015	0.0011	0.0007
8	94.80	94.90	94.90	0.0013	0.0009	0.0009
9	94.70	94.30	95.20	0.0009	0.0012	0.0010
10	95.10	95.70	91.40	0.0012	0.0011	0.0016

clusters that achieved the highest performance are adopted to implement the proposed solar tracking systems. The prediction rates and the MSE results of the three algorithms with CMLP for single and dual axis systems are shown in Tables 8–11.

Table 8 presents the results of data set one to predict orientation angle in single-axis solar tracking system. The results revealed that using four and six clusters of k-means clustering algorithm achieved the lowest MSE of 0.0016. The MSE of hierarchical and fuzzy c-means clustering algorithms are 0.0017 and 0.0019, respectively. On the other hand, k-means clustering algorithm predicted 97.60%, and 96.81% of data set for four and six clusters, respectively. While hierarchical and fuzzy c-means clustering algorithms predicted 97.33% and 96.98% for nine and five clusters, respectively. Comparing the MSE of four and six clusters of k-means clustering, using four clusters has better performance to predict the highest amount of data set and the lowest MSE. Further analysis to the data distribution that obtained by using k-means clustering algorithm with four clusters, which is the optimum number of clusters, it found that the best distribution is based on dividing each month into four quarters as described earlier. The data samples were distributed to the four clusters. The data that taken in the third and fourth quarters were clustered to clusters one and four respectively. While the data that measured in the first and second quarters were clusters to clusters two and three such that cluster two contains the data that measured from early morning at 8:00 a.m. until the day's noon at 1:00 p.m., and cluster three contains the data that measured from after noon until 5:00 p.m.

Table 9 shows the results of using data set two to predict orientation angle in single-axis solar tracking system. The results revealed that k-means clustering algorithm with five clusters predicted the highest prediction rate of 96.45%. Fuzzy c-means clustering algorithm with five clusters obtained very close prediction rate of 96.10%. While hierarchical clustering algorithm with eight clusters predicted the worst amount of data with prediction rate of 95.60%. On the other hand, the MSE of k-means clustering algorithm with ten clusters is 0.0003 which is the lowest MSE. Fuzzy c-means and hierarchical clustering algorithms achieved MSE of 0.0005, and 0.0021 with five and eight clusters, respectively. Comparing the optimum results of

prediction rates and MSE of the three algorithms, using k-means clustering algorithm with ten clusters performed better than other algorithms.

In contrary, Table 10 presents the results of predicting both tilt and orientation angles in dual-axis solar tracking system using data set one. The results revealed that k-means clustering algorithm predicted the highest amount of data with a prediction rate of 97.71% when eight clusters are used. While fuzzy c-means clustering algorithm predicted close amount of data with a prediction rate of 97.62% when six clusters are used.

Moreover, the hierarchical clustering algorithm predicted 97.27% with six clusters as well. To select the optimum clustering algorithm and the optimum number of clusters, MSE performance function is adopted.

K-means clustering algorithm achieved the lowest MSE of 0.0005 for 10 clusters. Fuzzy c-means clustering algorithm achieved low MSE of 0.0008 for six clusters, while the hierarchical clustering algorithm obtained 0.0011 MSE for two and six clusters. Referring to Table 10, and comparing the overall results of the prediction rate and MSE, the results showed that using ten clusters of k-means clustering algorithm obtained the lowest MSE and a high prediction rate. In addition, this prediction rate is very close to the highest prediction rate that obtained using eight clusters. The results revealed that the best clustering algorithm that can optimize the performance of CMLP network is k-means clustering algorithm. While the optimum number of clusters that can improve the performance is 10 clusters. Furthermore, the data distributed in the 10 clusters based on the time of the day.

The time of each day was divided into two parts based on the position of the sun across the sky. The time from early morning at 8:00 a.m. until the day's noon at 1:00 p.m. is the first part, while the time after noon until 5:00 p.m. is the second part. Data that collected in several half-months and prenoon time are clustered to clusters one, two, four, six, and eight. In contrast, the data that collected after noon are clustered to clusters three, five, seven, nine, and ten. This indicate the relationship between clusters that selected to build the proposed systems.

On the other hand, Table 11 shows the results of using data set two to predict tilt and orientation angles in dual-axis solar tracking system. The results revealed that fuzzy c-means clustering predicted 96.59% of data when five clusters are used. This prediction rate is the highest prediction rate compared to both hierarchical and k-means clustering algorithms that predicted 95.97% and 95.60% with two and four clusters, respectively. In contrast, using four clusters of k-means clustering algorithm achieved the lowest MSE of 0.0006. Hierarchical and fuzzy c-means clustering algorithms achieved 0.0007 and 0.0009 for seven and eight clusters, respectively.

Moreover, comparing the results of both MLP and CMLP networks showed that the CMLP predicted higher of data compared to MLP. This is probably because of CMLP network contains a direct weighted connection from input to output layers, which allows it to learn higher complex patterns as compared to MLP network.^{76,77} These additional direct connections also provide linear connections that CMLP network could represent both linear and nonlinear relationships.

4.2 | Results of phase two: Training of LR based on the optimum clustering algorithm

This section presents the appropriate LR model that can be integrated to MLP and CMLP networks to develop the proposed solar tracking systems. In addition, it checks the capability of the selected LR models in classifying the input data into clusters. The results of the trained LR

TABLE 12 The optimum combination of MLP and MLR/BLR to develop single and dual axis solar tracking systems using data set one and data set two

Data set	Number of axis	Number of clusters	Suitable logistic regression model
Data set one	Single-axis	7	MLR
Data set one	Dual-axis	2	BLR
Data set two	Single-axis	5	MLR
Data set two	Dual-axis	2	BLR

Abbreviations: BLR, binomial logistic regression; MLR, multinomial logistic regression.

TABLE 13 Confusion matrix for MLR that trained by k-means clustering when k=7 by using data set one for single-axis solar tracking system

Cluster Number	Predicted							Percent (%)
	1	2	3	4	5	6	7	
1	35	0	0	0	0	0	0	100
2	0	8	0	0	0	0	0	100
3	0	0	26	0	0	0	0	100
4	0	0	0	20	0	0	0	100
5	0	0	0	0	29	0	0	100
6	0	0	0	0	0	10	0	100
7	0	0	0	0	0	0	25	100
Overall (%)	22.79	5.22	17.01	13.2	18.9	6.54	16.34	100

models for both MLP and CMLP networks using data set one and data set two are presented in the following sections.

To test the accuracy of BLR and MLR when k-means clustering algorithm is used to train them with the selected numbers of clusters, the data samples are split into two parts (i.e., training and testing). Seventy percent of data are used to train the MLR and BLR, while 30% of data are used to test the accuracy of MLR and BLR. The performance of MLR and BLR in classifying the input variables of data set one and data set two is determined by finding the confusion matrix for overall data, therefore, test the accuracy of the LR model. The results of the trained LR models for both MLP and CMLP networks are presented in the following sections.

4.2.1 | LR based on the optimum clustering algorithm for MLP network

The selected optimum clustering algorithms and optimum number of clusters of MLP network are used to train the LR models. The trained LR models are used to classify the input variables into predefined clusters. Table 12 shows the optimum combination between the LR models and the MLP network that is adopted to develop both single and dual axis solar tracking systems using the data set one and data set two.

TABLE 14 Confusion matrix for BLR that trained by k-means clustering when $k=2$ by using data set one for dual-axis solar tracking system

Cluster number	Predicted		Percent (%)
	1	2	
1	90	0	100
2	0	63	100
Overall (%)	58.83	41.17	100

TABLE 15 Confusion matrix for MLR that trained by k-means clustering when $k = 5$ by using data set two for single-axis solar tracking system

Cluster number	Predicted					Percent (%)
	1	2	3	4	5	
1	35	0	0	0	0	100
2	0	20	0	0	0	100
3	0	0	18	0	0	100
4	0	0	0	40	0	100
5	0	0	0	0	40	100
Overall (%)	22.88	13.08	11.76	26.14	26.14	100

TABLE 16 Confusion matrix for BLR that trained by k-means clustering when $k = 2$ by using data set two for dual-axis solar tracking system

Cluster number	Predicted		Percent (%)
	1	2	
1	90	0	100
2	0	63	100
Overall (%)	58.83	41.17	100

As shown in Table 12, for single-axis solar tracking system by using data set one, a MLR model is trained by seven clusters of k-means clustering algorithm. In contrast, for dual-axis solar tracking system by using data set one, a BLR model is trained by two clusters.

However, for single-axis solar tracking system by using data set two, a MLR model is trained by five clusters of k-means clustering algorithm. While dual-axis solar tracking system is trained by a (BLR) model for data set two. Tables 13–16 show the confusion matrix for both MLR and BLR for both single and dual axis solar tracking systems. The confusion matrix is a table that used to describe the performance of a classifier.^{78–80}

TABLE 17 The optimum combination of MLP and MLR/BLR to develop single and dual axis solar tracking systems

Data set	Number of axis	Number of clusters	Suitable logistic regression model
Data set one	Single-axis	4	MLR
Data set one	Dual-axis	10	MLR
Data set two	Single-axis	10	MLR
Data set two	Dual-axis	4	MLR

Abbreviations: BLR, binomial logistic regression; MLR, multinomial logistic regression.

