Evaluation of Artificial Neural Networks (ANNs) and Multivariate Adaptive Regression Splines (MARS) for Monthly Mean Land Surface Temperature (LST) Modelling– A Case Study of Aowin District, The Republic of Ghana

M. S. Peprah[1*], E. K. Larbi[2], P. O. Appau[3], M. A. Mwin [1]
[1]School of Mines and Built Environment, University of Energy and Natural Resources, Sunyani, Ghana
[2]Geo-Informatics Division, Building and Road Research Institute (CSIR-BRRI), Kumasi, Ghana
[3]Ghana National Gas Limited Company, 225 Osibisa Close, Airport West, Accra, Ghana
Email: mspeprah91@gmail.com

(Received on 11 September 2024; In final form on 9 April 2025)

***DOI**: https://doi.org/10.58825/jog.2025.19.1.185*

Abstract: Accurate and precise estimations of Land Surface Temperature (LST) are essential in climatology, agribusiness, agronomy, urban planning, aviation, and hydrology studies. In this study, the feasibility of two soft computing methods, namely, fifteen different Artificial Neural Network (ANN) architectures and the data mining model of Multivariate Adaptive Regression Splines (MARS), is evaluated for predicting the monthly mean LST of Aowin District, Ghana. Various weather prediction variables, including precipitation, relative humidity, wind speed, and temperature time series historical data spanning 37 years (from 1st January 1985 to 31st December 2022), were used. The data was obtained from a satellite database repository and used in formulating the ANN and MARS models as input (independent variables) and output (dependent variable), respectively. Five different statistical performance indicators, namely mean error (*ME*), root mean absolute error (*RMAE*), mean squared error (*MSE*), root mean squared error (*RMSE*), and standard deviation (*SD*), were used to assess the accuracy and precision of LST estimates from both the ANN and MARS models for the research area. The results demonstrate the capability of both techniques in predicting the monthly mean LST. However, the MARS model produced the best LST estimate, with statistical metrics of *ME*, *RMAE*, *MSE*, *RMSE*, and *SD* being 1.8705E-07 °C, 0.0004 °C, 3.3449 °C, 5.7835 °C, and 1.6000E-09 °C, respectively. Both ANN and MARS methods can be effectively applied for LST estimation in the research region and for studying the potential impacts of climate change dynamics globally.

Keywords: Artificial Intelligence, Artificial Neural Networks, Multivariate Adaptive Regression Splines, Data Mining Models, Machine Learning, Temperature Modelling

1. Introduction

Climate change is a worldwide phenomenon that impacts many daily life (Gupta et al., 2022; Cassie et al., 2004; Sreehari & Ghantasala, 2019). Consequently, climate studies have become essential in fields such as disaster management, agribusiness, maritime, and aerial navigation, hydroelectric power operations, entertainment events, and domestic activities (Gupta et al., 2022). Moreover, climatic data is crucial for production and operational planning at the district, regional, and national levels, especially for heating systems (Fang & Lahdelma, 2016). Weather characteristics, including temperature, relative humidity, wind speed, wind power, solar radiation, and precipitation, have been essential subjects of study in recent decades. Conventional approaches, such as linear regression models (Azari et al., 2022; Fang & Lahdelma, 2016; Dominiak & Terray, 2005; Szymanowski et al., 2013), non-homogeneous regression (Lerch & Thorarinsdottir, 2013), Numerical Weather Prediction models (Soman et al., 2010), and autoregressive integrated moving average techniques (Fang & Lahdelma, 2016; Santamaria-Bonfil et al., 2015; Soman et al., 2010; Mohandes et al., 1998), have been employed in meteorological forecasting. These models often overestimate low values and underestimate high values due to non-linearity and non-parametric connections, resulting in suboptimal decision-making. Advancements in science, technology, and computational capabilities have positioned machine learning (ML) approaches as effective instruments for weather forecasting, addressing non-linearity and non-parametric challenges (Gupta et al., 2022). Artificial Intelligence (AI) has demonstrated efficacy as a data analysis instrument for accurately forecasting meteorological conditions with minimum human involvement (Zarinkamar & Mayorga, 2021). Research indicates that machine learning approaches frequently surpass conventional statistical methods in predictive accuracy (Radhika & Shashi, 2009).

Artificial Neural Networks (ANN), a technology based on artificial intelligence, are very proficient in managing uncertainties related to real-world issues and provide resilient solutions (Asenso-Gyambibi et al., 2024; Yakubu et al., 2018; Cadenas & Rivera, 2009). Artificial Neural Networks (ANN) are straightforward to construct across several domains and regularly produce superior accuracy compared to conventional methods (Radhika & Shashi, 2009; Asenso-Gyambibi et al., 2024; Yakubu et al., 2018). Recent research indicates that artificial neural networks (ANN) have achieved significant success in modelling time series data of meteorological parameters (Azad et al., 2014; Mohandes et al., 1998; Cadenas & Rivera, 2009; Soman et al., 2010; Ghorbani et al., 2015).

Multivariate Adaptive Regression Splines (MARS) is an adaptive supervised machine learning modelling approach developed by Friedman in 1991 to handle non-linear connections (Yakubu et al., 2018). MARS is very proficient in high-dimensional issues and has shown significant potential in modelling non-linear multivariate functions (Sumi et al., 2012). It proficiently forecasts both additive and interactive effects of predictors on response variables and has been effectively utilized in domains such as ozone concentration estimation (Roy et al., 2018), sea level prediction (Raj & Gharineiat, 2021), temperature forecasting (Diaz et al., 2020), atmospheric correction modelling (Kuter et al., 2015), and various other environmental and meteorological applications.

This research presents novel methods for forecasting land surface temperature utilizing monthly weather prediction factors (precipitation, relative humidity, temperature, wind velocity) derived from historical data in the Aowin District, Ghana. The used ANN algorithms comprise Bayesian Regularization Backpropagation (BRBPANN), Levenberg-Marquardt Backpropagation (LMBPANN), Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton Backpropagation (BFGSQNBPANN), Scaled Conjugate Gradient Backpropagation (SCGBPANN), among others. Furthermore, unsupervised artificial neural network methodologies such as Radial Basis Function Artificial Neural Network (RBFANN) and Generalized Regression Artificial Neural Network (GRANN) were utilized.

The MARS data mining model was utilized as well. The efficacy of the LST forecasting models was assessed utilizing five statistical metrics: *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*. These models can assist several sectors, including agriculture, climatology, energy production, and aviation. Furthermore, they provide essential instruments for policymakers, engineers, and environmental groups to forecast land surface temperature across diverse geographical areas with climate change.

2 Study Area

The Aowin Municipal District is situated in the western region of Ghana, especially in the midwestern area of the Western Region (Figure 1), positioned between latitudes 5° 25’ N and 6° 14’ N, and longitudes 2° 30’ W and 3° 05’ W (Ehiakpor et al., 2016; Aowin Suaman District Assembly, 2006). The district is bordered to the south by Jomoro District, to the east by Wasa Amenfi, to the north by Juabeso-Bia and Sefwi-Wiawso, and to the west by the Republic of Côte d'Ivoire (Ilona, 2024; Aowin Suaman District Assembly, 2006; Asafo-Adjei & Buabeng, 2016). The land area encompasses around 2,717 square kilometers, representing 11% of the region's overall size of 23,921 square kilometers (Ghana Statistical Service, 2010; Asafo-Adjei & Buabeng, 2016). The principal settlements in the district are Enchi and Dadieso, with Enchi designated as the district capital (Ehiakpor et al., 2016; Ghana Statistical Service, 2010). Aowin Municipality possesses a varied topography, with over 50% of its total territory above 305 meters in elevation above sea level. The elevations surpass 366 meters, and these highlands, located east of Enchi, exhibit a north-northwest to south-southwest orientation (Ghana Districts, 2017; Ilona, 2024). The district receives nine months of precipitation, with a peak in May and June. The yearly average precipitation varies from 1,500 mm to 1,800 mm, while temperatures oscillate between 28°C and 37°C (Ehiakpor et al., 2016). February and March are the warmest months, whilst August experiences the lowest temperatures. Relative humidity often remains elevated, fluctuating between 75% and 80% during the wet season, and declining to around 70% for the remainder of the year (Ilona, 2024).

The district's effective drainage system improves soil fertility, facilitating the cultivation of both food and cash crops (Ehiakpor et al., 2016). Aowin Municipality is located in the forest zone and has two types of vegetation: tropical rainforest and semi-deciduous forest. The tropical rainforest predominates the landscape, encompassing nearly the entire municipality, whereas the semi-deciduous forest is confined to a small area in the northeastern region, adjacent to the Sefwi Akontombra District and Sefwi-Wiawso Municipality (Ghana Districts, 2017; Aowin Suaman District Assembly, 2006). The district comprises eight forest reserves: Tano Anwia, Tano Nimire, Tano Ehuro, Boin Tano, Jema Assemkrom, Boi River, Disue River, and Yoyo. The predominant tree species in these reserves are emire, sapele, wawa, odum, mahogany, ofram, and asanfena. The district has substantial precipitation year-round, with a rainy season extending around eight months each year (Asafo-Adjei & Buabeng, 2016; Ghana Statistical Service, 2010).

The study region has a population of 192,527, reflecting a growth rate of 4.7%. The region has 312 communities, with the elevated growth rate ascribed to migrant farmers from various regions of the country moving to the district to pursue agricultural prospects (Ghana Statistical Service, 2010). Approximately 78% of the economically active population consists of farmers involved in forestry and fishing activities (Ghana Districts, 2017). Cocoa production constitutes the principal economic activity, with 65–70% of the district's vegetative cover allocated to cocoa plantations (Ghana Statistical Service, 2010; Ehiakpor et al., 2016). Besides cocoa, the district's primary income crops are oil palm, rubber, citrus, and small-scale coffee cultivation. Prominent food crops cultivated encompass plantain, cassava, cocoyam, vegetables, rice, yam, and maize (Aowin Suaman District Assembly, 2006). The district's geology includes Upper Birimian, hornblende, and a composite formation of both sorts (Ghana Districts, 2017). The Upper Birimian strata, being younger, exhibit significant folding, with dips frequently exceeding 60°, resulting in the creation of hill ranges (Aowin Suaman District Assembly, 2006).

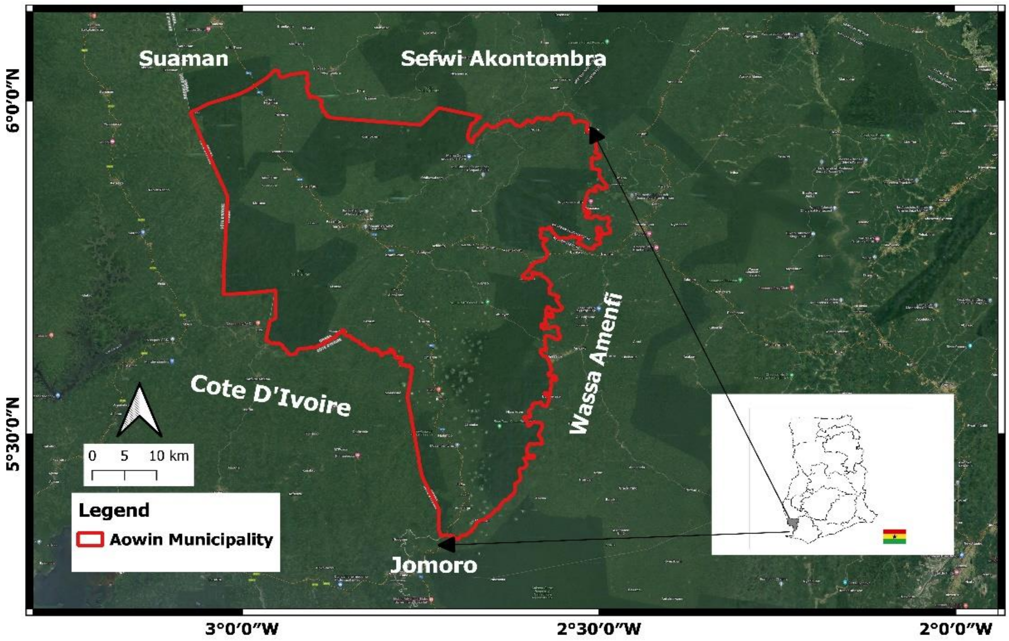

Figure 1. Map of the Study Area

3 Resources and Methods Used

3.1 Resources Used

The meteorological variables (Rainfall, Relative Humidity, Temperature, and Wind Speed) utilized in this study were sourced from the NASA climate data portal (https://power.larc.nasa.gov/data-access-viewer/). The collection has 37 years of time series data, from January 1, 1985, to December 31, 2022. The data encompasses the year, month, and mean measurements of the meteorological variables. These are the empirical observation data, and the optimal model should precisely forecast these values. Figure 2 depicts the flowchart for the model formulation.

