Math Geosci (2016) 48:687–721
DOI 10.1007/s11004-016-9638-x

Capability of Artificial Neural Network for Forward Conversion of Geodetic Coordinates (ϕ, λ, h) to Cartesian Coordinates (X, Y, Z)

Yao Yevenyo Ziggah[1,2] · Hu Youjian[1] · Xianyu Yu[1] · Laari Prosper Basommi[3]

Received: 13 September 2015 / Accepted: 18 May 2016 / Published online: 15 June 2016

Abstract The standard forward transformation equation plays a major role in coordinate transformation between global and local datums. Thus, it is a prerequisite step in the forward conversion of geodetic coordinates into cartesian coordinates in coordinate transformation from global to local datum and vice versa. Numerous studies have been carried out on converting cartesian coordinates to geodetic coordinates (reverse procedure) through the application of iterative, approximate, closed form, vector-based and computational intelligence algorithms. However, based on literature covered pertaining to this study, it was realized that the existing researches do not fully address the issue of applying and testing alternative techniques in the case of the forward conversion. Hence, the purpose of this present study was to explore the coordinate conversion performance of two different artificial neural network approaches (backpropagation artificial neural network (BPANN) and radial basis function neural network (RBFNN)) and multiple linear regression (MLR). The statistical findings revealed that the BPANN, RBFNN and MLR offered satisfactory prediction of cartesian coordinates. However, the RBFNN compared to BPANN and MLR showed better stability and more accurate prediction results. Furthermore, in terms of maximum three-dimensional position error, the RBFNN attained 0.004 m while 0.011 and 0.627 m were achieved, respectively, by MLR and BPANN. By virtue of the success achieved in this study, the main conclusion drawn here is that RBFNN provides a promising alter-

✉ Hu Youjian
hyj_06@163.com; 2508938072@qq.com

1 Department of Surveying and Mapping, Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, People's Republic of China

2 Department of Geomatic Engineering, Faculty of Mineral Resource Technology, University of Mines and Technology, Tarkwa, Western Region, Ghana

3 Department of Environment and Resource Studies, University of Development Studies, Wa, Ghana

native in the forward conversion of geodetic coordinates into cartesian coordinates. Therefore, the capability of artificial neural network as a powerful tool for solving majority of function approximation problems in geodesy has been demonstrated.

Keywords Backpropagation artificial neural network · Radial basis function neural network · Multiple linear regression · Standard forward transformation equation · Coordinate conversion

1 Introduction

The recent rapid advancement in satellite positioning technologies such as global positioning system (GPS) in capturing locations of stationary and non-stationary objects on, above and beneath the Earth's surface has increased the possibility of obtaining coordinate positions with improved accuracy. These satellite positioning technologies provide vast amounts of spatio-temporal datums in either curvilinear geodetic coordinates (φ, λ, h) or cartesian coordinate (X, Y, Z) system. In the quest for solving most practical GPS navigation, geodetic, cartographical and astro-geodetic problems, it is important to convert geodetic coordinates into cartesian coordinates and vice versa (Civicioglu 2012; Ligas and Banasik 2011; Shu and Li 2010; Zhu 1994). The process of converting geodetic coordinates to cartesian coordinates is known as the forward conversion.

This forward conversion is an intermediate step in converting the GPS position measurement to the local coordinate system (Vanicek and Steeves 1996; Cai et al. 2011). This is because before the advent of GPS, local geodetic datums were established based on classical surveying methods like triangulation, trilateration, traverse, astronomical observation among others (Andrei 2006; Tierra et al. 2008). Hence, the local geodetic datum involved data in only geodetic coordinates without the existence of cartesian coordinates. This prevalent situation makes it impossible to utilize GPS coordinates based on the global datum of world geodetic system 1984 (WGS84) in a local geodetic system. The first step in applying GPS data locally requires the determination of transformation parameters. The most widely used methods in literature for such an application include the similarity models of Bursa–Wolf, Molodensky–Badekas, Veis model and three-dimensional Affine (Ge et al. 2013; Pan et al. 2015; Solomon 2013; Zeng 2014, 2015; Ziggah et al. 2013). However, the major point here is that, before these aforementioned similarity models could be applied, there is the need to convert all geodetic data of common points to cartesian coordinates. It must be noted here that without such conversion, the similarity models cannot be used in the transformation parameter determination for coordinate transformation between the global and local datums.

To accomplish this task, the standard forward transformation equation (Hoar 1982; Leick 2004; Schofield 2001; Sickle 2010) given in Eq. (1) is mainly used as the first step in the coordinate transformation procedure

$$\begin{aligned} X &= (N + h)\cos\varphi\ \cos\lambda \\ Y &= (N + h)\ \cos\varphi\ \sin\lambda, \\ Z &= [N\,(1 - e^2) + h\,]\ \sin\varphi, \end{aligned} \tag{1}$$

where φ, λ and h is the geodetic latitude, geodetic longitude and geodetic height while X, Y, Z is the cartesian coordinates. N in Eq. (1) is the radius of curvature in the prime vertical defined by Eq. (2) as

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}}. \tag{2}$$

Here, e is the first eccentricity expressed in Eq. (3) as

$$e = \frac{\sqrt{a^2 - b^2}}{a}, \tag{3}$$

where a and b are the semi-major axis and semi-minor axis of the geodetic ellipsoid.