TABLE 18 Confusion matrix for MLR that trained by k-means clustering when $k = 4$ by using data set one for single-axis solar tracking system

Cluster number	Predicted group				Percent (%)
	1	2	3	4	
1	39	0	0	0	100
2	0	40	0	0	100
3	0	0	42	0	100
4	0	0	0	32	100
Overall (%)	25.49	26.14	27.45	20.92	100

TABLE 19 Confusion matrix for MLR that trained by k-means clustering when $k = 10$ by using data set one for dual-axis solar tracking system

Cluster number	Predicted										Percent (%)
	1	2	3	4	5	6	7	8	9	10	
1	8	0	0	0	0	0	0	0	0	0	100
2	0	16	0	0	0	0	0	0	0	0	100
3	0	0	10	0	0	0	0	0	0	0	100
4	0	0	0	16	0	0	0	0	0	0	100
5	0	0	0	0	15	0	0	0	0	0	100
6	0	0	0	0	0	12	0	0	0	0	100
7	0	0	0	0	0	0	20	0	0	0	100
8	0	0	0	0	0	0	0	16	0	0	100
9	0	0	0	0	0	0	0	0	30	0	100
10	0	0	0	0	0	0	0	0	0	10	100
Overall (%)	5.2	10.5	6.5	10.5	9.8	7.8	13.1	10.5	19.6	6.5	100

TABLE 20 Confusion matrix for MLR that trained by k-means clustering when $k = 10$ by using data set two for single-axis solar tracking system

Cluster number	Predicted										Percent (%)
	1	2	3	4	5	6	7	8	9	10	
1	20	0	0	0	0	0	0	0	0	0	100
2	0	20	0	0	0	0	0	0	0	0	100
3	0	0	8	0	0	0	0	0	0	0	100
4	0	0	0	8	0	0	0	0	0	0	100
5	0	0	0	0	10	0	0	0	0	0	100
6	0	0	0	0	0	20	0	0	0	0	100
7	0	0	0	0	0	0	15	0	0	0	100
8	0	0	0	0	0	0	0	12	0	0	100
9	0	0	0	0	0	0	0	0	20	0	100
10	0	0	0	0	0	0	0	0	0	20	100
Overall (%)	13.1	13.1	5.2	5.2	6.6	13.1	9.8	7.7	13.1	13.1	100

TABLE 21 Confusion matrix for MLR that trained by k-means clustering when $k = 4$ by using data set two for dual-axis solar tracking system

Cluster number	Predicted				Percent (%)
	1	2	3	4	
1	42	0	0	0	100
2	0	39	0	0	100
3	0	0	40	0	100
4	0	0	0	32	100
Overall (%)	27.45	25.49	26.14	20.92	100

As shown in Tables 13–16, the accuracy of MLR and BLR classifiers are 100% when the number of clusters that used to train MLR is seven, five, and two. Since both MLR and BLR successfully could predict all the classes of training data, thus both MLR and BLR would be used efficiently in finding the confidence classes to incorporate new training data into the model. This process would enhance the performance of the developed single and dual axis solar tracking systems when combined with MLP network.

4.2.2 | LR based on the optimum clustering algorithm for CMLP network

Similar to Table 12, Table 17 shows the optimum combination between the LR models and the CMLP network that is adopted to develop both single and dual axis solar tracking systems using data set one and data set two.

As shown in Table 17, and by using data set one, CMLR models are trained by four and ten clusters for single and dual axis solar tracking systems respectively. While by using data set two, CMLR models are trained by ten and four clusters for both single and dual axis solar tracking systems respectively. Tables 18–21 show the confusion matrix for both CMLR and MLR for both single and dual axis solar tracking systems, respectively.

As shown in Tables 18–21, the accuracy of CMLR classifier is 100% when the number of clusters that used to train MLR is four and ten. These results proved that MLR models are capable to find the confidence classes to incorporate new training data into the model. Thus, it would efficiently be used in developing efficient single and dual axis solar tracking systems when combined with CMLP network. While the accuracy of CMLR classifier is 100% when the number of clusters that used to train BLR is two. These results proved that BLR models are capable to find the confidence classes to incorporate new training data into the model and would efficiently be used in developing efficient single and dual axis solar tracking systems when combined with CMLP network as well.

4.3 | Results of phase three: Integration of LR and MLP/CMLP for single and dual axis tracking systems

Based on the results of previous sections, efficient single and dual axis solar tracking systems are developed. A combination of MLP or CMLP neural network and a trained BLR or MLR by k-means clustering algorithm with different number of clusters is used. The results of the developed single and dual axis solar tracking systems are compared to the results of conventional MLP and CMLP to evaluate the performance of the developed systems. Both prediction rate and MSE performance metrics are used to evaluate the developed solar tracking systems.

4.3.1 | Results of developed single and dual axis solar tracking system based on MLP

The block diagram of the developed single and dual axis solar tracking systems based on combining MLP neural network and BLR or MLR model is shown in Figure 11. As shown in Figure 11, the developed single or dual axis solar tracking system will use the predicted cluster from the trained LR model as input variable to the MLP neural network along with the input

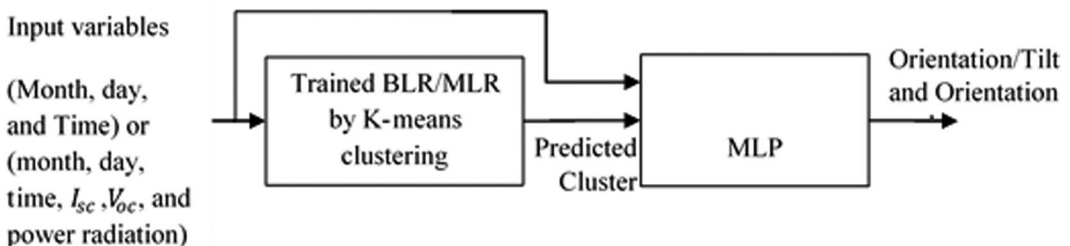


FIGURE 11 Block diagram of the developed single and dual axis solar tracking systems based on BLR/MLR and MLP. BLR, binomial logistic regression; MLP, multilayer perceptron; MLR, multinomial logistic regression

variables of month, day, and time variables as data set one, while month, day, time, I_{sc} , V_{oc} , and power radiation variables are used as data set two.

The developed single and dual axis solar tracking systems are tested, analyzed, and compared with conventional MLP network by following the methodology that explained earlier. Three different scenarios with one, two, and three hidden layers are tested to find the prediction rate and MSE. Tables 22 and 23 show the MSE and the prediction rate of the developed single-axis and dual-axis solar tracking systems respectively when data set one is used.

As shown in Tables 22 and 23, the overall prediction rate that obtained by using the developed single and dual axis solar tracking systems based on combining MLP with MLR and BLR is higher than the prediction rate of conventional MLP based system. Moreover, using one hidden layer in the developed single-axis tracking system achieved higher prediction rate compared to the three scenarios of one, two, and three hidden layers in conventional MLP based system. One hidden layer of MLP and MLR obtained a prediction rate of 96.78%, 96.37%, and 99.13% for one, two, and three hidden layers, respectively compared to 92.5%, 95.06, and 96.75% for one, two, and three hidden layers of conventional MLP, respectively.

Moreover, using one hidden layer in the developed single-axis solar tracking system obtained higher performance compared with three hidden layers of conventional MLP. The MSE of one, two, and three hidden layers are 0.240×10^{-2} , 0.130×10^{-2} , and 0.010×10^{-2} compared to 0.890×10^{-2} , 0.370×10^{-2} , and 0.250×10^{-2} , respectively. The prediction rate of one hidden layer in the developed MLP-MLR is approximately equal to the prediction rate of three hidden layers of conventional MLP. While the MSE of one hidden layer of the developed MLP-MLR is better than using three hidden layers of conventional MLP.

On the other hand, using BLR to develop dual-axis solar tracking system obtained a prediction rate of 96.80%, 98.64%, and 98.62% for one, two, and three hidden layers, respectively. In addition, the developed dual-axis solar tracking system achieved very low MSE of 0.100×10^{-2} , 2.070×10^{-4} , and 1.380×10^{-4} for one, two, and three hidden layers, respectively. One hidden

TABLE 22 The prediction rate (%) and MSE results of the developed dual-axis solar tracking system based on combining MLP and MLR when $K = 7$ by using data set one

Single-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional MLP	92.50	95.06	96.75	0.890×10^{-2}	0.370×10^{-2}	0.250×10^{-2}
Developed-MLP with MLR	96.78	96.37	99.13	0.240×10^{-2}	0.130×10^{-2}	0.010×10^{-2}

Abbreviations: MLP, multilayer perceptron; MLR, multinomial logistic regression.

TABLE 23 The prediction rate (%) and MSE results of the developed dual-axis solar tracking system based on combining MLP and BLR when $K = 2$ by using data set one

Single-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional MLP	95.49	96.13	96.68	0.280×10^{-2}	0.220×10^{-2}	0.160×10^{-2}
Developed-MLP with BLR	96.80	98.64	98.62	0.100×10^{-2}	2.070×10^{-4}	1.380×10^{-4}

Abbreviations: MLP, multilayer perceptron; MLR, multinomial logistic regression.