3.2 Methods used

3.2.1 Backpropagation Artificial Neural Network (BPANN)

BPANN is a supervised machine learning technique and a variant of multilayer feedforward neural networks, including three layers: the input layer, hidden layer, and output layer. This research utilized Rainfall, Relative Humidity, Temperature, and Wind Speed as input variables, with Temperature as the outcome variable. In the BPANN model construction, it is essential to normalize the dataset to mitigate the effects of differing dimensions and units of the variables (Huang et al., 2016). The initial data utilized for BPANN iteration and model development are represented in many units, each with distinct physical interpretations (Peprah & Larbi, 2021). To maintain consistency in the BPANN model, datasets are routinely standardized to a designated interval, such as [-1, 1], [0, 1], or another scaling standard. This investigation standardized the input and output variables within the range of [-1, 1], as specified by Equation (1) (Mueller & Hemond, 2013).

$$z_i = z_{\min} + \frac{(z_{\max} - z_{\min}) \times (x_i - x_{\min})}{(x_{\max} - x_{\min})} \qquad (1)$$

In this context, z_i denotes the normalized data, x_i signifies the observed temperature values, while x_{min} and x_{max} indicate the minimum and maximum measured temperature values, with z_{min} and z_{max} assigned values of 1 and -1, respectively. To ascertain the ideal weight configuration and the most effective learning technique for the research region, the network was trained utilizing thirteen distinct machine learning algorithms, including Levenberg-Marquardt, Bayesian Regularization, and the Resilient Backpropagation technique. Scaled Conjugate Gradient, Fletcher-Powell Conjugate Gradient, Polak-Ribiere Conjugate Gradient, Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton. Conjugate Gradient with Powell/Beale Restarts. Gradient Descent, Gradient Descent with Momentum, Gradient Descent with Adaptive Learning Rate, Gradient Descent with Momentum and Adaptive Learning Rate, and One-Step Secant.

The dataset was divided into training (70%) and testing (30%) subsets. The training aim was to identify the weight configuration among neurons that globally minimizes the error function. The primary purpose of the testing set was to evaluate the generalization capability of the trained network. The training was halted when the error on the testing dataset started to rise. The optimal model was chosen based on the minimal evaluation and validation statistical metrics: mean error (*ME*), root mean absolute error (*RMAE*), mean square error (*MSE*), root mean square error (*RMSE*), and standard deviation (*SD*). The metrics are further discussed in the model assessment and validation section.

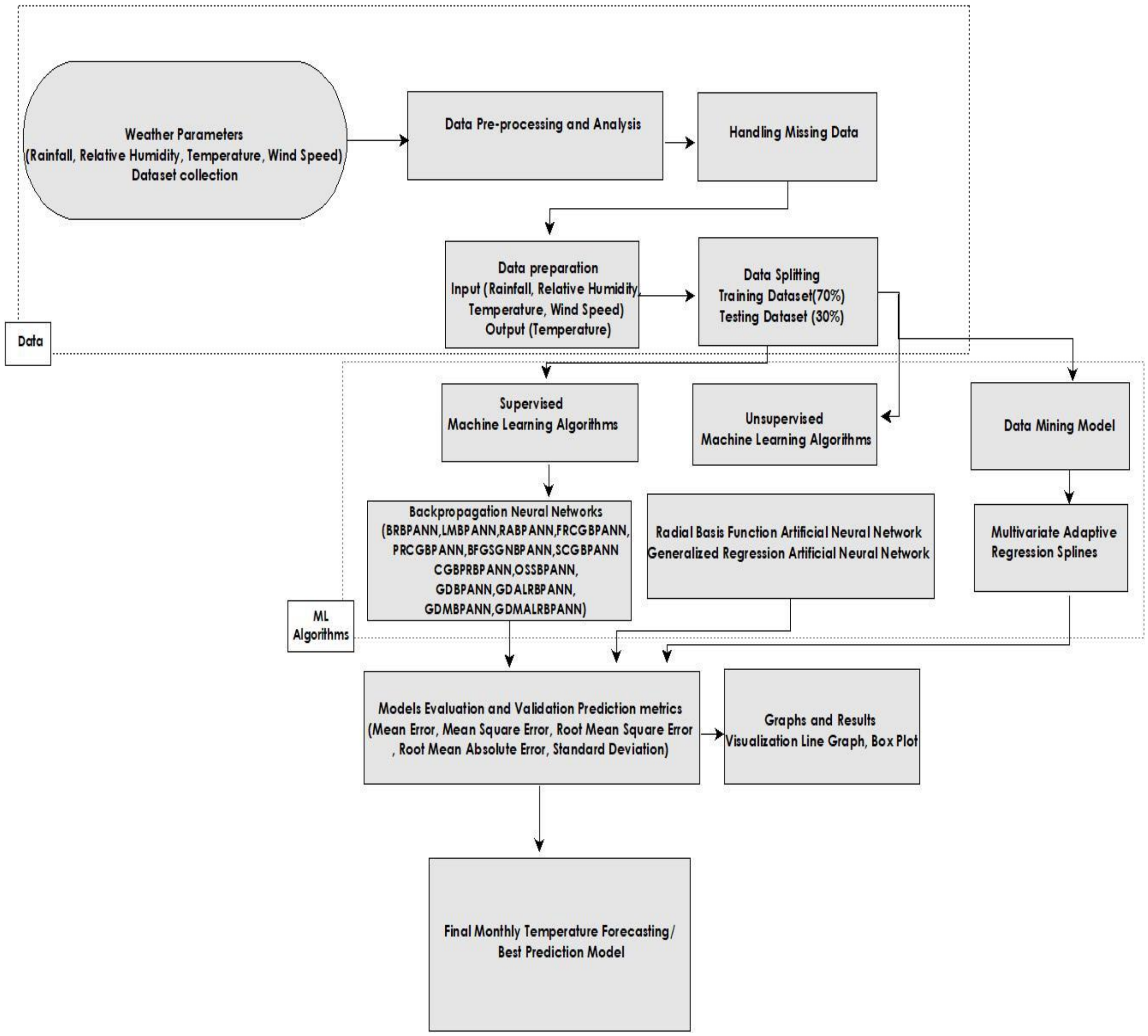
Weather Parameters
(Rainfall, Relative Humidity, Temperature, Wind Speed)
Dataset collection
Data Pre-processing and Analysis
Handling Missing Data
Data preparation
Input (Rainfall, Relative Humidity, Temperature, Wind Speed)
Output (Temperature)
Data Splitting
Training Dataset(70%)
Testing Dataset (30%)
Data
Supervised
Machine Learning Algorithms
Unsupervised
Machine Learning Algorithms
Data Mining Model
Backpropagation Neural Networks
(BRBPANN,LMBPANN,RABPANN,FRCGBPANN,
PRCGBPANN,BFGSGNBPANN,SCGBPANN
CGBPRBPANN,OSSBPANN,
GDBPANN,GDALRBPANN,
GDMBPANN,GDMALRBPANN)
Radial Basis Function Artificial Neural Network
Generalized Regression Artificial Neural Network
Multivariate Adaptive
Regression Splines
ML
Algorithms
Models Evaluation and Validation Prediction metrics
(Mean Error, Mean Square Error, Root Mean Square Error
, Root Mean Absolute Error, Standard Deviation)
Graphs and Results
Visualization Line Graph, Box Plot
Final Monthly Temperature Forecasting/
Best Prediction Model

Figure 2. Flowchart of the Research Methodology

The Tansig and Purelin hyperbolic activation functions were employed for the hidden and output layers, respectively, during network training to include non-linearity into the network. The hyperbolic tangent function is defined in Equation (2).

$$V_{(x)} = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \qquad (2)$$

x denotes the cumulative weighted inputs. BPANN is an iterative machine learning training methodology, wherein the network was iteratively trained by adjusting the number of hidden neurons from 1 to 50, until the ideal model architecture was attained.

Bayesian Regularization Backpropagation Artificial Neural Network (BRBPANN)

BRBPANN, using MATLAB syntax, written as trainbr, is a training method that changes the weights and bias values using Levenberg-Marquardt optimization (Kaur & Salaria, 2013). This technique minimizes a mixture of squared errors and weights to achieve the ideal balance for constructing a network that generalizes well (Kaur and Salaria, 2013). According to Foresee and Hagan (1997), the strategy for enhancing generalization is known as regularization. The purpose of training is to minimize the sum of squared errors, E_D. This indicates that the first training goal function is $F = E_D$. However, regularization introduces an extra factor, E_W, and the objective function is thus defined as illustrated in Equation (3) (Foresee and Hagan, 1997).

$$F = \beta E_D + \alpha E_w \qquad (3)$$

E_W represents the sum of the squared network weights, whereas E_D is the total of network errors; α and β are the parameters of the objective function. Foresee and Hagan (1997) assert that the relative magnitude of these characteristics determines the focus of training. When α is significantly less than β, the algorithm focuses on minimizing network mistakes; conversely, when α is significantly more than β, the training prioritizes the decrease of weight size, which may lead to increased network faults, ultimately producing a more streamlined network. The difficulty with regularization is determining the appropriate values for these parameters. α and β are established by Bayes' rule, with a comprehensive methodology for their calculation explained by Foresee and Hagan (1997).

Levenberg-Marquardt Backpropagation Artificial Neural Network (LMBPANN)

The LMBPANN algorithm, represented by the MATLAB syntax trainlm, is an iterative method for minimizing a multivariate error function E, defined as the sum of the squares of the discrepancies between the actual output (y_i) and the goal output (t_i) (Adeoti & Osanaiye, 2013), as expressed in Equation (4):.

$$E = \frac{1}{2}\sum\left(y_i - t_i\right) \quad (4)$$

The LMBPANN was developed to achieve second-order speed without the necessity of calculating the Hessian matrix. Nonetheless, the Hessian matrix (H) and the gradient (g) may be estimated using Equations (5) and (6), respectively, when the performance function is expressed as a sum of squares.

$$H = J^T J \quad (5)$$

$$g = J^T e \quad (6)$$

J represents the Jacobian matrix, which comprises the first derivatives of the network errors concerning the biases and weights, whereas e is the network error vector. The Jacobian matrix may be derived by a conventional backpropagation method, which is considerably less intricate than the computation of the Hessian matrix (Baghirli, 2015). The LMBPANN method utilizes this approximation of the Hessian matrix in the subsequent Newton-like update, as seen in Equation (7):

$$w_{i+1} = w_i - \left[J^T J + \mu I\right]^{-1} J^T e \quad (7)$$

w denotes connection weights, µ signifies the damping term, and I represents the identity matrix. The LMBPANN employs a mix of the Gauss-Newton technique and gradient descent in its iterative procedure (Peprah and Larbi, 2021). When µ is 0, it transforms into a Gauss-Newton technique, employing the estimated Hessian matrix. When µ is substantial, it functions as a gradient descent algorithm with a minimal step size. Newton's approach exhibits superior speed and precision in proximity to an error minimum; hence, the objective is to transition to Newton's method expeditiously. Consequently, µ diminishes following each successful step (reduction in the performance function) and is augmented just when a tentative step might enhance the performance function. Consequently, the performance function will consistently diminish with each iteration of the algorithm (Baghirli, 2015).

Resilient Algorithm Backpropagation Artificial Neural Network (RABPANN)

The RABPANN technique, represented in MATLAB as (trainrp), is a neural network training method that operates analogously to the conventional backpropagation procedure. The fundamental difference resides in the method of updating the connection weights (Prasad et al., 2013). In traditional backpropagation, the update is determined by the size of the partial derivative, as expressed in Equation (8).

$$\Delta w_{jk}(m) = \alpha \times x_j(m) \times \delta_k(m) \quad (8)$$

where α represents the learning rate, $x_j(m)$ signifies the inputs transmitted back to the i_{th} neuron at time step m, and $\delta_k(m)$ defines the associated error gradient. For the RABPANN, an individual delta Δ_{jk} is calculated to ascertain the magnitude of the weight w_{jk} update for each connection. The learning rule defined in Equation (9) is employed in the computation of Δ_{jk}.

$$\Delta_{jk}(m) = \begin{cases} \Delta_{jk}(m-1) \times n^{+}, \\ if\ \frac{\partial E}{\alpha w_{jk}}(m-1) \times \frac{\partial E}{\partial w_{jk}}(m). > 0 \\ \Delta_{jk}(m-1) \times n^{-}, \\ if\ \frac{\partial E}{\partial w_{jk}}(m-1) \times \frac{\partial E}{\partial w_{jk}}(m) < 0 \\ \Delta_{jk}(m-1), else \end{cases} \quad (9)$$

where $0 < n^{-} < 1 < n^{+}$.

Significantly, in the RABPANN method, the weight adjustment is determined not by the amount of the derivatives but by the behavior of the signs of two consecutive derivatives. Whenever the partial derivative of the relevant weight w_{jk} alters its sign, it signifies that the previous update was excessively big, causing the algorithm to bypass a local minimum. In this scenario, the update value $\Delta_{jk}$is diminished by the factor n^{-}. If the derivative maintains its sign, the updated value is somewhat raised to enhance convergence in shallow areas (Riedmiller and Braun, 1992; Prasad et al., 2013). The weight update rule stays consistent with Equation (10), except when the partial derivative reverses sign. In such instances, the prior update step that caused a leap beyond the minimum is reversed by Equation (11). Upon a change of sign, the adaptation process recommences. This modification of update values and weights occurs each time the complete pattern set is submitted to the network.

$$\Delta w_{jk}(m) \begin{cases} -\Delta_0, if\ \frac{\partial E}{\partial w_{jk}}(m) > 0 \\ +\Delta_0, if\ \frac{\partial E}{\partial w_{jk}}(m) < 0 \\ 0, else \end{cases} \quad (10)$$

$$\Delta_{jk}(m) = -\Delta_{jk}(m-1), \quad if\ \frac{\partial E}{\partial w_{jk}}(m) \times \frac{\partial E}{\partial w_{jk}}(m-1) < 0 \quad (11)$$

Fletcher-Reeves Conjugate Gradient Backpropagation Artificial Neural Network (FRCGBPANN)

The FRCGBPANN technique, represented in MATLAB syntax as (traincgf), is a variant of the Conjugate Gradient method established by Fletcher and Reeves in 1964. This approach is capable of training any network provided that its weights, net inputs, and transfer functions possess derivative functions. Backpropagation is utilized to compute the derivatives of performance concerning the weight and bias vectors *M*. Each vector *Mi* is modified by Equation (12), stated as:

$$M = M + \alpha(\delta M) \quad (12)$$

where δM denotes the search direction, with α being the parameter chosen to optimize performance along that trajectory. The line search tool is employed to identify the minimum point. The primary search direction is the negative gradient of the performance function. In following iterations, the search direction is determined using the updated gradient and the preceding search direction, as illustrated in Equation (13).

$$\delta M = -gM + \beta(\delta M_{old}) \quad (13)$$

where gM denotes the gradient. The parameter β can be calculated by many methods. The Fletcher-Reeves variant of the conjugate gradient technique is computed using Equation (14).

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \quad (14)$$

where $g_{k-1}^T g_{k-1}$ represents the squared norm of the preceding gradient, and the current gradient's squared norm is denoted as well.

Polak-Ribiere Conjugate Gradient Backpropagation Artificial Neural Network (PRCGBPANN)

The PRCGBPANN technique, represented in MATLAB syntax as (traincgp), is an alternative iteration of the conjugate gradient method introduced by Polak and Ribiére in 1969. In this method, the search direction (p) for each iteration is defined by Equation (15).

$$P_k = -g_k + \beta_{kPk-1} \quad (15)$$

The constant βk for the Polak-Ribiére update is calculated using Equation (16).

$$\beta_k = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}} \quad (16)$$

Equation (16) represents the inner product of the prior gradient change with the current gradient, normalized by the square of the prior gradient's norm.

Scaled Conjugate Gradient Backpropagation Artificial Neural Network (SCGBPANN)

The SCGBPANN training technique, represented in MATLAB syntax as (trainscg), is a conjugate gradient approach initially formulated by Moller in 1993. The BPANN machine-learning technique fundamentally modifies the weights in the direction of steepest descent, namely the greatest negative gradient direction (Baghirli, 2015). This direction represents the steepest decline of the performance function. Hagan et al. (1996) discovered that although the steepest descent path yields a swift reduction, it does not inherently provide the quickest convergence.

In the construction of the SCGBPANN model, a search is conducted in this manner, resulting in typically quicker convergence than the steepest descent while maintaining the error reduction attained in prior phases (Baghirli, 2015; Kisi and Uncuoglu, 2005). This orientation is termed the conjugate direction (Baghirli, 2015). Furthermore, the step size is modified at each iteration, and a search is performed in the conjugate gradient direction to ascertain the ideal step size (Baghirli, 2015). This method aids in reducing the performance function along that trajectory.

SCGBPANN starts its process by exploring the steepest descent direction during the first iteration, as delineated in Equation (17) (Baghirli, 2015). Additionally, SCGBPANN uses line search methodologies to estimate the step size, hence eliminating the necessity of calculating the Hessian matrix and identifying the best displacement along the prevailing search direction, as seen in Equation (18). The subsequent search direction is established by conjugating with the preceding search direction by Equation (19). The standard method for deriving the new search direction entails integrating the current steepest descent direction with the prior direction (Baghirli, 2015; Sandhu and Chhabra, 2011; Hagan et al., 1996).

$$p* = -g* \quad (17)$$

$$x_{k+1} = x_k + \alpha_k g_k \quad (18)$$

$$\rho_k = -g_k + \beta_k \rho_{k-1} \quad (19)$$

The several categories of conjugate algorithms are differentiated by the method of calculating the factor βk (Baghirli, 2015; Kisi and Uncuoglu, 2005). Likewise, there exists a significant likelihood of employing an alternative method for calculating the step size instead of the line search strategy. The objective is to integrate the model trust region methodology (Baghirli, 2015), recognized from the LMBPANN machine learning algorithm, with the SCGBPANN technique. This methodology is referred to as the SCG and is further explained in Moller's (1993) publications. This model method, as indicated by Equation 19, represents the approximation of the Hessian matrix. ε represents the total error function, while ∇ε denotes its gradient. Scaling factors λk and δk are added to approximate the Hessian matrix, initialized by the user at the algorithm's commencement, constrained so that $0 < \lambda_k < 10^{-b}$ and $0 < \delta_k < 10^{-4}$ (Baghirli, 2015). In the construction of the SCGBPANN model, the computation and orientation of the βk factor are conducted by Equations (20) and (21), respectively.

$$\mathrm{S}_k = \frac{\partial\left(\omega_k + \partial_k \dot{\rho}_k\right) - g(\dot{\omega}_k)^1}{\partial_k} + \lambda_k \rho_k \quad (20)$$

$$\beta_k = \frac{\left|g_{k+1}\right|^2 - g_{k+1} g_k^T}{g_k^T g_k} \quad (21)$$

$$\rho_{k+1} = -g_{k+1} + \beta_k \rho_k \quad (22)$$

Furthermore, design parameters are separately modified at each iteration stage, which is a critical component for the algorithm's success. This method provides a considerable benefit compared to line search-based algorithms (Baghirli, 2015). Furthermore, the model technique employs a quadratic approximation in the vicinity of a point ω, represented by $\varepsilon_{qw}(y)$, as specified in Equation (23):

$$\varepsilon_{q\omega}(y) = \varepsilon(\omega) + \acute{\varepsilon}(\omega)_y^r + \frac{1}{2} y^r \ddot{\varepsilon}(\omega) y \quad (23)$$

To ascertain the minimum of $\varepsilon_{qw}(y)$, It is essential to identify the key locations (Peprah and Larbi, 2021). The critical points provide solutions to the linear system established by Moller (1993). The SCG method is adaptable and capable of training any network, given that its weight, net input, and transfer functions possess derivative functions (Peprah and Larbi, 2021; Sandhu and Chhabra, 2011).

Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton (BFGSQNBPANN)

The BFGSQNBPANN technique, referred to as (trainbfg) in MATLAB, is a Quasi-Newton approach within the second derivative line search category for addressing unconstrained optimization issues (Ibrahim et al., 2014). This approach employs a quadratic Taylor approximation of the goal function near a specific point, as defined in Equation (24) (Biglari and Ebadian, 2015):

$$f(x+\delta) \approx q(\delta) = f(x) + \delta^T g(x) + \frac{1}{2} \delta^T H(x) \delta \quad (24)$$

where g(x) represents the gradient vector and H(x) denotes the Hessian matrix. The requisite condition for a local minimum of g(δ) concerning δ is expressed in the linear system defined in Equation (25) as follows:

$$g(x) + H(x)\delta = 0 \quad (25)$$

This subsequently yields the Newton direction δ for the line search, as defined by Equation (26):

$$\delta = -H(x)^{-1} - g(x) \quad (26)$$

The precise Newton direction, a fundamental characteristic of Newton-type approaches, is dependable when the Hessian matrix is present, positive definite, and the discrepancy between the actual objective function and its quadratic approximation is minimal. Quasi-Newton techniques utilize matrices that estimate the Hessian matrix or its inverse, instead of calculating the Hessian matrix precisely, as is done in Newton-type methods. These matrices are often represented as $\beta \approx H$ and $D \approx H^{-1}$. They are modified at each iteration and may be generated using a variety of strategies, from basic procedures to more sophisticated approaches. This modelling approach utilizes an updating algorithm to approximate the Hessian matrix $H(x^*)$, as defined in Equation (27):

$$\beta_{i+1} = \beta_i - \frac{\beta_i \delta_i \delta_i}{\delta_i^T \beta_i \delta_i} + \frac{y_i y_i^T}{y_i^T \delta_i} \quad (27)$$

where $s_i = x_{i+1} - x_i$, $y_i = g_{i+1} - g_i$. Initially, β_0 may be assigned any symmetric positive definite matrix, such as the identity matrix, which is frequently utilized. This modelling approach exhibits superlinear convergence, with resource intensity approximated at $O(n^2)$ each iteration for an n-component argument vector.

Conjugate Gradient with Powell/Beale Restarts Backpropagation Artificial Neural Network (CGPBRBPANN)

The CGPBRBPANN algorithm is represented in MATLAB using the syntax traincgb. Sandhu and Chhabra (2011) state that in conjugate gradient algorithms, the search direction is sometimes reverted to the negative gradient. The typical reset transpires when the quantity of network weights and biases matches the number of iterations. Nevertheless, other reset methodologies can improve training efficiency. One method is the Powell/Beale Restarts methodology (Powell, 1977; Beale, 1972). This strategy reinitiates the search direction when there is negligible orthogonality remaining between the current gradient and the preceding gradient (Sandhu and Chhabra, 2011). Equation (28) is employed to ascertain the appropriate moment to revert the search direction to the negative gradient.

$$\left|g_{n-1}^T g_n\right| \geq 0.2 \left\|g_n\right\|^2 \quad (28)$$

where g_n is the gradient at the n_{th} iteration. Upon fulfillment of this condition, the search direction is recalibrated to the negative gradient. This model is capable of training any network provided that its weights, net inputs, and transfer functions possess derivative functions. Backpropagation is employed to compute the derivatives of performance concerning the weight and bias vectors M. Each vector M is modified according to Equation (11). The line search tool is utilized to identify the minimum point.

One Step Secant Backpropagation Artificial Neural Network (OSSBPANN)

The OSSBPANN algorithm, represented in MATLAB with the syntax (trainoss), seeks to integrate conjugate gradient techniques with Quasi-Newton (secant) approaches (Mukkamala et al., 2003). This model does not retain the entire Hessian matrix but presumes that the Hessian at each iteration is the identity matrix. This method permits the computation of the new search direction without the necessity of inverting the matrix (Mukkamala et al., 2003).

The approach is capable of training any network provided that its weights, net input, and transfer functions possess derivative functions. Backpropagation is employed to compute the derivatives of performance concerning the weight and bias vectors M. Each vector M_i is modified by Equation (11), analogous to conjugate gradient models. The line search function identifies the minimum point. The first search direction is the negative gradient of performance. In succeeding iterations, the search direction is determined by the current gradient and the variations in weights and gradient from the preceding iteration, as defined by Equation (29).

$$\delta M = -gM + \alpha C\left(M_{step}\right) + \beta C\left(\delta gM\right) \quad (29)$$

where gM represents the gradient, M_{step} denotes the alteration in weights from the preceding iteration, δgM signifies the variation in the gradient from the prior iteration, and αC and βC are the combinatorial scalar products of gM, M_{step}, and δgM.

Gradient Descent Backpropagation Artificial Neural Network (GDBPANN)

The GDBPANN model is represented in MATLAB using the syntax (traingd). In the construction of the GDBPANN model, the weights and biases are adjusted by the negative gradient of the performance function (Moini and Lakizadeh, 2011). Backpropagation is employed to compute the derivatives of performance functions, ϑ, concerning the weight ω and bias vector M. Each vector M_i is modified by the gradient descent, as delineated in Equation (30) as follows:

$$\delta M = \alpha \times \frac{\delta \vartheta}{\delta x} \quad (30)$$

In this context, α represents the learning rate. The learning rate is applied to the negative gradient to calculate adjustments to the weights and biases. An elevated learning rate produces larger increments, potentially destabilizing the algorithm. In contrast, a diminished learning rate yields smaller increments, which may prolong the convergence of the algorithm.