TABLE 24 The overall prediction rate (%) and MSE results of the developed single-axis solar tracking system based on combining MLP and MLR when $K=5$ by using data set two

Single-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional MLP	93.72	93.79	96.47	0.350×10^{-2}	0.150×10^{-2}	0.090×10^{-2}
Developed-MLP with MLR	96.48	97.15	97.85	0.050×10^{-2}	0.020×10^{-2}	6.130×10^{-5}

Abbreviations: MLP, multilayer perceptron; MLR, multinomial logistic regression.

TABLE 25 The overall prediction rate (%) and MSE results of the developed dual-axis solar tracking system based on combining MLP and BLR when $K=2$, by using data set two

Dual-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional MLP	94.64	94.94	95.36	0.130×10^{-2}	0.090×10^{-2}	0.070×10^{-2}
Developed-MLP with BLR	96.70	96.75	98.65	0.070×10^{-2}	0.040×10^{-2}	0.017×10^{-2}

Abbreviations: BLR, binomial logistic regression; MLP, multilayer perceptron.

layer in the developed MLP-BLR system obtained lower MSE than the three scenarios of one, two, and three hidden layers in conventional MLP.

On the other hand, data set two is tested and evaluated to find the MSE and the prediction rate of the developed single and dual axis solar tracking systems. Tables 24 and 25 show the results of MSE and prediction rate of the developed single and dual axis solar tracking systems compared to the conventional MLP network.

As shown in Tables 24 and 25, the overall prediction rate that obtained by using the developed single-axis solar tracking system based on MLP and MLR is higher than the prediction rate of conventional MLP based system. Moreover, using one hidden layer in the developed single-axis solar tracking system achieved higher prediction rate compared to that using the three scenarios of one, two, and three hidden layers in conventional MLP based system. One hidden layer of MLP and MLR obtained a prediction rate of 96.48%, 97.15%, and 97.85% with one, two, and three hidden layers respectively. On the other hand, one hidden layer in the developed MLP-MLR system obtained lower MSE than the three scenarios of one, two, and three hidden layers in conventional MLP. The MSE of one, two, and three hidden layers are 0.050×10^{-2} , 0.020×10^{-2} , and 6.130×10^{-5} , respectively. Moreover, increasing the number of hidden layers could increase the performance of the developed single-axis solar tracking system by decreasing the achieved MSE. The optimum MSE obtained when three hidden layers are used.

On the other hand, the overall prediction rate that obtained by using the developed dual-axis solar tracking system based on MLP and BLR is higher than the prediction rate of conventional MLP based system. One hidden layer of MLP and BLR obtained a prediction rate of 96.703%, 96.75%, and 98.65% with one, two, and three hidden layers, respectively. Moreover, the developed dual-axis solar tracking system achieved very low MSE. One hidden layer in the developed MLP-BLR system obtained lower MSE than the three scenarios of one, two, and three hidden layers in conventional MLP. The MSE of one, two, and three hidden layers are

0.070×10^{-2} , 0.040×10^{-2} , and 0.017×10^{-2} , respectively for MLP and BLR network. Moreover, increasing the number of hidden layers increased the performance of the developed dual-axis solar tracking system by decreasing the achieved MSE. The optimum MSE obtained when three hidden layers are used.

All the results revealed that increasing the number of hidden layers could increase the performance of the developed single and dual axis solar tracking systems and decrease the achieved MSE. The optimum MSE obtained when three hidden layers are used. These results revealed that the developed single and dual axis solar tracking systems could minimize the overhead of increasing the number of hidden layers in conventional MLP. Thus, decrease the power consumption and the processing time.

Tables 26 and 27 show a comparison between the number of nodes, the number of connections, and time complexity using one hidden layer in developed solar tracking system versus using three hidden layers in conventional MLP for data set one and data set two, respectively.

As shown in Tables 26 and 27, the developed intelligent solar tracking controllers are better than conventional MLP based controller because only one hidden layer will be adopted to implement the developed model practically instead of using three hidden layers.

In contrast, although the number of inputs in developed system is greater than the conventional system, but less number of neurons are used, therefore, less number of overall connections. The overall connections in conventional MLP using data set one are 240 and 250 connections for single-axis and dual-axis systems respectively, and using data set two are 270 and 280 for single-axis and dual-axis systems, respectively.

While the number of overall connections of the developed single and dual axis tracking systems are 50 and 60 connections respectively for data set one, and 80 and 90, respectively for data set two. Increasing the number of connections could increase the number of multiplication

TABLE 26 A comparison between conventional MLP and MLP-BLR/MLR based systems for data set one

Parameter	MLP-3 hidden layers		MLP-BLR/MLR-1 hidden layer	
	Single-axis	Dual-axis	Single-axis	Dual-axis
Number of Inputs	3	3	4	4
Number of neurons	31	32	11	12
Number of connections (inputs-hidden nodes)	30	30	40	40
Number of connections (hidden nodes-outputs)	10	20	10	20
Number of connections (hidden nodes)	200	200	0	0
Overall connections	240	250	50	60
Cost Function (MSE)	0.250×10^{-2}	0.160×10^{-2}	0.240×10^{-2}	0.100×10^{-2}
Prediction Rate (%)	96.75	96.68	96.78	96.80
Time Complexity for (n) samples	$O(d \times h) + O$ $(h \times h) + O$ $(h \times h) + O$ $(h \times c)$	$O(d \times h) + O$ $(h \times h) + O$ $(h \times h) + O$ $(h \times c)$	$O(d \times h) + O$ $(h \times c) +$ $(v^2 \times c)$	$O(d \times h) + O$ $(h \times c) +$ $(v^2 \times c)$

TABLE 27 A comparison between conventional MLP and MLP-BLR/MLR based systems for data set two

Parameter	MLP-3 hidden layers		MLP-BLR/MLR-1 hidden layer	
	Single-axis	Dual-axis	Single-axis	Dual-axis
Number of Inputs	6	6	7	7
Number of neurons	31	32	11	12
Number of connections (inputs-hidden nodes)	60	60	70	70
Number of connections (hidden nodes-outputs)	10	20	10	20
Number of connections (hidden nodes)	200	200	0	0
Overall connections	270	280	80	90
Cost Function (MSE)	0.090×10^{-2}	0.070×10^{-2}	0.050×10^{-2}	0.070×10^{-2}
Prediction Rate (%)	96.47	95.36	96.48	96.70
Time Complexity for (n) samples	$O(d \times h) + O$ $(h \times h) + O$ $(h \times h) + O$ $(h \times c)$	$O(d \times h) + O$ $(h \times h) + O$ $(h \times h) + O$ $(h \times c)$	$O(d \times h) + O$ $(h \times c) +$ $(v^2 \times c)$	$O(d \times h) + O$ $(h \times c) +$ $(v^2 \times c)$

terms (i.e., $X1 \cdot W1$), which increase the processing time and the memory size of the system, therefore, increase the time complexity that measures the estimation time to run the system. Moreover, the cost/performance can be calculated by comparing the MSE of both cases together (Rady, 2011).³⁹ The MSE of the developed system is less than the conventional MLP-based system. As a result, the developed model is simpler, more accurate, needs less memory size, fast processing time, and less power consumption as well. As shown in Tables 26 and 27, the number of hidden layers that used to implement the solar tracking model plays the major role to increase the time complexity of the solar tracking systems.

The time complexity of using three hidden layers is much greater than using single hidden layer in the developed models although of using LR models in combination with MLP network. The improvement of time complexity of the developed MLP-BLR/MLR systems when data set one is used are 72.50% and 68.30% for single and dual axis solar tracking systems, respectively. While the improvement of the developed systems when data set two is used are 46.67% and 45% for single and dual axis solar tracking systems, respectively.

4.3.2 | Results of developed single and dual axis solar tracking system based on CMLP

The block diagram of the developed single and dual axis solar tracking systems based on combining CMLP network and MLR models is shown in Figure 12. As shown in Figure 12, the developed single or dual axis solar tracking system will use the predicted cluster from the trained LR model as input variable to the CMLP network along with the input variables, in addition to the input variables of month, day, and time or month, day, time, I_{sc} , V_{oc} , and power radiation.

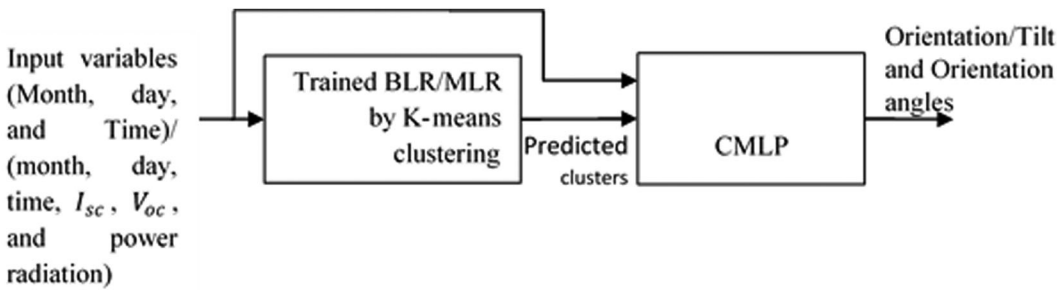


FIGURE 12 Block diagram of the developed single and dual axis solar tracking systems based on BLR/MLR and CMLP. BLR, binomial logistic regression; CMLP, cascade multilayer perceptron; MLR, multinomial logistic regression

The developed single and dual axis solar tracking systems are tested, analyzed, and compared with conventional CMLP network by adopting the methodology that explained earlier. Three different scenarios with one, two, and three hidden layers are tested to find the prediction rate and MSE. Tables 28 and 29 show the MSE and the prediction rate of the developed single and dual axis solar tracking systems when data set one is used.