Gradient Descent with Adaptive Learning Rate Backpropagation Artificial Neural Network (GDALRBPANN)

In MATLAB, GDALRBPANN is represented by the syntax (traingda). In contrast to the typical traingd method, which maintains a constant learning rate during the training phase, traingda utilizes an adaptable learning rate. The algorithm's performance is significantly influenced by the configuration of the learning rate (Peteiro-Barral and Guijarro-Berdinas, 2013). Should the learning rate be excessively elevated, the algorithm may fluctuate and exhibit instability. If the learning rate is excessively small, the method may need an extended duration to converge. Establishing the ideal learning rate before training is difficult, as it fluctuates throughout the training process as the algorithm navigates the performance landscape. To resolve this, the traingda method adaptively modifies the learning rate, striving to maximize the step size while ensuring stability. The learning rate adjusts according to the intricacy of the local error landscape (Peteiro-Barral and Guijarro-Berdinas, 2013). In the training algorithm, the initial network output and error are computed. During each iteration, updated weights and biases are determined utilizing the current learning rate, followed by the computation of new outputs and mistakes.

Gradient Descent with Momentum Backpropagation Artificial Neural Network (GDMBPANN)

In MATLAB, GDMBPANN is represented by the syntax traingdm. The traingdm model enables a network to adapt to both the local gradient and recent trends in the error surface, functioning as a low-pass filter (Garcez et al., 2008). Momentum enables the network to disregard minor elements in the error landscape. In the absence of momentum, the network may become ensnared in superficial local minima. The network may navigate past such entrapments with momentum. The training procedure relies on two parameters: the learning rate, α, and the momentum constant, γ. The momentum constant, γ, quantifies momentum and ranges from 0 (indicating no momentum) to values approaching 1 (indicating substantial momentum). A momentum constant of 1 renders the network entirely unresponsive to the local gradient, impeding effective learning. Backpropagation is employed to compute the derivatives of the performance function ϑ concerning the weights ω and bias vectors M. Each vector M_i is modified by gradient descent with momentum, as delineated in Equation (31):

$$\delta M = \gamma \times \left(\delta M_{previous}\right) + \alpha \times (1-\gamma) \times \frac{\delta \vartheta}{\delta M} \quad (31)$$

where $\delta M_{previous}$ denotes the prior alteration to the weight or bias.

Gradient Descent with Momentum and Adaptive Learning Rate (GDMALRBPANN)

GDMALRBPANN is represented in MATLAB using the syntax (traingdx). The traingdx method integrates an adaptable learning rate with momentum-based training. This model resembles traingda, but incorporates the momentum coefficient, γ, as an extra training parameter. The algorithm is capable of training any network, contingent upon the existence of derivatives for its weights, net inputs, and transfer functions. Backpropagation is employed to compute the derivatives of the performance function, ϑ, concerning the weight and bias vectors, M. Each vector, M_i, is modified utilizing gradient descent with momentum, as delineated in Equation (32):

$$\delta M = \gamma \times \left(\delta\, M_{previous}\right) + \alpha \times \gamma \times \frac{\delta \vartheta}{\delta M} \quad (32)$$

where $\delta M_{previous}$ denotes the preceding alteration to the weight or bias, and α represents the learning rate. In each cycle, if performance declines relative to the established target, the learning rate is augmented by a factor (often 1.05). If the performance improves by more than the factor (usually 1.04), the learning rate is modified by a different factor (typically 0.7), and the modification that enhanced the performance is not implemented.

3.2.2 Model Development

This study assessed the predictive efficacy of supervised artificial neural network (ANN) models for forecasting monthly mean land surface temperature (LST) using thirteen distinct training techniques. The essential parameters for fine-tuning were established in advance, and the model architecture was derived from prior research conducted by Hornik et al. (1989), Braspenning et al. (1995), and Beale et al. (2017). The selected model architecture included a single hidden layer utilizing a hyperbolic tangent transfer function and an output layer employing a linear transfer function. A trial-and-error technique was employed to ascertain the appropriate number of neurons in the hidden layer, examining a spectrum from 1 to 50 neurons. This investigation determined that employing 1 to 50 neurons in the hidden layer produced optimal outcomes for the backpropagation artificial neural network (BPANN) models in this study. Nevertheless, certain specifics of the trial-and-error approach were not disclosed.

Thus, the selected model architecture for this research was [4 – 1 – 1], signifying 4 inputs (Rainfall, Relative Humidity, Temperature, Wind Speed), one hidden layer, and one output layer (Temperature). This framework was employed to assess the efficacy of various training methods. The BPANN models underwent training for 5000 epochs, utilizing a learning rate of 0.03 and a momentum coefficient of 0.9. The MATLAB R2018a software was used to execute the BPANN utilizing the thirteen procedures delineated in table 1. The MATLAB environment was run on a machine equipped with an Intel(R) Core(TM) i5-7200U CPU working at 2.5GHz, a 2.71 GHz processor, 12.0 GB of RAM, a 64-bit operating system, and an X64-based processor.

In unsupervised artificial neural networks, including the RBFANN and GRANN models, training significantly depended on the width parameter (spread constant). Thus, the ideal value for this parameter was established using a sequential trial-and-error method during each iterative training procedure, spanning from 0 to 1.

3.2.3 Radial Basis Function Artificial Neural Network (RBFANN)

The RBFANN model is an unsupervised learning approach grounded in functional approximation. It comprises three functionally separate layers: an input layer, a hidden layer, and an output layer. The input layer consists of sensory units that link the network to its surroundings. The second layer, known as the hidden layer, executes a nonlinear transformation from the input space to the hidden space. The output layer is linear, delivering the network's response to the activation pattern applied at this layer. This study examined Rainfall, Relative Humidity, Temperature, and Wind Speed as input factors, with Temperature as the outcome variable. The dataset utilized for model formulation was partitioned into training data, accounting for 70% of the entire dataset, and testing data, representing 30%. The RBFANN serves as an accurate interpolator (Erdogan, 2009); hence, a linear function is employed in the input neurons, and the connections between the input and hidden layers remain unweighted (Kaloop et al., 2017). This study employs the Gaussian function, with the output neuron representing the summation of the weighted hidden output layer, as illustrated in Equation (33) (Erdogan, 2009).

Table 1. Supervised Backpropagation Training Functions and their Respective Algorithms

Training Function Syntax in MATLAB	**Algorithm Type**
trainlm	Levenberg-Marquardt
trainbr	Bayesian Regularization
trainbfg	Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton
traincgb	Conjugate Gradient with Powell/Beale Restarts
traincgp	Polak-Ribiere Conjugate Gradient
traincgf	Fletcher-Reeves Conjugate Gradient
traingd	Gradient Descent
traingdm	Gradient Descent with Momentum
traingda	Gradient Descent with Adaptive Learning Rate
trainscg	Scaled Conjugate Gradient
traingdx	Gradient Descent with Momentum and Adaptive Learning Rate
trainoss	One Step Secant
trainrp	Resilient algorithm Backpropagation

$$y(x) = \sum_{j=1}^{n} \kappa_j \chi_j(x) \qquad (33)$$

where n denotes the quantity of hidden neurons, $x \in R^M$ signifies the input, k_j indicate the output layer weights of the radial basis function network, and $\chi_j(x)$ represents the Gaussian radial basis function, as articulated in Equation 34 (Srichandan, 2012; Idri et al., 2010):

$$\chi_j(x) = e\left(\frac{-\left\|x_i - c_j\right\|^2}{\sigma_j^{\,2}}\right) \qquad (34)$$

where $c_j \in R^m$ and σ represent the center and breadth of the jth hidden neuron, respectively, $\| \quad \|$ signifies the Euclidean distance.

3.2.4 Generalized Regression Artificial Neural Network (GRANN)

GRANN was initially presented by Specht (1991) as a variant of the RBFANN, grounded in kernel regression networks (Hannan et al., 2010; Peprah and Larbi, 2021), with a one-pass learning approach and a highly parallel architecture (Dudek, 2011). GRANN has four layers: the input layer, the pattern layer (radial basis layer), the summation layer, and the output layer. This research identified Rainfall, Relative Humidity, Temperature, and Wind Speed as input factors, with Temperature as the output variable. The quantity of input units in the first layer is contingent upon the overall number of observational parameters. This layer is linked to the pattern layer, wherein each neuron signifies a training pattern along with its associated output. The pattern layer is subsequently linked to the summation layer. The summation layer comprises two forms of summation: a singular division unit and a summation unit (Hannan et al., 2010; Kumi-Boateng and Peprah, 2020). The summation layer, in conjunction with the output layer, normalizes the output datasets. Radial basis and linear activation functions are employed in the hidden and output layers, respectively, during the neural network training. Every unit in the pattern layer is linked to two neurons in the summation layer. One neuron calculates the total of the weighted responses of the pattern, whereas the other neuron determines the unweighted outputs of the pattern neurons. The output layer computes the ratio of each neuron unit's output to the others, resulting in the estimated output variables, as shown in Equation (35):

$$y_i = \frac{\sum_{i=1}^{n} y_i . \exp - G(x, \chi_i)}{\sum_{i=1}^{n} \exp - G(x, \chi_i)} \qquad (35)$$

where y_i represents the weighted connection between the ith neuron in the pattern layer and the summation neuron, and the number of training patterns is denoted as G, which is defined by Equation 36 as:

$$G(x, \chi_i) = \sum_{k=1}^{m} \frac{(\chi_i - \chi_{ik})^2}{\sigma} \qquad (36)$$

where m denotes the quantity of elements in an input vector, and x_i and x_{ik} signify the j_{th} member of x and x_i, respectively. Throughout neural network training, the spread parameter was adjusted between 0 and 1 until the output with the least residuals, according to model assessment criteria, was obtained. The identical technique was employed for the training of the RBFANN model algorithm.

3.2.5 Multivariate Adaptive Regression Splines Model (MARS)

The MARS model, initially presented by Friedman (1991), is a non-linear, non-parametric regression-based supervised machine learning technique. MARS is a sequential linear regression method adept at managing higher-dimensional inputs (independent variables). It accomplishes this at equal intervals to investigate intricate and non-linear correlations between the response (dependent) and input (independent) variables (Raj and Gharineiat, 2021). The MARS model generates predictions by analysing the correlations between response and predictor variables (Yakubu et al., 2018).

The training datasets are segmented into splines, which consist of distinct piecewise linear segments exhibiting varying gradients (Chen and Cao, 2014). The model contains a weighted summation of basis functions (BFs) organized in pairs, established according to a knot and subgroups to delineate an inflection zone (Raj and Gharineiat, 2021). The MARS model delineates data either globally or by linear regression between any two knots. The linear combination of the basis functions is expressed in Equation (37) as:

$$f(x) = \alpha_0 + \sum_{i=0}^{n} \alpha_i \beta F(x) \qquad (37)$$

In this context, $\alpha_{i \cdots n}$ represents the unknown coefficients estimated using the least squares approach, n denotes the number of terms included in the model, and βF(x) refers to the basis function derived from knots of a piecewise linear basis function. In the creation of the MARS model, the basis functions (BFs) are chosen according to the generalized cross-validation function (GCV) specified in Equation (38) as follows:

$$GCV = \frac{\frac{1}{N} \sum_{i=1}^{n} \left[y_i - \tilde{y}\right]^2}{\left[1 - \frac{C(M)}{N}\right]^2} \qquad (38)$$

where n represents the number of data points, y_i denotes the actual values of the data points, $\tilde{y}_i$ signifies the predicted values for the data points, and *C(M)* is the penalty factor described by Equation (39) as:

$$C(M) = M + \delta(M) \qquad (39)$$

where, ∂ represents the cost penalty factor associated with each basis function in the optimization process. Overfitting may occur when an excessive number of basis functions are chosen during the forward phase; hence, it is crucial to exclude some basis functions in the backward phase to identify the best model.