As shown in Tables 28 and 29, the overall prediction rate that obtained by using one hidden layer of the developed single-axis solar tracking system achieved higher prediction rate as compared to using three hidden layers in conventional CMLP based system. The prediction rate of one, two, and three hidden layers of the developed single-axis solar tracking system achieved 97.60%, 98.52%, and 98.97%, respectively, compared to 96.35%, 96.52%, and 96.83% for one, two,

TABLE 28 The prediction rate (100%) and MSE results of the developed single-axis solar tracking system based on combining CMLP and MLR when $K = 4$ by using data set one

Single-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional CMLP	96.35	96.52	96.83	0.520×10^{-2}	0.250×10^{-2}	0.080×10^{-2}
Developed-CMLP with MLR	97.60	98.52	98.97	0.160×10^{-2}	0.070×10^{-2}	0.020×10^{-2}

Abbreviations: CMLP, cascade multilayer perceptron; MLR, multinomial logistic regression.

TABLE 29 The prediction rate (100%) and mse results of the developed dual-axis solar tracking system based on combining CMLP and MLR when $k = 10$ by using data set one

Dual-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional CMLP	93.87	97.37	97.98	0.180×10^{-2}	0.160×10^{-2}	0.070×10^{-2}
Developed-CMLP with MLR	97.06	98.71	99.19	0.050×10^{-2}	6.030×10^{-5}	6.250×10^{-5}

Abbreviations: CMLP, cascade multilayer perceptron; MLR, multinomial logistic regression.

and three hidden layers, respectively in conventional CMLP. On the other hand, the developed single-axis solar tracking system achieved low MSE. Using two hidden layers of the developed single-axis tracking system achieved better MSE compared to the three scenarios of conventional CMLP. One, two, and three hidden layers of developed single-axis solar tracking system achieved MSE of 0.16×10^{-2} , 0.070×10^{-2} , 0.020×10^{-2} , respectively compared to 0.025×10^{-2} , 0.025×10^{-2} , and 0.080×10^{-2} , respectively.

Moreover, the prediction rate of dual-axis solar tracking system based on CMLP and MLR is higher than the prediction rate of conventional CMLP based system. Using one hidden layer in the developed dual-axis tracking system achieved very close prediction rate compared to using three hidden layers in CMLP based system. The prediction rate of one, two, and three hidden layers achieved 97.06%, 98.71%, and 99.19%, respectively. On the other hand, the developed dual-axis solar tracking system achieved extremely low MSE. One hidden layer in the developed CMLP-MLR system obtained 0.050×10^{-2} MSE which is lower MSE than the three scenarios of one, two, and three hidden layers in CMLP. Three hidden layers obtained the minimum MSE of 6.250×10^{-5} . The MSE values are superior and can guarantee the capability of the developed system in predicting both tilt and orientation angles efficiently.

On the other hand, data set two is tested and evaluated to find the MSE and the prediction rate of the developed single and dual axis solar tracking systems. Three different scenarios with one, two, and three hidden layers are tested to find the prediction rate and MSE. Tables 30 and 31 show the results of MSE and prediction rate of the developed single and dual axis solar tracking systems compared to the conventional CMLP network.

As shown in Tables 30 and 31, the MSE that obtained by using one hidden layer of the developed single-axis solar tracking system based on CMLP and MLR is equal to the MSE of three hidden layers of conventional CMLP-based system. One, two, and three hidden layers of

TABLE 30 The overall prediction rate (100%) and MSE results of the developed single-axis solar tracking system based on combining CMLP and MLR when $K = 10$ by using data set two

Single-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional CMLP	94.87	96.40	97.80	0.420×10^{-2}	0.090×10^{-2}	0.030×10^{-2}
Developed-CMLP with MLR	95.80	98.10	98.30	0.030×10^{-2}	0.020×10^{-2}	5.330×10^{-5}

Abbreviations: CMLP, cascade multilayer perceptron; MLR, multinomial logistic regression.

TABLE 31 The overall prediction rate (100%) and MSE results of the developed dual-axis solar tracking system based on combining CMLP and MLR when $K = 4$ by using data set two

Dual-axis based	Prediction rate (%)			MSE		
	Number of hidden layers			Number of hidden layers		
	One	Two	Three	One	Two	Three
Conventional CMLP	94.27	95.90	96.03	0.110×10^{-2}	0.240×10^{-2}	0.060×10^{-2}
Developed-CMLP with MLR	95.60	97.03	97.60	0.060×10^{-2}	0.040×10^{-2}	0.020×10^{-2}

Abbreviations: CMLP, cascade multilayer perceptron; MLR, multinomial logistic regression.

the developed CMLP obtained MSE of 0.030×10^{-2} , 0.020×10^{-2} , and 5.330×10^{-5} , respectively. On the other hand, the overall prediction rate of the developed single-axis solar tracking system is higher than the prediction rate of conventional CMLP. One hidden layer in the developed CMLP-MLR system obtained 95.80%, while two and three hidden layers obtained 98.10% and 98.30%, respectively.

Moreover, increasing the number of hidden layers increased the performance of the developed single-axis solar tracking system by decreasing the achieved MSE. The optimum MSE obtained when three hidden layers are used. This is because increasing the number of layers would increase the degree of nonlinearity and would add a level of abstraction.^{81–83}

On the other hand, the overall prediction rate that obtained by using the developed dual-axis solar tracking system based on CMLP combined with the MLR is higher than the prediction rate of conventional CMLP-based system. One hidden layer of the developed CMLP-MLR obtained a prediction rate of 95.60%, 97.03%, and 97.60% with one, two, and three hidden layers, respectively. Moreover, the developed dual-axis solar tracking system achieved very low MSE. One hidden layer in the developed CMLP-MLR system obtained a MSE equal to the MSE of three hidden layers of conventional CMLP. The MSE of one, two, and three hidden layers are 0.060×10^{-2} , 0.040×10^{-2} , and 0.020×10^{-2} , respectively. Moreover, increasing the number of hidden layers increased the performance of the developed dual-axis solar tracking system by decreasing the achieved MSE. The optimum MSE obtained when three hidden layers are used.

Moreover, all the results revealed that increasing the number of hidden layers increased the performance of the developed single and dual axis solar tracking systems, and decreased the achieved MSE. The optimum MSE obtained when three hidden layers are used. These results revealed that the developed single and dual axis solar tracking systems could minimize the overhead of increasing the number of hidden layers in conventional CMLP as well as of conventional MLP. Tables 32 and 33 show a comparison between using one hidden layer in developed solar tracking system versus using three hidden layers in conventional CMLP for data set one, and data set two, respectively.

Tables 32 and 33 show a comparison between using one hidden layer in developed solar tracking system versus using three hidden layers in conventional CMLP for data set one, and data set two, respectively.

As shown in Tables 32 and 33, the developed intelligent solar tracking controllers are better than conventional CMLP based controller because only one hidden layer would be adopted to implement the developed model practically instead of using three hidden layers for conventional CMLP. The overall connections in conventional CMLP using data set one are 423 and 456 connections for single-axis and dual-axis systems, respectively, and for using data set two are 516 and 552 for single-axis and dual-axis systems, respectively. While the number of overall connections of the developed single and dual axis tracking systems are 54 and 68 connections, respectively for data set one, and 87 and 104, respectively for data set two.

Similar to the proposed MLP based systems, increasing the number of connections could increase the number of multiplication terms (i.e., $X1*W1$), which increase the processing time and the memory size of the system. The MSE of the developed system is less than the conventional CMLP based system. As a result, the developed model is simpler, more accurate, needs less memory size, fast processing time, and less power consumption as well.

Finally, similar to the proposed single-axis solar tracking systems, the developed dual-axis solar tracking systems based on CMLP and MLP revealed that both developed systems are sufficient to optimize the conventional methods, decreasing the power consumption, decreasing the complexity of the solar tracking system and fasten the processes of positioning the

TABLE 32 A comparison between conventional CMLP and CMLP-BLR/MLR based systems for data set one

Model	CMLP-3 hidden layers		CMLP- MLR-1 hidden layer	
	Single-axis	Dual-axis	Single-axis	Dual-axis
Number of inputs	3	3	4	4
Number of neurons	31	32	11	12
Number of connections (inputs-hidden nodes)	90	90	40	40
Number of connections (hidden nodes-outputs)	33	66	14	28
Number of connections (hidden nodes)	300	300	0	0
Overall connections	423	456	54	68
Cost function (MSE)	0.080×10^{-2}	0.070×10^{-2}	0.160×10^{-2}	0.050×10^{-2}
Prediction rate	96.83%	97.98%	97.60%	97.06%
Time complexity for (<i>n</i>) samples	$O(d \times h) + O$ $(d \times h \times h) +$ $O(d \times h \times$ $h \times h) + O$ $(d \times h \times h \times$ $h \times c)$	$O(d \times h) + O$ $(d \times h \times h) + O$ $(d \times h \times h \times h) +$ $O(d \times h \times h \times$ $h \times c)$	$O(d \times h) + O$ $(d \times h \times c) +$ $(v^2 \times c)$	$O(d \times h) + O$ $(d \times h \times c) +$ $(v^2 \times c)$

solar tracking systems toward the sun. The improvement of time complexity of the developed CMLP-BLR/MLR systems when data set one is used are 98.16% and 99.84% for single and dual axis solar tracking systems, respectively. While the improvement of the developed systems when data set two is used are 98.07% and 98.34% for single and dual axis solar tracking systems.

5 | DISCUSSION

Comparing all the achieved results of the developed single and dual axis solar tracking systems with literature review results, it is found that the achieved results are in line with the results that obtained by Rabee et al.⁹ The researchers have proved that neural network principles are able to cope with nonlinear data. In addition, it is proved that neural network could be used as a good controller for solar tracking systems.^{3,4,9,10} Moreover, the results of the proposed systems revealed that neural network principles can deal with time series data and can be used to avoid the disadvantages of conventional methods.

It is shown by literature review that neural networks need to maintain the generalization of the forecaster and avoid the overfitting.⁹ These two problems have been maintained by either modifying the error optimization function,¹⁰ or by increasing the number of hidden layers and hidden neurons.^{9,10} In addition, it is proven that using these solutions could increase the processing time, and thus increase the consumed energy.⁹

However, 83.3% of the research are based on using random-generated data,^{3,9} or data that measured in very short time.^{4,10}

TABLE 33 A comparison between conventional CMLP and CMLP-BLR/MLR based systems for data set two

Model	MLP-3 hidden layers		MLP-BLR/MLR-1 hidden layer	
	Single-axis	Dual-axis	Single-axis	Dual-axis
Number of inputs	6	6	7	7
Number of neurons	31	32	11	12
Number of connections (inputs-hidden nodes)	180	180	70	70
Number of connections (hidden nodes-outputs)	36	72	17	34
Number of connections (hidden nodes)	300	300	0	0
Overall connections	516	552	87	104
Cost function (MSE)	0.0009	0.0005	0.0009	0.0007
Prediction rate	97.80%	96.03%	95.08%	95.60%
Time complexity for (<i>n</i>) samples	$O(d \times h) + O(d \times h \times h) + O(d \times h \times h \times h) + O$ $(d \times h \times h \times h \times c)$	$O(d \times h) + O(d \times h \times h) + O(d \times h \times h \times h) + O(d \times h \times h \times h \times c)$	$O(d \times h) + O(d \times h \times c) + (n \times v^2 \times c)$	$O(d \times h) + O(d \times h \times c) + (n \times v^2 \times c)$

Using inaccurate data is the main reason to slow the convergence to reach the desired goal. In addition, using inaccurate data to train the neural network will add an additional risk to find faulty and inaccurate results.

In contrast, this study used real experimental data to train, validate, and test the neural network models. Therefore, the performance results of the proposed systems would be considered as trusted results. Moreover, the results revealed that both MLP and CMLP can be used sufficiently to control solar tracking systems. However, the results revealed that using three hidden layers performed better than using one and two hidden layers. In addition, it is found that using 10 neurons in the hidden layers performed better than using 1–9 or 11–15 neurons. Therefore, the proposed systems could work properly by increasing the number of hidden layers and hidden nodes up to specific limit which is the same finding by literature review.^{2,9,10}

However, increasing the complexity of the proposed controllers (i.e., MLP and CMLP) would not achieve the primary goal of using artificial intelligence techniques to control solar tracking systems. Thus, it is necessary to propose a simple system that can efficiently control solar tracking systems. On the other hand, and depends on the characteristics of neural network principles, it is found that MLP principles suffer from slow convergence to reach the desired goal, high cost, and high time complexity problems. In addition, it has been proven by literature review that using a reference value along with input variables to neural network can help to approximate the desired goal faster, decrease the complexity of the proposed controller, increase the performance, and avoid the overfitting.³

Moreover, this study mainly aims to design a new solar tracking system controller by integrating MLP principles with a trained LR model. This LR model is trained by k-means

clustering algorithm with different number of clusters for each case. However, it has been proven by this study that the output of the trained LR can be used as a reference value along with the input variables of MLP principles. However, the output of the trained LR can help to identify the relationships between data set variables because LR is more robust to find the relationship between samples of data compared to other statistical models.^{15,16} These relationships can guide the MLP principles to converge fast, thus, improve the performance of MLP. In addition, LR models are trained to preclassify the input variables into clusters. The trained LR models can easily incorporate new training data into the model.^{84,85} They are able to accurately refine the data into an appropriate cluster. This is because it can easily adjust the classification threshold to find confidence classes.^{18,86}

The results of this study have proven that combining the trained LR models with MLP principles could simplify the proposed controller by minimizing the number of hidden layers, the number of hidden nodes, the number of overall connections, and the time complexity. Thus, the output of the trained LR models can be considered as optimum reference value to MLP. In addition, integrating LR models with MLP can be used globally to support, improve, and enhance the literature review.

The proposed models achieved the objective of developing new principle to drive efficient single and dual axis solar tracking systems. Integrating MLP or CMLP with MLR or BLR models that trained by k-means clustering algorithm is employed to achieve the objective of this study. The results revealed that using one hidden layer of the developed single and dual axis solar tracking systems achieved higher prediction rates and lower MSE as compared to using three hidden layers in the conventional MLP and CMLP based systems. In addition, the new MLP developed models achieved less number of overall connections for both single and dual axis solar tracking systems (i.e., 79.17% and 76% less connections with data set one, and 70.37% and 67.86% for data set two, respectively). While the new CMLP developed models achieved less number of overall connections for both single and dual axis solar tracking systems (i.e., 87.23% and 85.09% less connections with data set one, and 83.14% and 81.16% for data set two, respectively).

Moreover, using one hidden layer in the developed MLP and CMLP models achieved less number of neurons for both single and dual axis solar tracking systems as compared to using three hidden layers in conventional models. In addition, the developed MLP models achieved less time complexity for both single and dual axis solar tracking systems as compared to conventional models. While the developed CMLP models achieved less time complexity for both single and dual axis solar tracking systems as compared to conventional models. Putting everything together, all the objectives and the aim of the research were covered and proved.

6 | COMPARISON OF THE PROPOSED METHOD WITH OTHER RECENT PUBLISHED PREDICTORS

As presented in literature review, several articles were published recently to develop different intelligent models that can predict the optimum tilt, orientation, or tilt and orientation angles of photovoltaic modules globally. These developed models can be considered as the most recent proposed intelligent models that proved their capability to adjunct solar panels. However, the efficiency and the robustness of these models vary from one model to another. This variation depends on different variables (i.e., performance, consumed energy, number of variables used, cost, and time complexity).

This section focuses on compare the most recent developed predictors that proposed to predict tilt and orientation angles with the developed models in this article MLP-BLR/MLR and CMLP-BLR/MLR. Four recent articles are considered in the comparison to evaluate and assess the superiority and the contribution level of the proposed models in this article.

As explained earlier, AL-Rousan et al.² and Şahin,¹³ have used multilayer feed forward artificial neural network MLP, where, AL-Rousan et al.² have compared their results with both linear regression and CMLP. They have proven that linear regression is not capable to predict tilt and orientation angles. The predicted results using linear regression were less than 70% of the data samples used. This result justified why linear models are not suitable to be used in predicting tilt and orientation angles, where, nonlinear models are optimum to be used in such application. In line with their results, Şahin,¹³ has proven the capability of MLP as well. However, 126 and 84 of input and output variables were used to develop the proposed models respectively. These variables were analyzed statistically using PAWS statistics 18 program. The huge amount of input and output variables used, besides to the analysis preprocess need long processing time and add more complexity to the proposed model. In addition, the process of using MLP with all of the selected variables should be repeated every month after adjusting the solar panels manually which will add extra complexity.

Abdallah et al.¹² have proposed a multilevel world comprehensive feed forward neural network model. This new proposed model consists of two stages of neural networks with one hidden layer in each network. The proposed model could obtain very high accuracy and efficient results worldwide. However, the number of hidden layers and hidden nodes for the hidden layers vary from using one hidden layer with 100 nodes in the two stages of neural network to using two hidden layers with 350 nodes for each hidden layer in the two stages. They found that using one hidden layer with 350 nodes in the hidden layer for each stage is the optimum scenario. On the other hand and based on the characteristics of neural networks, using large number of neurons would increase the on-linearity level, thus it needs long processing time and consume more energy.