3.2.6 Model Evaluation and Validation

The assessment of the best model entails evaluating its prediction efficacy on separate validation datasets and comparing its projections with the recorded LST observations. The validation procedure involves the computation of evaluation metrics like Mean Error (*ME*), Root Mean Absolute Error (*RMAE*), Mean Square Error (*MSE*), Root Mean Square Error (*RMSE*), and Standard Deviation (*SD*). The mathematical formulations for these metrics are provided in Equations (40) through (44).

$$ME = \frac{1}{n}\sum_{i=1}^{n}\left(\alpha_i - \beta_i\right) \qquad (40)$$

$$RMAE = \sqrt{\frac{1}{n}\sum_{i=1}^{N}\left|\alpha_i - \beta_i\right|} \qquad (41)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\alpha_i - \beta_i\right)^2 \qquad (42)$$

$$RMSE = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\left(\alpha_i - \beta_i\right)\right)^2} \qquad (43)$$

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\mu - \bar{\mu}\right)^2} \qquad (44)$$

where *n* represents the total number of observations, α_i and β_i symbolize the measured and forecast land surface temperature (LST) values from the model algorithms, μ signifies the residual between the measured and predicted temperature data, $\overline{\mu\mu}$ indicates the mean of the residuals, and i is an integer ranging from 1 to n.

4 Results and Discussion

4.1 Results

To ascertain the most effective supervised machine learning algorithm (BRBPANN, LMBPANN, SCGBPANN, FRCBPANN, PRCGBPANN, RABPANN, BFGSQNBPANN, CGPBRBPANN, GDBPANN, GDMBPANN, GDALRBPANN, GDMALRBPANN, OSSBPANN), numerous iterative training sessions were executed to attain the optimal model outcomes. The ideal solution for the two unsupervised machine learning algorithms (RBFANN and GRANN), resulting in a smooth function, was achieved after several training iterations, adjusting the spread parameter from 0 to 1, with a maximum of 50 hidden neurons. The MARS model utilized 21 basis functions throughout the training and testing stages for forecasting monthly LST time series historical data. tables 2 to 4 illustrate an almost perfect correlation between the observed and anticipated output values (Temperature) for each model. This is further evidenced by the minimal values of the model assessment and validation metrics, including *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*, observed during the training and testing of the predictable temperature values. The discrepancies between the observed and predicted LST values in both training and testing outcomes are minimal, as indicated by the employed statistical metrics. The outcomes obtained from both supervised and unsupervised ANN machine learning approaches, along with the MARS data mining model, closely correspond to the recorded LST values of the study area.

The authors performed an extensive evaluation of supervised machine learning, Table 5 delineates the quantity of basis functions employed in the MARS model formulation, unsupervised machine learning, and MARS methodologies to identify the optimal LST prediction model for the research region utilizing the whole dataset, as outlined in table 6. The statistical analysis in table 6 indicated that the suggested ANNs and MARS machine learning algorithms attained commendable outcomes with high accuracy, whereas GRANN produced good results but with little precision. The finding is corroborated by the statistical evaluations of *ME*, *RMAE*, *MSE*, *RMSE*, and *SD* displayed in table 6. Figures 3 to 18 provide line graph representations of the anticipated data for each model. whereas table 6 encapsulates the outcomes of both the ANNs and MARS models. Figure 19 illustrates the box plots for the implemented ANNs and MARS methodologies.

4.2 Discussion

4.2.1 Development of the Supervised Artificial Neural Networks Models

A single-layer Backpropagation Artificial Neural Network (BPANN) model was trained with Tansig and Purelin activation functions for the hidden and output layers, respectively. The ideal model architecture, contingent upon the quantity of hidden neurons, was established using a systematic trial-and-error methodology utilizing assessment measures including *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*. The quantity of concealed neurons ranged from 1 to 50, and the network underwent training for 5000 epochs with a learning rate of 0.03, a minimum performance gradient of 0.0000001, a target of 0, a maximum of 6 validation failures, and a momentum coefficient of 0.9.

The optimal configurations of the BPANN models were attained through iterative training to predict the LST of the study area, utilizing thirteen distinct BPANN machine learning algorithms: BRBPANN, LMBPANN, BFGSQNBPANN, CGPBRBPANN, PRCGBPANN, FRCGBPANN, GDBPANN, GDMBPANN, GDALRBPANN, SCGBPANN, GDMALRBPANN, OSSBPANN, and RABPANN, with their corresponding MATLAB syntax as trainbr, trainlm, trainbfg, traincgb, traincgp, traincgf, traingd, traingdm, traingda, trainscg, traingdx, trainoss, and trainrp, respectively.

The optimal model configurations for predicting the LST achieved by BRBPANN, LMBPANN, BFGSQNBPANN, CGPBRBPANN, PRCGBPANN, FRCGBPANN, GDBPANN, GDMBPANN, GDALRBPANN, SCGBPANN, GDMALRBPANN, OSSBPANN, and RABPANN were determined to be [4-7-1], [4-8-1], [4-2-1], [4-2-1], [4-7-1], [4-3-1], [4-12-1], [4-1-1], [4-1-1], [4-2-1], [4-2-1], [4-2-1], and [4-1-1], respectively. These structures have four input variables (Rainfall, Relative Humidity, Temperature, Wind Speed), the maximum number of neurons employed to attain optimal results, and one output variable (forecasted temperature). These architectures produced the lowest values for statistical analysis metrics (*ME*, *RMAE*, *MSE*, *RMSE*, and *SD*).

Table 2 presents the summary outcomes of the training and testing conducted with all 13 supervised ANN machine learning algorithms. The statistical results in table 2 indicate that supervised machine learning approaches yield good outcomes with enhanced accuracy in predicting LST in the research region. Furthermore, in forecasting Land Surface Temperature (LST) for the study area, the OSSBPANN and GDALRBPANN models demonstrated efficiency, achieving optimal results in 86 and 110 iterations within 0 and 1 second, respectively. In contrast, the GDBPANN and GDMBPANN models necessitated significantly more time, requiring 5000 iterations in 5 seconds each to reach optimal model outcomes.

The disparities in residuals across the models are promising, and the mean error (*ME*), root mean absolute error (*RMAE*), mean square error (*MSE*), root mean squared error (*RMSE*), and standard deviation (*SD*) for both training and testing yield favourable results. BRBPANN and LMBPANN have shown superiority in model assessment and validation metrics for forecasting land surface temperature in the research region. Supervised artificial neural networks are effective and efficient models for LST predictions, providing enhanced accuracy. Figures 3 to 15 illustrate the line graph representation of the supervised machine learning techniques for the thirteen distinct models utilized in this study.

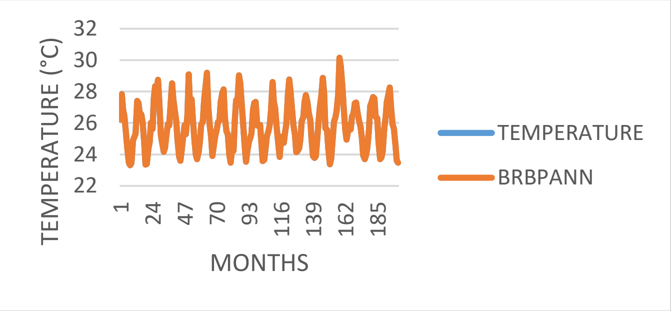

Figure 3. Line Graph Visualization of the BRBPANN model

Table 2. Model results for Supervised Machine Learning Predictions (Units in °C)

TRAINING							
PCI	**ME**	**RMAE**	**MSE**	**RMSE**	**SD**	**EPOCH**	**TIME**
BRBPANN	4.5444E-08	0.0002	4.9328E-12	2.2210E-06	1.1800E-08	876	10sec
LMBPANN	1.6111E-07	0.0004	4.8998E-11	6.9996E-06	3.5000E-08	353	3sec
BFGSQNBPANN	-4.8686E-05	0.0070	2.1156E-06	0.0015	7.2900E-06	394	1sec
CGPBRBPANN	-0.00073	0.0271	0.0002	0.0123	2.8500E-05	121	0sec
PRCGBPANN	-0.0005	0.0229	0.0009	0.0306	9.4600E-05	125	0sec
FRCGBPANN	-0.0010	0.0311	0.0008	0.0291	3.0200E-05	127	0sec
GDBPANN	-0.0037	0.0609	0.0171	0.1307	5.3000E-05	5000	5sec
GDMBPANN	-0.0071	0.0844	0.0124	0.1115	2.6500E-04	5000	5sec
GDALRBPANN	0.0277	0.1663	0.0271	0.1647	6.2200E-04	110	1sec
SCGBPANN	0.0043	0.0654	0.0009	0.0294	6.3200E-05	132	0sec
GDMALRBPANN	0.0034	0.0580	0.0093	0.0966	1.9200E-04	383	0sec
OSSBPANN	0.0016	0.0402	0.0058	0.0761	1.7500E-04	86	0sec
RABPANN	-0.0025	0.0499	0.0020	0.0447	4.7400E-06	800	1sec
TESTING							
PCI	**ME**	**RMAE**	**MSE**	**RMSE**	**SD**	**EPOCH**	**TIME**
BRBPANN	3.3742E-07	0.0006	4.1212E-12	2.0301E-06	6.55E-09	876	10sec
LMBPANN	-2.6702E-07	0.0005	2.7129E-11	5.2085E-06	1.9300E-08	353	3sec
BFGSQNBPANN	-5.7509E-06	0.0024	9.6787E-07	0.0010	6.1400E-06	394	1sec
CGPBRBPANN	0.0027	0.0520	0.0002	0.0131	1.1500E-05	121	0sec
PRCGBPANN	0.0043	0.0655	0.0007	0.0257	8.1400E-05	125	0sec
FRCGBPANN	0.0101	0.1005	0.0007	0.0265	6.1400E-05	127	0sec
GDBPANN	-0.0115	0.1072	0.0242	0.1555	1.2500E-04	5000	5sec
GDMBPANN	-0.0026	0.0507	0.0061	0.0780	6.9800E-04	5000	5sec
GDALRBPANN	0.0215	0.1467	0.0180	0.1343	1.0900E-03	110	1sec
SCGBPANN	0.0091	0.0953	0.0005	0.0226	9.4300E-05	132	0sec
GDMALRBPANN	0.0127	0.1129	0.0047	0.0688	5.0900E-04	383	0sec
OSSBPANN	0.0035	0.0592	0.0027	0.0516	3.6300E-04	86	0sec
RABPANN	-0.0007	0.0260	0.0008	0.0288	1.8400E-04	800	1sec