Kim et al.¹⁴ have used five of machine learning algorithms namely, least absolute shrinkage and selection operator, random forest, support vector machine, and gradient boosting, besides to linear regression to predict tilt angle of photovoltaic panel. They have used different variables to develop their models. Several variables used are unnecessary to build their models. They examined 32 variables and selected 14 variables as optimum variables to be used. Besides, they used huge amount of data samples 173,568. They found that their models could increase the amount of solar energy generated by PV modules compared to fixed panels, where, Gradient Boosting performed better compared to other learning algorithms.

To assess and examine the idea of integrating the trained LR with MLP and CMLP, all the mentioned recent articles should be compared with the developed model (i.e., multilayer feed forward artificial neural network, multilevel world comprehensive feed forward neural network, and Gradient Boosting models).

Several aspects should be considered namely, number of axis, number of input variables, targeted results, and time complexity of both worst and best cases. Table 34 presents the comparison between the optimum cases of the developed solar tracking controllers.

As shown in Table 34, it is clear that the number of input variables vary from one model to another, Şahin,¹³ has used a huge number of input and output variables, Kim et al.¹⁴ have used a relatively large number of variables compared to Abdallah et al.¹² AL-Rousan et al.² and the proposed models in this article, where, two to four variables were used. In addition, they have focused on studying dual-axis solar tracking systems. This would increase the advantages

TABLE 34 Comparison between most recent developed models to predict tilt/orientation angle.

Parameter	Abdallah et al. ¹²	Şahin ¹³	AL-Rousan et al.	Kim et al.	Developed models
Model used	Multilevel world comprehensive feed forward neural network	Multilayer feed forward neural network	Multilayer feed forward neural network	Gradient boosting	MLP-LR MLP-LR CMLP-LR
Number of inputs	Stage 1: 2 Stage 2: 4	126 input variables divided into 7 groups	Single-axis: 3 Dual-axis: 3	14	Single-axis: 4 Dual-axis: 4
Targeted output	Stage 1: Tilt and Orientation angles Stage 2: three outputs of annual radiation	84 of Tilt angle	Single-axis: Orientation angle Dual-axis: Tilt and Orientation angles	Tilt angle	Single-axis: Orientation angle Dual-axis: Tilt and Orientation angles
Number of hidden nodes in hidden layer (if NNs model is used)	Stage 1: 350 Stage 2: 350	-	Single-axis: 30 Dual-axis: 30	-	Single-axis: 10 Dual-axis: 10
Number of connections (inputs-hidden nodes)	Stage 1: 700 Stage 2: 1400	-	Single-axis: 30 Dual-axis: 30	-	Single-axis: 40 Dual-axis: 40
Number of connections (hidden nodes-outputs)	Stage 1: 700 Stage 2: 1050	-	Single-axis: 10 Dual-axis: 20	-	Single-axis: 10 Dual-axis: 20

(Continues)

TABLE 3.4 (Continued)

Parameter	Abdallah et al. ¹²	Şahin ¹³	AL-Rousan et al.	Kim et al.	Developed models
Number of connections (hidden nodes)	Stage 1: 0 Stage 2: 0	-	Single-axis: 200 Dual-axis: 200	-	Single-axis: 0 Dual-axis: 0
Overall connections	Stage 1: 1400 Stage 2: 2450	-	Single-axis: 240 Dual-axis: 250	-	Single-axis: 50 Dual-axis: 60
Time complexity for (n) samples	$O(d \times h) + O(h \times c) + O(h \times c) + O(d \times h) + O(h \times c)$	$O(d \times h) + O(h \times c)$	Single-axis: $O(d \times h) + O$ $(d \times h) + O$ $(h \times h) + O$ $(h \times h) + O$ $(h \times c)$	Single-axis: O $(d \times h) + O(d \times h \times h) + O(d \times h \times h \times h) + O(d \times h \times h \times h \times c)$	Single-axis: $O(d \times h) + O(h \times c) + (v^2 \times c)$ Dual-axis: $O(d \times h) + O(h \times c) + (v^2 \times c)$
			Dual-axis: O $(d \times h) + O$ $(h \times h) + O$ $(h \times h) + O$ $(h \times c)$	Dual-axis: O $(d \times h) + O(d \times h \times h) + O(d \times h \times h \times h) + O(d \times h \times h \times h \times c)$	Dual-axis: $O(d \times h) + O(h \times c) + (v^2 \times c)$ $O(d \times h \times c) + (v^2 \times c) \times c$

of the developed models due to their capability to predict the exact position of the sun by adjusting the used solar panels vertically and horizontally at the same time.

On the other hand, the selected variables and the number of selected variables have a strong impact on the performance and the time complexity of the proposed models. Increasing the number of variables and using inappropriate combination of input variables would decrease the performance and increase the time complexity. In addition and as mentioned earlier, increasing the number of hidden nodes in the hidden layers, and the number of hidden layers in the network would increase the complexity, the cost, processing time, and consumed energy.

However, Kim et al.¹⁴ have used Gradient Boosting Machine (GBM) model that depends on the number of samples, number of features, and the number of trees used. They didn't mention any detailed information about the developed GBM model used, thus, the default number of trees normally adopted will be assumed. One hundred is the default number of trees normally used. Considering this number of trees, the time complexity can be calculated using the formula:

$$\text{Complexity}_{GBM} = O(npn_{trees})$$

where n is the number of samples, p denotes the number of features, and n_{trees} is the number of trees (iterations) used in the model.

The time complexity for the developed GBM is relatively high due to the huge number of samples used which can affect on its performance. A total of 173,568 samples of solar power generation data were acquired from 22 PV modules were used, besides to 14 features were selected and used to develop the models. Taking the selected number of data samples and features together, and although the GBM showed low root mean square error, the estimation time to run the system is relatively high. This can be concluded from the mean feature of GBM model which continue improving to minimize all errors, and this would overemphasize outliers and cause overfitting. Furthermore, applying the developed GBM by other researchers is not applicable due to the unclarity to describe the developed models in the published article, and due to unavailability of the same amount of data samples used. Therefore, GBM model is computationally expensive and requires many trees to be applied which can consume space and time compared to the used neural network models.

On the other hand, all the rest models depend on using multilayer feed forward neural network model which proved its capability in predicting both tilt and orientation angles of solar photovoltaic modules. As mentioned earlier, Şahin,¹³ has used 126 and 84 input and output variables, respectively. Besides, the structure and architecture of the developed neural network are not clear. No detailed information mentioned about the number of hidden layers and the number of hidden nodes adopted in the hidden layers, while it is clear that the number of features and the number of samples used are huge, thus, the time complexity would be increased and this study will be excluded from the comparison.

However, Abdallah et al.¹² have proposed a new approach of neural network models, while Al-Rousan et al.² have used conventional MLP and CMLP neural networks to develop their models. It is clear that the number of hidden neurons and hidden layers vary in the compared models. Abdallah et al.¹² have used two stages of neural network, each stage consists of one hidden layer with 350 neurons. While Al-Rousan et al.² have used three hidden layers with 10 neurons for each. Comparing the number of connections of input-hidden nodes found that Abdallah et al.¹² have used 23 and 46 more times connections than MLP model in both Stage 1 and Stage 2, respectively, and they have used 7 and 14 more times connections than CMLP in both Stage 1 and Stage 2, respectively. While the proposed models of MLP-LR and CMLP-LR

have used a single hidden layer with 10 hidden nodes. Thus, the number of connections of input-hidden nodes are 40 in both MLP-LR and CMLP-LR. Besides, the number of connections of hidden-outputs found that Abdallah et al.¹² has the highest number of connections compared to both the developed MLP-LR and CMLP-LR. Abdallah et al.¹² have used 70 and 40 more times connections than both MLP-LR and CMLP-LR models in both Stage 1 and Stage 2, respectively, while they used 70 and 21 more times connections than both MLP and CMLP developed models in both Stage 1 and Stage 2, respectively. On the other hand, the number of connections between the hidden nodes in the proposed MLP-LR and CMLP-LR is zero for both cases. While Al-Rousan et al.² have used 200 and 300 for both MLP and CMLP, respectively. Comparing the overall connections indicates that the developed MLP-LR obtained the minimum number of overall connections for both single and dual axis cases. Besides, the developed CMLP-LR obtained relatively small number of overall connections for both cases of single and dual axis. While the overall connections of both Al-Rousan et al.² models and Abdallah et al.¹² cases are very large compared to the proposed MLP-LR and CMLP-LR models. Thus, and putting everything together, it is clear that the estimation time to run the proposed MLP-LR and CMLP-LR and the time complexity is relatively small compared to other models, and it can be concluded that using a single hidden layer of the proposed MLP-LR and CMLP-LR models can achieve higher performance, lower estimation time, and less time complexity compared to other recent published models. This would present the originality and the novelty of this article. Besides, the comprehensive comparison with recent published algorithms would clarify the degree of the superiority of the proposed MLP-LR and CMLP-LR models and to prove that preclassify the input variables into clusters and premapping of relationship between samples of data as input to neural network along with the original input data could probably a strong guide to help neural network to reach the desired goal and predict the output variables faster and more accurate compared to other recent published algorithms.

7 | CONCLUSION

The main aim of this study is to develop new principle to drive efficient single and dual axis solar tracking systems, and to solve the problem of using high level of nonlinearity in MLP.