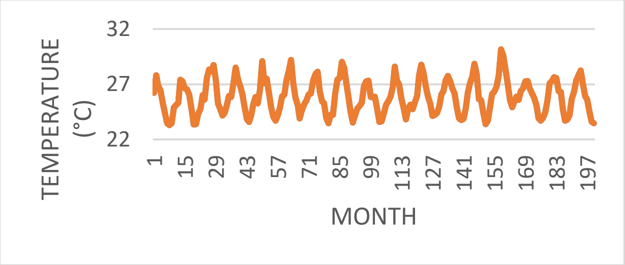

Figure 4. Line Graph Visualization of the LMBPANN model

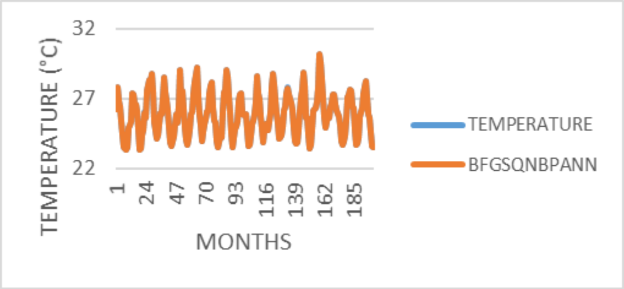

Figure 5. Line Graph Visualization of the BFGSQNBPANN model

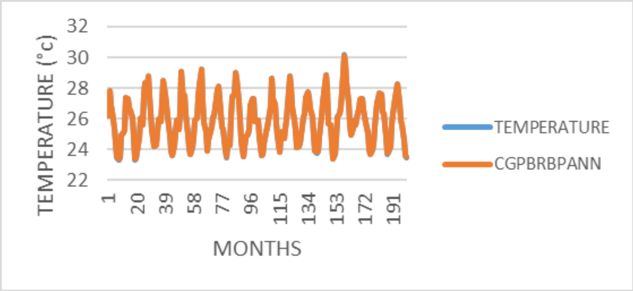

Figure 6. Line Graph Visualization of the CGPBRBPANN model

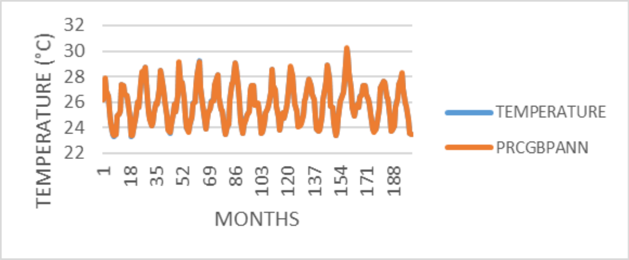

Figure 7. Line Graph Visualization of the PRCGBPANN model

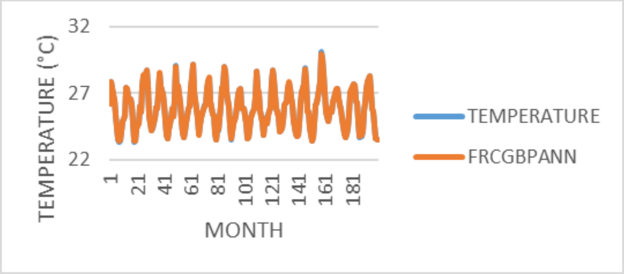

Figure 8. Line Graph Visualization of the FRCGBPANN model

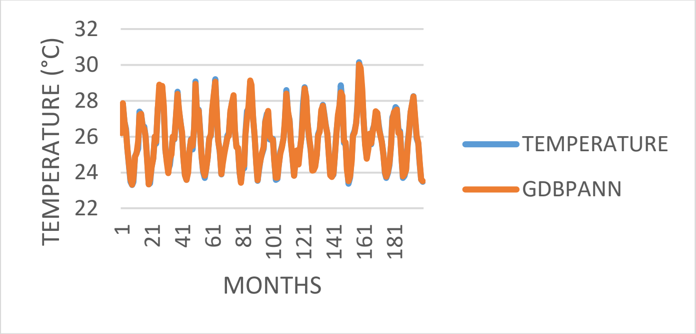

Figure 9. Line Graph Visualization of the GDBPANN

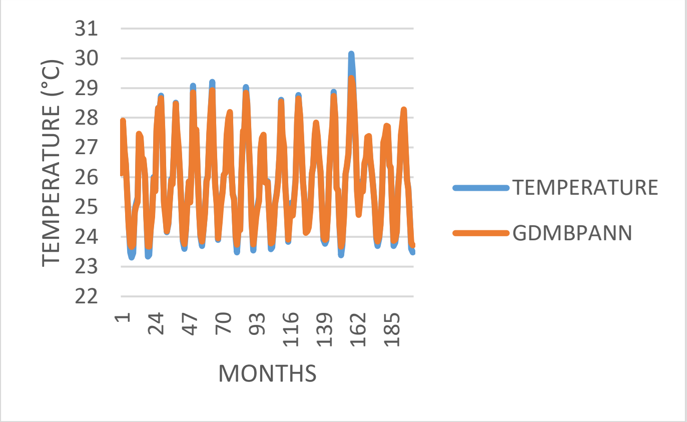

Figure 10. Line Graph Visualization of the GDMBPANN model

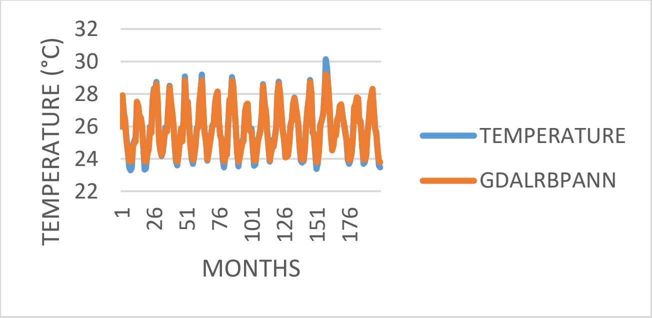

Figure 11. Line Graph Visualization of the GDALRBPANN model

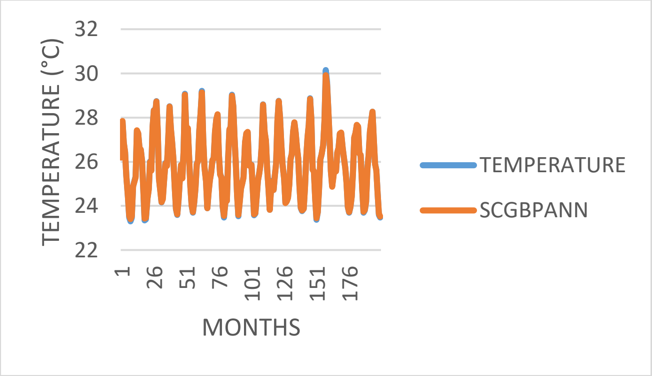

Figure 12. Line Graph Visualization of the SCGBPANN model

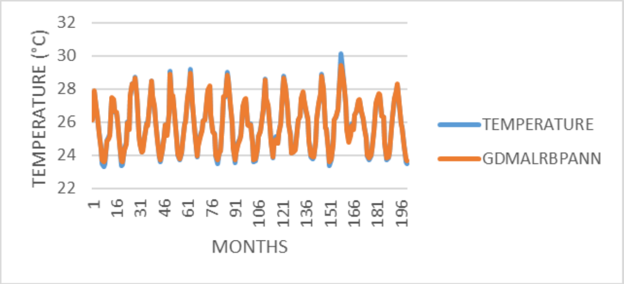

Figure 13. Line Graph Visualization of the GDMALRBPANN model

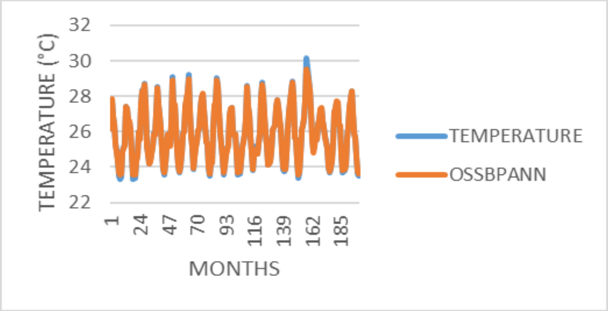

Figure 14. Line Graph Visualization of the OSSBPANN model

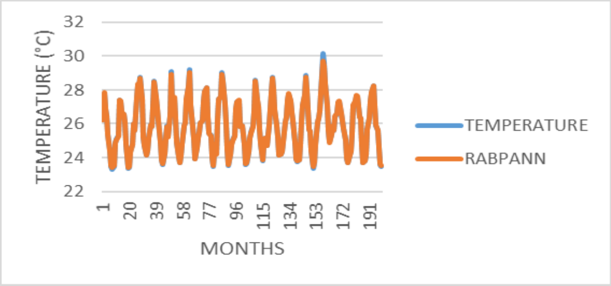

Figure 15. Line Graph Visualization of the RABPANN model

Table 3. Model results for Unsupervised Machine Learning Predictions (Units in °C)

TRAINING							
PCI	**ME**	**RMAE**	**MSE**	**RMSE**	**SD**	**EPOCH**	**TIME**
RBFANN	8.7554E-10	2.9600E-05	8.4187E-12	2.9015E-06	1.1500E-09	50	12.4sec
GRANN	0.0007	0.0271	0.0233	0.1527	9.5300E-04	50	12.5sec
TESTING							
PCI	**ME**	**RMAE**	**MSE**	**RMSE**	**SD**	**EPOCH**	**TIME**
RBFANN	2.0809E-7	0.0005	2.2985E-11	4.7943E-06	5.1900E-09	50	12.4sec
GRANN	0.1654	0.4067	0.8376	0.9152	4.4000E-03	50	12.5sec

4.2.2 Development of the Unsupervised Artificial Neural Networks Models

The optimum model architectures attained by the two unsupervised machine learning algorithms (RBFANN and GRANN) for predicting the land surface temperature (LST) of the region, with the minimal evaluation metrics, were identified as [4 50 1 1] and [4 50 0.1 1], respectively. This indicates the presence of 4 input variables (independent datasets), a maximum of 50 hidden neurons, a spread constant (ranging from 0 to 1), and 1 output variable (dependent dataset). The ideal spread parameter was established by adjusting it from 0 to 1 after each recurrent training session until the most favourable results were attained, as shown by statistical measures including *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*.

Table 3 presents the summary outcomes of the training and testing of the anticipated LST attained by the two strategies. Figures 16 and 17 illustrate the line graph representations generated by the two models. According to the statistical data presented in table 3, both unsupervised approaches attained commendable results in accurately forecasting the LST for the research region, demonstrating significantly enhanced accuracy and precision. Both models executed 50 iterations in under 12 seconds to get optimal outcomes. The *ME*, *RMAE*, *MSE*, *RMSE*, and *SD* for both training and testing are satisfactory and promising. The RBFANN outperformed the GRANN model in estimating the LST based on the statistical metrics obtained in this study.

Nonetheless, both unsupervised machine learning systems demonstrated efficacy and realism as tools for LST forecasting in the region, with significantly enhanced accuracy.

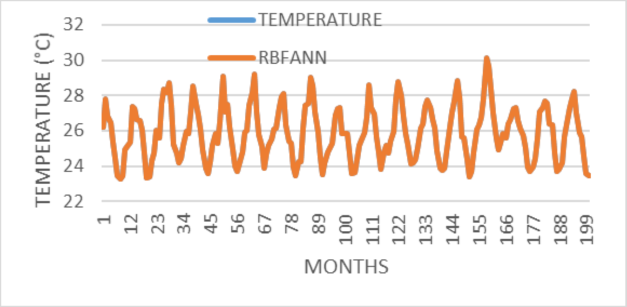

Figure 16. Line Graph Visualization of the RBFANN model

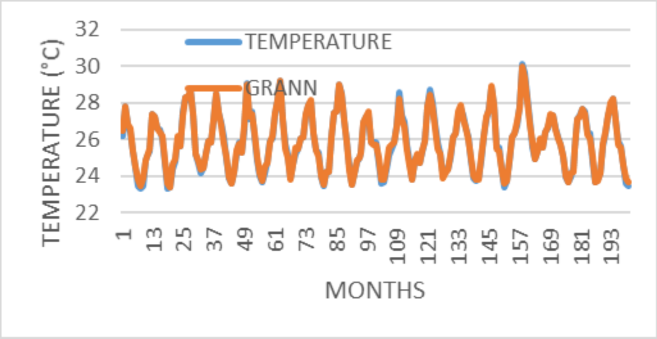

Figure 17. Line Graph Visualization of the GRANN model

4.2.3 Development of the Multivariate Adaptive Regression Splines (MARS) Technique Model

In the MARS model formulations for LST forecasting, 21 basis functions were included in the training and testing phases to predict the monthly historical time series LST data. In the final model formulation, two basis functions were maintained, signifying that nineteen basis functions were eliminated throughout the backward training phase owing to overfitting. The training and testing outcomes are presented in table 4. Table 5 illustrates the quantity of basis functions included in the final model formulation, whereas figure 18 depicts the line graph visualization of the MARS model. Equation (45) is the optimum model utilized for predicting the monthly LST via the MARS model.

The Salford Predictive Modeler Software (SPM V. 8.0) was utilized to develop the MARS model for predicting LST. The independent variables were Rainfall, Relative Humidity, Temperature, and Wind Speed, whilst the dependent variable was Temperature data. The SPM program was selected following the endorsement of Yakubu et al. (2018), who observed that it is engineered for exceptional accuracy, rapid performance, and efficacy in developing predictive, descriptive, and analytical models from datasets of any magnitude. The program is a robust, user-friendly mathematical instrument that corresponds with the authors' choices for research inquiries relative to other prominent data mining methods.

$$y(i) = 25.11 + 1 \times BF1 - 1 \times BF2 \qquad (45)$$

4.2.4 Comparing the predictive performance results of the ANNs Machine Learning and MARS models

The predictive performance ranking of all employed approaches (Supervised, Unsupervised, and MARS) was assessed utilizing their model evaluation and validation statistical metrics: *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*. Table 6 presents the assessment outcomes for all employed procedures. According to table 6, the model exhibiting the most favourable statistical indicators was determined to be the most effective for fitting the observed land surface temperature of the research area. Table 6 demonstrates that the MARS model produced the most favourable statistical values for *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*, with respective values of 1.8705E-07 °C, 0.0004 °C, 3.3449E-13 °C, 5.37835E-07 °C, and 1.6000E-09 °C. The box plot of all models illustrated in figure 19 and the model results presented in table 6 further corroborate these findings. The statistical indicators illustrate the prediction accuracy of the MARS model by presenting the metrics of the discrepancies between the observed and predicted LST values as residuals.

Table 6 further illustrates the predictive capabilities of all the models utilized in this study. MARS attained superior model validation and assessment metrics relative to the alternatives. The GRANN model had the poorest performance for *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*, with values of 0.0498 °C, 0.2233 °C, 0.2662 °C, 0.5159 °C, and 7.7600 °C, respectively. This study concludes that the MARS model surpasses the ANN models in accurately anticipating monthly LST data. The enhanced efficacy of MARS in this study relative to ANN models may be ascribed to its reduced training duration and its greater capacity to address non-linearity and non-parametric challenges in the input datasets.

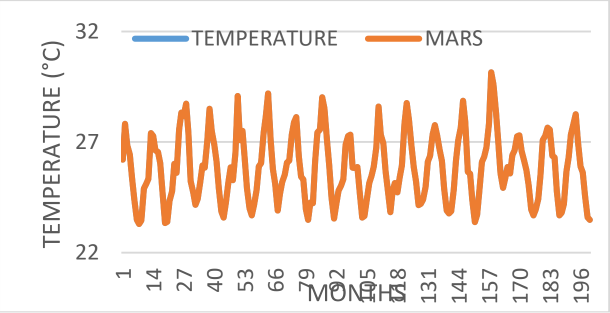

Figure 18. Line Graph Visualization of the MARS model

Table 4. Model results MARS models Predictions (Units in °C)

TRAINING					
PCI	**ME**	**RMAE**	**MSE**	**RMSE**	**SD**
MARS	1.9908E-07	0.0004	3.4304E-13	5.8570E-07	2.2500E-09
TESTING					
PCI	**ME**	**RMAE**	**MSE**	**RMSE**	**SD**
MARS	1.5876E-07	0.0004	3.1436E-13	5.6068E-07	1.0800E-09

Table 5. Basis Functions Equations for the MARS Model

Basis Functions	**Equations**
$BF1$	$\max(0, T_i - 25.11)\,;$
$BF2$	$\max(0, 25.11 - T_i)\,;$

Table 6. Statistical Analysis of all the models (Units in °C)

PCI	ME	RMAE	MSE	RMSE	SD
BRBPANN	1.3252E-07	0.0004	4.6908E-12	2.1658E-06	8.4300E-09
LMBPANN	3.3423E-08	0.0002	4.2473E-11	6.5172E-06	2.4800E-08
BFGSQNBPANN	-3.5881E-05	0.0060	1.7733E-06	0.0013	5.0800E-06
CGPBRBPANN	0.0003	0.0171	0.0002	0.0125	1.7800E-05
PRCGBPANN	0.0009	0.0302	0.0009	0.0292	6.3200E-05
FRCGBPANN	0.0023	0.0483	0.0008	0.0283	2.8400E-05
GDBPANN	-0.0060	0.0777	0.0192	0.1385	4.2200E-05
GDMBPANN	-0.0058	0.0759	0.0105	0.1027	1.8300E-04
GDALRBPANN	0.0258	0.1607	0.0244	0.1560	4.4000E-04
SCGBPANN	0.0057	0.0756	0.0008	0.0276	4.1200E-05
GDMALRBPANN	0.0062	0.0785	0.0080	0.0892	1.2900E-04
OSSBPANN	0.0022	0.0467	0.0049	0.0697	1.2200E-04
RABPANN	-0.0020	0.0442	0.0016	0.0406	2.2300E-06
RBFANN	6.2675E-08	0.0003	1.2763E-11	3.5726E-06	9.4400E-10
GRANN	0.0498	0.2233	0.2662	0.5159	7.7600E-04
MARS	1.8705E-07	0.0004	3.3449E-13	5.7835E-07	1.6000E-09

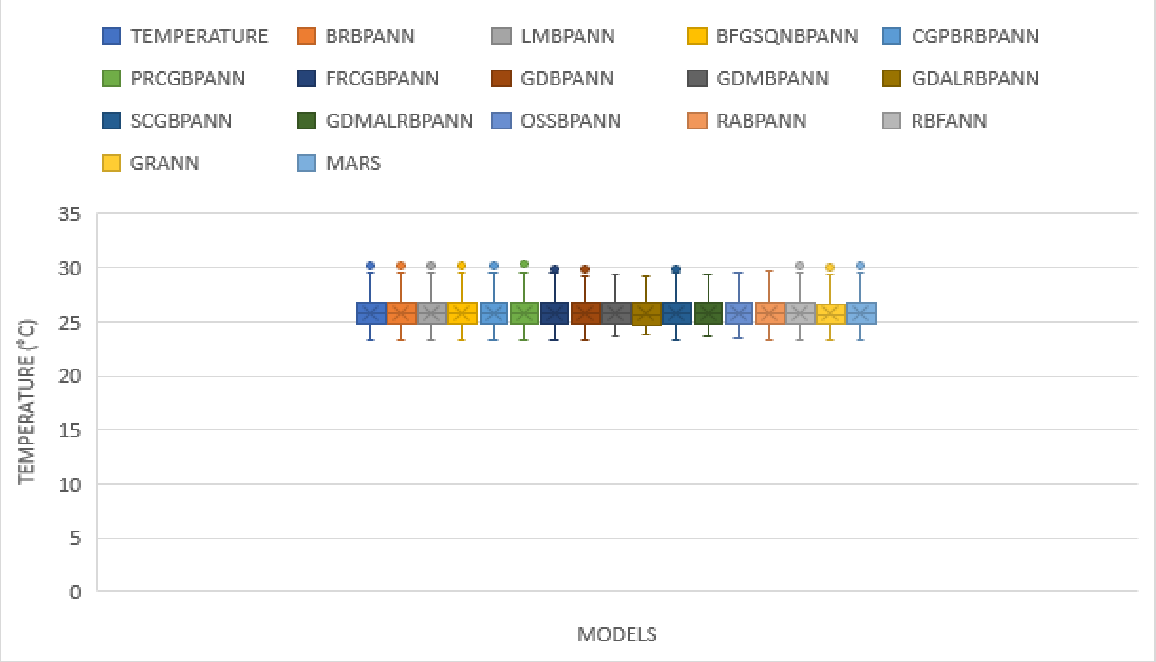

Figure 19. Box plot Visualization of all the models

The predictive performance ranking of all utilized methodologies (Supervised, Unsupervised, and MARS) was evaluated using their model assessment and validation statistical metrics: *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*. Table 6 delineates the evaluation results for all used methodologies. Table 6 indicates that the model with the most advantageous statistical metrics was identified as the most successful for modelling the observed land surface temperature of the study region. Table 6 indicates that the MARS model yielded the most advantageous statistical values for *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*, with corresponding values of 1.8705E-07 °C, 0.0004 °C, 3.3449E-13 °C, 5.37835E-07 °C, and 1.6000E-09 °C. The box plot of all models depicted in figure 19 and the model results shown in table 6 further substantiate these conclusions. The statistical indicators demonstrate the predictive accuracy of the MARS model by showcasing the metrics of the disparities between the observed and anticipated LST values as residuals.

Table 6 further demonstrates the predictive efficacy of all models included in this investigation. MARS achieved improved model validation and evaluation metrics compared to the alternatives. The GRANN model had the least favourable performance for *ME*, *RMAE*, *MSE*, *RMSE*, and *SD*, yielding values of 0.0498 °C, 0.2233 °C, 0.2662 °C, 0.5159 °C, and 7.7600 °C, respectively. This study reveals that the MARS model outperforms the ANN models in reliably predicting monthly LST data. The superior effectiveness of MARS in this study compared to ANN models may be attributed to its shorter training time and its better ability to tackle non-linearity and non-parametric issues in the input datasets.

5 Conclusions and Recommendations

Weather characteristics, including temperature forecasting, pose significant challenges for energy producers, agriculture, climatologists, aircraft operators, marine operators, and researchers. This problem has stimulated countless investigations in both past and recent decades. Existing research indicates that machine learning (ML) approaches frequently surpass classical methods owing to their capacity to manage data non-linearity, intermittence, and high variability challenges that older techniques inadequately handle, resulting in subpar performance. The recent emergence of Artificial Intelligence (AI)-driven machine learning methodologies has demonstrated efficacy in addressing non-linearity issues inherent in conventional forecasting models. This research utilized multivariate time series data over 37 years, including Rainfall, Temperature, Relative Humidity, and Wind Speed from January 1985 to December 2022. The research employed a data mining methodology, namely the Multivariate Adaptive Regression Splines (MARS), in conjunction with fifteen Artificial Neural Networks (ANNs) machine learning models. Levenberg-Marquardt, Bayesian Regularization, Broyden-Fletcher-Goldfarb-Shanno Quasi-Newton method, Conjugate Gradient with Powell/Beale Restarts. Polak-Ribiere Conjugate Gradient Method, Fletcher-Reeves Algorithm Conjugate Gradient, Gradient Descent, Momentum-Enhanced Gradient Descent, Adaptive Learning Rate Gradient Descent, Scaled Conjugate Gradient, Momentum and Adaptive Learning Rate Gradient Descent, One-Step Secant Method, Resilient Backpropagation Algorithm, Radial Basis Functions, and Generalized Regression. These models were utilized to forecast historical monthly land surface temperature data for the Aowin District. Selected sub-series of the variables (Rainfall, Relative Humidity, Temperature, Wind Speed) served as inputs for the ANNs and MARS models, while the original time series of the variable (Temperature) functioned as the output. Models were trained on 70% of the dataset, while the remaining 30% was allocated for validation and assessment. Evaluations of the statistical model encompassed mean error (*ME*), root mean absolute error (*RMAE*), mean squared error (*MSE*), root mean squared error (*RMSE*), and standard deviation (*SD*). The statistical analysis revealed that the MARS model outperformed the ANN models. The MARS model attained superior performance metrics, with *ME*, *RMAE*, *MSE*, *RMSE*, and *SD* values of 1.8705E-07 °C, 0.0004 °C, 3.3449E-13 °C, 5.37835E-04 °C, and 1.6000E-09 °C, respectively. The findings indicate that the MARS model had superior effectiveness and reduced LST predicting error relative to the ANN models. The research indicates that the MARS model is superior for predicting monthly LST with enhanced accuracy for the study area. The performance of ANNs was comparatively inferior to that of the MARS model, likely because of the variability in the nonlinear dynamics of the independent variables, which the MARS-based models represented more adeptly.

The effective modelling of meteorological factors (Rainfall, Relative Humidity, Temperature, Wind Speed) with the MARS model, aided by basis functions, probably enhanced the representation of LST dynamics. The study utilized satellite database archives, and subsequent research integrating more data, including wind direction, sun radiation, and other factors, might reinforce these results. This study reveals that MARS and ANN models are proficient in forecasting monthly LST across diverse locations. The authors focused on evaluating ANN-based machine learning and the MARS data mining methodology to illustrate their enhanced efficacy in LST forecasting. Future research may incorporate more climate change factors and examine various ML-based methodologies to determine their applicability across different regions of Ghana. Utilizing hybrid models that integrate neural networks with statistical models may improve the capacity to predict intricate non-linear patterns in LST. Future research may investigate the predictive accuracies of various ANN models, including Genetic Expression Programming (GEP), Gradient Boosting Tree, Random Forest, Statistical Methods, Numerical Weather Prediction (NWP), Cascade Forward Backpropagation Network, Layer Recurrent Network, Focused Time Delay Neural Network, Decision Tree, CART, K-Mean, Elman Neural Network, Wavelet, M5Tree model, Self-Organizing Map (SOM), Genetic Artificial Neural Networks, Convolutional Recurrent Neural Network (CRNN), Support Vector Machine (SVM), and Least Squares Support Vector Machine (LSSVM), alongside diverse regression and deep learning methodologies. A comparative evaluation of various models can yield a more thorough grasp of LST prediction for the research area. .

Acknowledgment
The authors express gratitude to the anonymous reviewer for their helpful feedback, time, and efforts in enhancing this article.

Conflict of Interest
The authors assert that there are no potential conflicts of interest associated with this study. The work does not violate any copyright, property right, or any other rights of third parties, and the Authors are the exclusive proprietors of the work.

References

Adeoti O. A. and P. A. Osanaiye (2013), “Effect of Training Algorithms on The Performance of ANN for Pattern Recognition of Bivariate Process”, International Journal of Computer Applications, Vol. 69, No. 20, pp. 8 – 12.

Aowin Suaman District (2006). Aowin Suaman District Assembly: http://aowinsuaman.ghanadistricts.gov.gh
Asafo-Adjei E. and E. Buabeng (2016). An Evaluation of Challenges Facing Smallholders in Ghana: A Case Study for the Aowin Suaman District Journal of Biology, Agriculture and Healthcare, Vol.6, No.3, 22-29.

Asenso-Gyambibi D., J. A. Danquah, E. K. Larbi, M. S. Peprah and N. L. Quaye-Ballard (2024). Enhancing Survey Field Data with Artificial Intelligence: A Real-Time Kinematic GPS Study. Arabian Journal of Geosciences, 17(184), 1-12, https://doi.org/10.1007/s12517-024-11989-2.