To achieve this objective, integrating MLP and CMLP with MLR or BLR models that trained by k-means clustering algorithm is employed. These developed models are used to propose new efficient and low complexity single and dual axis solar tracking systems controllers. These controllers could increase the performance of the current solar trackers, increase the prediction rate, decrease the MSE, and decrease the time complexity.

Real data sets for single and dual axis solar tracking systems were used. Several variables are used to predict orientation angle in single-axis solar tracking system or both tilt and orientation angles in dual-axis solar tracking systems.

The results revealed that using one hidden layer of the developed single and dual axis solar tracking systems achieved higher prediction rates and lower MSE as compared to using three hidden layers in the conventional MLP and CMLP based systems. In addition, the new MLP developed models achieved less number of overall connections for both single and dual axis solar tracking systems. While the new CMLP developed models achieved less number of overall connections for both single and dual axis solar tracking systems.

Moreover, using one hidden layer in the developed MLP and CMLP models achieved less number of neurons for both single and dual axis solar tracking systems as compared to using

three hidden layers in conventional models. In addition, the developed MLP models achieved less time complexity for both single and dual axis solar tracking systems as compared to conventional models. While the developed CMLP models achieved less time complexity for both single and dual axis solar tracking systems as compared to conventional models. Putting everything together, all the objectives and the aim of the research were covered and proved.

In addition, this study contributed to the fields of solar energy and intelligent systems, where, a new structure of neural networks with a single hidden layer that performs better than three hidden layers in conventional MLP is developed, and a new generation of intelligent solar tracking systems where the level of nonlinearity is minimized and the time complexity and estimation time are decreased is proposed. Using a reference value to help neural network to find the relationship between the data samples and guide the neural network to faster predict the desired goal is a novel idea. Besides, a comprehensive comparison between the proposed models in this article and the most recent published articles in the field is presented. This article would be a good reference to other researchers in the field.

The future trend of the proposed solar tracking systems is to implement the proposed systems practically using real solar photovoltaic module to test the proposed systems and optimize its performance.

Moreover, improving and optimizing the current methods and principles of solar tracking systems is very important future trend to increase the performance and the efficiency of solar tracking systems. Several changes on the current tracking methods can be applied to maximize the solar tracker gain power. Using hybrid techniques, new optimization method for current principles, and using new artificial intelligent techniques are possible methods to increase the efficiency of solar tracking systems.

On the other hand, it is important to test, evaluate, and validate the proposed intelligent method using different applications. It is believed that the proposed models by integrating (MLP/CMLP) with LR models could be used efficiently in different real applications (i.e., agricultural, environmental, medical, etc.).

CONFLICT OF INTERESTS

The authors declare that there are no conflict of interests.

ORCID

Nadia AL-Rousan  <http://orcid.org/0000-0001-8451-898X>

Nor Ashidi Mat Isa  <https://orcid.org/0000-0002-2675-4914>

Mohammad Khairunaz Mat Desa  <https://orcid.org/0000-0002-3903-1133>

Hazem AL-Najjar  <http://orcid.org/0000-0002-6143-2734>

REFERENCES

1. Benganem M, Mellit A, Alamri S. ANN-based modelling and estimation of daily global solar radiation data: a case study. *Energy Convers Manage*. 2009;50(1):1644-1655.
2. AL-Rousan N, Mat Isa NA, Mat Desa MK. Correlation analysis and MLP/CMLP for optimum variables to predict orientation and tilt angles in intelligent solar tracking systems. *Int J Energy Res*. 2021;45(1):453-477.
3. Zaki SA, Mostafa M. Maximum power point tracking for PV system using advanced neural networks technique. *Int J Emerging Technol Adv Eng*. 2012;2(1):58-63.
4. Panait M, Tudorache T. A simple neural network solar tracker for optimizing conversion efficiency in off-grid solar generators. *Int Conf Renew Energies Power Quality*. 2008;1(6):256-260.
5. AL-Rousan NA, Isa NAM, Desa MKM. Efficient single and dual axis solar tracking system controllers based on adaptive neural fuzzy inference system. *J King Saud Univ-Eng Sci*. 2020;32(7):459-469.

6. Cheikh MA, Haddadi M, Zerguerras A. Design of a neural network control scheme for the maximum power point tracking (MPPT). *Revue des Énergies Renouvelables*. 2007;10(1):109-118.
7. Essefi RM, Souissi M, Abdallah HH. Maximum power point tracking control using neural networks for stand-alone photovoltaic systems. *Int J Modern Nonlin Theory Appl*. 2014;3(1):53-65.
8. Mohammed JK, Tahir F. Sun tracking system based on neural network. *J Eng Devel*. 2013;17(1):123-133.
9. Bou-Rabee M, Sulaiman SA, Saleh MS, Marafi S. Using artificial neural networks to estimate solar radiation in Kuwait. *Renew Sustain Energy Rev*. 2017;72(1):434-438.
10. Kayri I, Gencoglu MT. Prediction of power production from a single axis photovoltaic system by Artificial Neural Networks. In: *14th International Conference on Engineering of Modern Electric Systems (EMES)*; 2017:212-215.
11. Sun Y, Li S, Lin B, Fu X, Ramezani M, Jaithwa I. Artificial neural network for control and grid integration of residential solar photovoltaic systems. *IEEE Trans Sustain Energy*. 2017;8(4):1484-1495.
12. Abdallah R, Natsheh E, Juaidi A, Samara S, Manzano-Agugliaro F. A multi-level world comprehensive neural network model for maximum annual solar irradiation on a flat surface. *Energies*. 2020;13(23):6422-6436.
13. Şahin M. Determining optimum tilt angles of photovoltaic panels by using artificial neural networks in Turkey. *Tehnički Vjesnik*. 2019;26(3):596-602.
14. Kim GY, Han DS, Lee Z. Solar panel tilt angle optimization using machine learning model: a case study of Daegu City, South Korea. *Energies*. 2020;13(3):529-539.
15. Pohar M, Blas M, Turk S. Comparison of logistic regression and linear discriminant analysis: a simulation study. *Metodoloski zvezki*. 2004;1(1):143-161.
16. Feng J, Xu H, Mannor S, Yan S. Robust logistic regression and classification. *Adv Neural Inf Process Syst*. 2014;27(1):253-261.
17. Ahmed IAI, Cheng W. The performance of robust methods in logistic regression model. *Open J Stat*. 2020;10(1):127-139.
18. Serra JG, Ruiz P, Molina R, Katsaggelos AK. Bayesian logistic regression with sparse general representation prior for multispectral image classification. In: *IEEE International Conference on Image Processing (ICIP)*; 2016:1893-1897.
19. Moscatelli M, Parlapiano F, Narizzano S, Viggiano G. Corporate default forecasting with machine learning. *Expert Syst Appl*. 2020;116(1):113567-113577.
20. Sabouri S, Hajrasouliha A, Song Y, Greene WH. Logistic regression. *Basic Quantitative Research Methods for Urban Planners*. Routledge; 2020:270-304. (Chapter 13).
21. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Trans Pattern Anal Mach Intell*. 2002;24(1):881-892.
22. Jafar OM, Sivakumar R. A comparative study of hard and fuzzy data clustering algorithms with cluster validity indices. *Proc Int Conf Emerging Res Comput Inform Commun Appl*. 2013;2(10):775-782.
23. Chen M, Ludwig SA. Color Image Segmentation Using Fuzzy C-Regression Model. *Advances in Fuzzy Systems*. 2017;2017(1):1-16.
24. Nadia AR, Hazem AN, Osama A. Assessment of predicting hourly global solar radiation in Jordan based on Rules, Trees, Meta, Lazy and Function prediction methods. *Sustainable Energy Technologies and Assessments*. 2021;44:100923. <http://doi.org/10.1016/j.seta.2020.100923>
25. Ding H, Liu Y, Huang L, Li J. K-means clustering with distributed dimensions. In: *International Conference on Machine Learning*; 2016:1339-1348.
26. Khrissi L, El Akkad N, Satori H, Satori K. Clustering method and sine cosine algorithm for image segmentation. *Evol Intell*. 2021:1-14.
27. Andryani DS, Bustamam A, Lestari D. The implementation of hybrid clustering using fuzzy c-means and divisive algorithm for analyzing DNA human Papillomavirus cause of cervical cancer. In: *AIP Conference Proceedings*; 2017:020003.
28. Kazemi H, Yazdjerdi K, Asadi A, Mozafari MR. Application of genetic and K-means algorithms in clustering Babakoochi Anticline joints north of Shiraz, Iran. *Central Eur Geol*. 2020;63(1):38-48.

29. Xie WB, Lee YL, Wang C, Chen DB, Zhou T. Hierarchical clustering supported by reciprocal nearest neighbors. *Inf Sci*. 2020;527(1):279-292.
30. Chehreghani MH. Hierarchical correlation clustering and tree preserving embedding. *Adv Electr Comput Eng*. 2017;17(4):77-88.
31. Deng S. Clustering with fuzzy C-means and common challenges. *J Phys: Conf Ser*. 2020;1453(1):1453-1460.
32. Jiang Z, Liu X. Novel coupled DP system for fuzzy C-means clustering and image segmentation. *Appl Intell*. 2020;50(12):4378-4393.
33. Panda S, Sahu S, Jena P, Chattopadhyay S. Comparing fuzzy-C means and K-means clustering techniques: a comprehensive study. *Adv Comput Sci Eng Appl*. 2012;166(1):451-460.
34. Cebeci Z, Yildiz F. Comparison of K-means and fuzzy C-means algorithms on different cluster structures. *J Agric Inform*. 2015;6(1):13-23.
35. Tanaseichuk O, Khodabakshi A, Petrov D, et al. An efficient hierarchical clustering algorithm for large datasets. *Austin J Proteomics Bioinform Genomics*. 2015;2(1):1008-1014.
36. Salvador S, Chan P. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: *16th IEEE International Conference on Tools with Artificial Intelligence*; 2004: 576-584.
37. Jain K, Murty MN, Flynn PJ. Data clustering: a review. *ACM Comput Surveys*. 1999;31(1):264-323.
38. Sajana T, Rani CS, Narayana K. A survey on clustering techniques for big data mining. *Indian J Sci Technol*. 2016;9(3):1-12.
39. Rady H. Ryni's entropy and mean square error for improving the convergence of multilayer back propagation neural networks: a comparative study. *Int J Electr Comput Sci*. 2011;11(5):68-79.
40. Cios KJ, Swiniarski RW, Pedrycz W, Kurgan LA. Unsupervised learning: clustering. *Data Mining*. 2007:257-288.
41. Babyak MA. What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models. *Psychosom Med*. 2004;66(1):411-421.
42. Yarnold PR. Univariate and multivariate analysis of categorical attributes with many response categories. *Optimal Data Analysis*. 2013;2(1):177-190.
43. Breiman L. Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Stat Sci*. 2001;16(1):199-231.
44. Faraway JJ. Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models. *Static Comput*. 2016;124(1):1-395.
45. Yau C. *R Tutorial with Bayesian Statistics Using OpenBUGS*. Amazon Kindle; 2015.
46. Böhning D. Multinomial logistic regression algorithm. *Annal Instit Stat Math*. 1992;44(1):197-200.
47. Hosmer DW Jr., Lemeshow S, Sturdivant RX. Applied logistic regression. *Wiley Series Prob Stat*. 2013; 398(1):60-511.
48. Landwehr N, Hall M, Frank E. Logistic model trees. *Mach Learn*. 2005;59(1):161-205.
49. Dorofki M, Elshafie AH, Jaafar O, Karim OA, Mastura S. Comparison of artificial neural network transfer functions abilities to simulate extreme runoff data. *Int Proc Chem Biol Environ Eng*. 2012;33(1):39-44.
50. Karlik B, Olgac AV. Performance analysis of various activation functions in generalized MLP architectures of neural networks. *Int J Artif Intell Expert Syst*. 2011;1(1):111-122.
51. Nadia AR, Isa NAM, Desa MKM. Advances in solar photovoltaic tracking systems: a review. *Renew Sustain Energy Rev*. 2018;82(1):2548-2569.
52. Goyal GK, Goyal S. Cascade artificial neural network models for predicting shelf life of processed cheese. *J Adv Inform Technol*. 2013;4(1):80-83.
53. Roy S. Factors influencing the choice of a learning rate for a backpropagation neural network. In: *IEEE World Congress on Computational Intelligence Neural Networks*; 1994:503-507.
54. Wang J, Yang J, Wu W. Convergence of cyclic and almost-cyclic learning with momentum for feedforward neural networks. *IEEE Trans Neural Netw*. 2011;22(1):1297-1306.
55. Mallick PK. Research advances in the integration of big data and smart computing: IGI Global. 2015.
56. Qian N. On the momentum term in gradient descent learning algorithms. *Neural Netw*. 1999;12(1):145-151.
57. Likas A, Stafylopatis A. Training the random neural network using quasi-Newton methods. *Eur J Oper Res*. 2000;126(1):331-339.

58. Lande P, Gadewar A. Application of artificial neural networks in prediction of compressive strength of concrete by using ultrasonic pulse velocities, *IOSR. J Mech Civil Eng*. 2012;3(1):34-42.
59. Jumaat SAB, Crocker F, Wahab MHA, Radzi NHBM. Investigate the photovoltaic (PV) module performance using artificial neural. *Network (ANN), Open Systems (ICOS)*. 2016:59-64.
60. Keong KC, Mustafa M, Mohammad AJ, et al. Levenberg-Marquardt flood prediction for Sungai Isap residence. In: *2016 IEEE Conference on Systems, Process and Control (ICSPC)*; 2016:160-165.
61. Xiao L, Chen X, Zhang X A joint optimization of momentum item and Levenberg-Marquardt algorithm to level up the BPNN's generalization ability. *Math Prob Eng*. 2014;2014(1):1-11.
62. Yu H, Wilamowski BM. Levenberg-marquardt training. *Industrial Electronics Handbook*. 2011;5(1):1-10.
63. Alpaydin E, Kaynak C, Alimoglu F. Cascading multiple classifiers and representations for optical and pen-based handwritten digit recognition. In: *International Workshop on Frontiers in Handwriting Recognition (IWFHR '00)*, Amsterdam; 2000:453-462.
64. Mizutani E, Dreyfus SE. On complexity analysis of supervised MLP-learning for algorithmic comparisons. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*; 2021, 1(1): 347-352.
65. Jurafsky D, Martin JH. *Speech and Language Processing*. Vol 3. London: Pearson; 2014.
66. Peng C-YJ, Lee KL, Ingersoll GM. An introduction to logistic regression analysis and reporting. *J Educ Res*. 2002;96(1):3-14.
67. Rawlings JO, Pantula SG, Dickey DA. *Applied Regression Analysis: A Research Tool*. Springer Science & Business Media; 2001.
68. Hegre H, Karlsen J, Nygård HM, Strand H, Urdal H. Predicting armed conflict, 2010–2050. *Int Stud Q*. 2013; 57(1):250-270.
69. Hegre H. Logistisk regresjon: binomisk, multinomisk og ranger. *Tilgjengelig Fra*. 2011;2011(1).
70. Bayaga A. Multinomial logistic regression: usage and application in risk analysis. *J Appl Quantit Methods*. 2010;5(1):288-297.
71. Rajpurkar S, Bhatt DR, Malhotra P. Book recommendation system. *Int J Innov Res Sci Technol*. 2015;1(11): 314-316.
72. Nadia A-R, Al-Rousan M, Shareiah A. A fuzzy logic model of a tracking system for solar panels in northern Jordan based on experimental data. *Renew Energy Res Appl*. 2012:1-6.
73. Yano M. *Calendar, Astrology, and Astronomy*. Malden: Blackwell Publishing; 2003.
74. Cunningham D. Analysis of the Aberlemno 1 (serpent stone): evidence of accurate knowledge of the Moon's orbit in Pictish artwork. *Midnight Sci*. 2013;9(1):2012-2013.
75. Guo H, Liu G, Ding Y. Moon-based Earth observation: scientific concept and potential applications. *Int J Digital Earth*. 2018;11(1):546-557.
76. Aziz MA, Ismail N, Yassin I, Zabidi A, Ali MM. Agarwood oil quality classification using cascade-forward neural network. *Control Syst Graduate Res Colloquium*. 2015:112-115.
77. Wang Z, Khan WA, Ma HL, Wen X. Cascade neural network algorithm with analytical connection weights determination for modelling operations and energy applications. *Int J Prod Res*. 2020;58(23):7094-7111.
78. Hossin M, Sulaiman MN. A review on evaluation metrics for data classification evaluations. *Int J Data Mining Knowl Manage Process*. 2015;5(1):1-11.
79. Maria Navin JR, Balaji K. Performance analysis of neural networks and support vector machines using confusion matrix. *Int J Adv Res Sci Eng Technol*. 2016;3(1):2016-2019.
80. Gupte C, Gadewar S. Diagnosis of Parkinson's disease using acoustic analysis of voice. *Int J Sci Res Netw Secur Commun*. 2017;5(1):14-18.
81. Larochelle H, Bengio Y, Louradour J, Lamblin P. Exploring strategies for training deep neural networks. *J Mach Learn Res*. 2009;10(1):1-40.
82. Liu T, Fang S, Zhao Y, Wang P, Zhang J. Implementation of training convolutional neural networks. *Comput Vision Pattern Recogn*. 2015;2(1):1-10.
83. Sze V, Chen YH, Yang TJ, Emer JS. Efficient processing of deep neural networks. *Synth Lect Comput Archit*. 2020;15(2):1-341.

84. Yu-Wei CDC. *Machine learning with R Cookbook*. Packt Publishing Ltd; 2015:1-7.
85. Bertsimas D, King A. Logistic regression: from art to science. *Stat Sci*. 2017;32(1):367-384.
86. Cheng W, Hüllermeier E. Combining instance-based learning and logistic regression for multilabel classification. *Mach Learn*. 2009;76(1):211-225.

How to cite this article: AL-Rousan N, Mat Isa NA, Mat Desa MK, AL-Najjar H. Integration of logistic regression and multilayer perceptron for intelligent single and dual axis solar tracking systems. *Int J Intell Syst*. 2021;36:5605-5669.
<https://doi.org/10.1002/int.22525>