Azad H. B., S. Mekhilef and V. Gounder (2014). Long-Term Wind Speed Forecasting and General Pattern

Recognition using Neural Networks. IEEE Transactions on Sustainable Energy, 5(2), 546-553.

Azari B., K. Hassan, J. Pierece and S. Ebrahimi (2022). Evaluation of Machine Learning Methods Application in Temperature Prediction. Computational Research Progress in Applied Science & Engineering (CRPASE): Transactions of Civil and Environmental Engineering, 8(1), 1-12.

Baghirli O (2015) Comparison of Lavenberg-Marquardt, Scaled Conjugate Gradient and Bayesian Regularization Backpropagation Algorithms for Multistep Ahead Wind Speed Forecasting Using Multilayer Perceptron Feedforward Neural Network. Published MSc Thesis Report, Uppsala University, Gotland, 35 pp.

Beale MH, M. T. Hagan and H. B. Demuth (2017) Neural Network Toolbox™ User's Guide. The MathWorks Inc, USA, 431 pp.

Beale E. M. L. (1972) A Derivation of Conjugate Gradients. In: Lootsma FA (ed) Numerical methods for nonlinear optimization. Academic Press, London, pp 39–43.

Biglari F and A. Ebadian (2015) Limited Memory BFGS Method Based on a High-Order Tensor Model. Comput Optim Appl 60(2):413–422.

Braspenning P. J., F. Thuijsman and A. J. M. M. Weijters (1995) Artificial Neural Networks: An Introduction to ANN Theory and Practice. Springer Sci Bus Media, 293 pp.

Cadenas E. and W. Rivera (2009). Short Term Wind Speed Forecasting in La Venta, Oaxaca, Mexico, using Artificial Neural Networks. Renewal Energy, 34, 274-278.

Cassie D., A. St-Hilaire and N. El-Jabi (2004). Prediction of Water Temperatures using Regression and Stochastic Models, 57th Canadian Water Resources Association Annual Congress, Water and Climate Change: Knowledge for better Adaptation, June 16-18 2004, Montreal, Ql, Canada, 1-6.

Cheng M. Y. and M. T. Cao (2014). Accurately Predicting Building Energy Performance using Multivariate Adaptive Regression Splines. Applied Soft Computing, 22, 178-188, http://dx.doi.org/10.1016/j.asoc.2014.05.015.

Diaz J., F .J. Fernandez and M. M. Prieto (2020). Hot Metal Temperature Forecasting at Steel Plant using Multivariate Adaptive Regression Splines. Metals, 10(41), 1-16, https://doi.org/10.3390/met10010041.

Dominiak S. and P. Terray (2005). Improvement of ENSO Prediction using a Linear Regression Model with a Southern Indian Ocean Sea Surface Temperature Predictor, Geophysical Research Letters, 32(L18702, 1-4.

Dudek G. (2011). Generalized Regression Neural Network for Forecasting Time Series with Multiple Seasonal Cycles. Springer-Verlag Berlin Heidelberg 1, 1-8.

Ehiakpor D. S., G. Danso-Abbeam and J. E. Baah (2016). Cocoa farmer’s perception on climate variability and its effects on adaptation strategies in the Suaman district of western region, Ghana, Cogent Food & Agriculture, 2:1, 1210557, DOI:10.1080/23311932.2016.1210557

Erdogan S. (2009). A Comparison of Interpolation Methods for Producing Digital Elevation Models at the Field Scale. Earth Surface Processes and Landforms. 34, 366-376.

Fang T. and R. Lahdelma (2016). Evaluation of a Multiple Linear Regression Model and SARIMA Model in Forecasting Heat Demand for District Heating System, Applied Energy, 179, 544-552.

Fletcher R and C. M. Reeves (1964) Function Minimization by Conjugate Gradient. Comput J 7(2):149–154.

Foresee F. D. and M. T. Hagan (1997) Gauss-Newton approximation to Bayesian learning. In: Proc Int Jt Conf Neural Netw 1997, Vol 3, pp 1930–1935.

Friedman J. H. (1991). Multivariate adaptive regression splines”, Annals Statistics, 19, 1-67.

Garcez A. S. A., L. C. Lamb and D. M. Gabbay (2008) Neural-Symbolic Cognitive Reasoning. Springer Sci Bus Media, 198 pp.

Ghana Districts (2017). A repository of all Local Assemblies in Ghana Ghana Districts: A repository of all Local Assemblies in Ghana Accessed: August 25, 2024

Ghana Statistical Service (2010). Population and housing census estimates. Stats Ghana, Accra Ghana. www.ghana statistical service.com

Ghorbani M. A., R. Khatibi, M. H. Fazelifard, L. Naghipour and O. Makarynskyy (2015). Short-Term Wind Speed Predictions with Machine Learning Techniques. Meteoro I Atmos Phys, 1(16), https://doi.org/10.1007/s00703-015-0398-9

Gupta I., H. Mittal, D. Rikhari and A. K. Singh (2022). MLRM: A Multiple Linear Regression Based Model for Average Temperature Prediction of a Day, A Preprint, 1-8, arXiv:2203.05835vI.

Hagan M.T., H. B. Demuth and M. H. Beale (1996) Neural Network Design. PWS Publishing, Boston, MA.

Hannan S.A., R. R. Manza and R. J. Ramteke (2010) Generalized Regression Neural Network and Radial Basis Function for Heart Disease Diagnosis. Int J Comput Appl 7(13):7–13.

Hornik K., M. Stinchcombe and H. White (1989) Multilayer feed forward networks are universal approximators. Neural Netw 2(5):359–366.

Huang F.M., P. Wu and Y. Y. Ziggah (2016) GPS Monitoring Landslide Deformation Signal Processing Using Time Series Model. Int J Signal Process Image Process Pattern Recognit 9(3):321–332.

Ibrahim M. A. H., M. Mamat and W. J. Leong (2014), "The Hybrid BFGS-CG Method in Solving Unconstrained Optimization Problems", Abstract and Applied Analysis, Vol. 2014, pp. 1 – 6.

Idri A., A. Zakrani and A. Zahi (2010) Design of Radial Basis Function Neural Networks for Software Effort Estimation. Int J Comput Sci Issue 7(4):11–17.

Ilona (2024). Ancient Origins: The History and Culture of Aowin in Western North Region, Ghana, https://mrcsl.org/aowin/ Accessed: August 26, 2024

Kaloop M. R., M. Rabah, J. W. Hu and A. Zaki (2017) Using Advanced Soft Computing Techniques for Regional Shoreline Geoid Model Estimation and Evaluation. Mar Georesour Geotechnol 36(6):1–11.

Kaur H. and D. S. Salaria (2013) Bayesian Regularization Based Neural Network Tool for Software Effort Estimation. Glob J Comput Sci Technol 13(2):44–50.

Kişi Ö. and E. Uncuoğlu (2005) Comparison of three back-propagation training algorithm for two case studies. Indian J Eng Mater Sci 12:434–442.

Kumi-Boateng B. and M. S. Peprah (2020) Modeling Local Geometric Geoid using Soft Computing and Classical Techniques: A Case Study of the University of Mines and Technology (UMaT) Local Geodetic Reference Network. Int J Earth Sci Knowl Appl 2(3):166–177.

Kuter S., G. W. Weber, Z. Akyurek and A. Ozmen (2015). Inversion of top of Atmospheric Reflectance Values by Conic Multivariate Adaptive Regression Splines. Inverse Problems in Science and Engineering, 23(4), 651-669.

Lerch S. and T. L. Thorarinsdottir (2013). Comparison on non-homogeneous Regression Models for Probabilistic Wind Speed Forecasting. Tellus A: Dynamic Meteorology and Oceanography, 65(1), 21206, 1-14, https://doi.org/10.3402/tellusa.v65i0.21206

Moller M. F. (1993). A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks 6(4), 525-533.

Mohandes M. A., S. Rehman and T. O. Halawani (1998). A Neural Networks Approach for Wind Speed Prediction. Renewal Energy, 13(3), 345-354.

Moini M. and A. Lakizadeh (2011), Concrete Workability: An Investigation on Temperature Effects Using Artificial Neural Networks, Author House. 148 pp.

Mueller V. A. and F. H. Hemond (2013). Extended artificial neural networks: in-corporation of a priori chemical knowledge enables use of ion selective electrodes for in-situ measurement of ions at environmental relevant levels. Talenta. 117: 112-118.

Mukkamala S., A. H. Sung and A. Abraham (2003), "Intrusion Detection Using Ensemble of Soft Computing Paradigms", In Intelligent Systems Design and Applications, Springer, Berlin, Heidelberg, pp. 239-248.

Peprah M. S. and E. K. Larbi (2021). Lake Water Level Prediction Model Based on Artificial Intelligence and Classical Techniques – An Empirical Study on Lake Volta Basin, Ghana. International Journal of Earth Sciences Knowledge and Applications, 3(2), 134-150.

Peteiro-Barral D. and B. Guijarro-Berdiñas, B. (2013), "A Study on the Scalability of Artificial Neural Networks Training Algorithms Using Multiple-Criteria Decision-Making Methods", Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Springer, Berlin, Heidelberg, pp. 162-173.

Polak E. R. and G. Ribière, G. (1969), "Note Sur la Convergence de Methodes de Directions Conjugat", Revue Francaise d'Informatique et Recherche Operationnelle, Vol. 16, pp. 35 – 43.

Powell M. J. D. (1977), "Restart Procedures for the Conjugate Gradient Method", Mathematical Programming, Vol. 12, No. 1, pp.241-254.

Prasad N., R. Singh and S. P. Lal (2013), "Comparison of Back Propagation and Resilient Propagation Algorithm for Spam Classification", Proceedings of the 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation, pp. 29- 34.

Radhika Y. and M. Shashi (2009). Atmospheric Temperature Prediction using Support Vector Machines. International Journal of Computer Theory and Engineering, 1(1), 1793-8201.

Raj N. and Z. Gharineiat (2021). Evaluation of Multivariate Adaptive Regression Splines and Artificial Neural Network for Prediction of Mean Sea Level Trend Around Northern Australian Coastlines. Mathematics, 9(2696), 1-20, https://doi.org/10.3390/math9212696.

Riedmiller M. and H. Braun (1992), "RPROP. A Fast Adaptive Learning Algorithm" Proceedings of the 1992 International Symposium on Computer and Information Sciences, Antalya, Turkey, pp.279-285.

Roy S. S., C. Pratyush and C. Barna (2018). Predicting Ozone Layer Concentration using Multivariate Adaptive Regression Splines, Random Forest and Classification and Regression Tree. Soft Computing Applications, Advances in Intelligent Systems and Computing, 634, https://doi.org/10.1007/978-3-319-62524-9_11

Sandhu P. S. and S. Chhabra (2011). A Comparative Analysis of Conjugate Gradient Algorithms and PSO Based Neural Network Approaches for Reusability Evaluation of Procedure Based Software Systems. Chiang Mai Journal of Science 38 (2), 123-135.

Santamaria-Bonfil, G., A. Reyes-Ballesteros and C. Gershenson (2016). Wind Speed Forecasting for Wind Farms: A Method Based on Support Vector Regression. Renewal Energy, 85, 790-809.

Soman S. S., H. Zareipour, O. Malik and P. Mandal (2010). A Review of Wind Power and Wind Speed Forecasting Methods with Different Time Horizons. IEEE Xplore, 1-8, https://doi.org/10.1109/NAPS.2010.5619586
Specht D. (1991). A General Regression Neural Network. IEEE Transactions on Neural Networks 2 (6), 568-576.

Srichandan S. (2012). A New Approach of Software Effort Estimation using Radial Basis Function Neural Networks. ISSN (Print). 1 (1): 2319-2526.

Sumi S. M., M. F. Zaman and H. Hirose (2012). A Rainfall Forecasting Method using Machine Learning Models and its Application to Fukuoka City Case. International Journal of Applied mathematics and Computer Science, 1-13.

Sreehari E. and G. S. Pradeep (2019). Climate Changes Prediction using Simple Linear Regression, Journal of Computational and Theoretical Nanoscience, 16, 1-4.

Szymanowski M., M. Kryza and W. Spallek (2013). Regression-baes Air Temperature Spatial Prediction Models: An Example from Poland. Meteorologische Zeitschrift, 22(5), 577-585.

Yakubu I., Y. Y. Ziggah and M. S. Peprah (2018). Adjustment of DGPS Data using Artificial Intelligence and Classical Least Squares Techniques. Journal of Geomatics, 12(1), 13-20.

Zarinkamar R. T. and R. V. Mayorga (2021). Outdoor Relative Humidity Prediction via Machine Learning Techniques. Journal of Environmental Informatics Letters, 6(2), 114-124.