Although Eq. (1) is mostly applied in geodetic practice, there exist other numerical methods such as artificial neural network (ANN) that can serve as an alternative approach. It is well acknowledged that the introduction of ANN applications has revolutionized the field of mathematical geodesy in terms of its attainable accuracy and in most cases, its dominance over the empirical methods. This assertion is well documented in literature. For example, ANN has been applied to solve most coordinate transformation problems between global and local datums (Gullu 2010; Gullu et al. 2011; Lin and Wang 2006; Mihalache 2012; Tierra et al. 2008, 2009; Tierra and Romero 2014; Turgut 2010; Yilmaz and Gullu 2012; Zaletnyik 2004), for GPS height conversion (Fu and Liu 2014; Liu et al. 2011; Lei and Qi 2010; Tieding et al. 2010; Wu et al. 2012a), in geodetic deformation modelling (Bao et al. 2011; Du et al. 2014a, b; Gao et al. 2014; Pantazis and Eleni-Georgia 2013; Yilmaz and Gullu 2014; Yilmaz 2013), earth orientation parameters determination (Liao et al. 2012; Schuh et al. 2002; Yu et al. 2015), precise orbital prediction (He-Sheng 2006; Li et al. 2014), gravity anomaly estimation (Hajian et al. 2011; Hamid and Mohammad 2013; Tierra and De Freitas 2005), geoid determination (Kavzoglu and Saka 2005; Pikridas et al. 2011; Stopar et al. 2006; Sorkhabi 2015; Veronez et al. 2006, 2011), transforming from cartesian coordinates to geodetic coordinates (Civicioglu 2012) and many others.

It is important to know that, over the past years, there has been a surge of interests within the geodetic community in converting cartesian coordinates into geodetic coordinates. This process is known as the reverse coordinate conversion technique. Numerous methods such as iterative, approximate, closed form, vector-based algorithms (Feltens 2007, 2009; Fok and Iz 2003; Gerdan and Deakin 1999; Shu and Li 2010) and artificial intelligence algorithms (Civicioglu 2012) have been proposed and used. More importantly, in Civicioglu (2012) the proposed differential search algorithm and computational-intelligence algorithms utilized were only tested in the reverse transformation and not the forward conversion. Hence, ANN was applied in this study to ascertain its ability to convert geodetic coordinates to cartesian coordinates (forward conversion).

Additionally, the regression technique was applied in this study as an alternative mathematical procedure to carry out the forward conversion. Several research works have been carried out using multiple linear regression (MLR) and simple linear regression (SLR) procedures in coordinate transformation from global datum to local datum and vice versa (Dawod et al. 2010; Featherstone 1997; Odutola et al. 2013;

Ziggah et al. 2012). It has been established that the regression techniques achievable accuracies are applicable to surveying and mapping related works. Therefore, this study adopted the MLR method over the SLR based on the multiple input parameters utilized in Eq. (1) to carry out the forward coordinate conversion. This will help maintain system consistency between MLR and Eq. (1). Here, the objective is to investigate and compare the performance of ANN to MLR in converting geodetic coordinates to cartesian coordinates.

However, a distinction should be drawn here that the artificial neural networks and the multiple linear regression technique have so far been applied only in coordinate transformation between different datums and not in coordinate conversion. The difference is that, in coordinate conversion as is the case in the present study, there is no change of datum and the geodetic ellipsoid parameters chosen for the transformation process are all based on the same geodetic datum (OGP 2012). That is, the geodetic coordinates converted into cartesian coordinates and vice versa are all based on the same geodetic datum, whereas, in coordinate transformation, the source and target coordinate reference systems are based on different datums. Here, transformation parameters are empirically determined and thus maybe subject to measurement errors (OGP 2012).

Furthermore, almost all the publications have not fully addressed the issue of applying alternative techniques in the forward conversion of geodetic coordinates to cartesian coordinates. Moreover, upon careful review of existing researches the authors realized that the ANN techniques in most cases serve as a better practicable alternative technology to the existing approaches.

In line with the above, this present study, for the first time, explored the regression method and the supervised learning technique of ANN in the forward conversion of geodetic coordinates to cartesian coordinates. To achieve this, the backpropagation artificial neural network (BPANN), radial basis function neural network (RBFNN) and multiple linear regression (MLR) methods were applied. This study also highlights the comparison between BPANN and RBFNN through the use of a set of training, validation and test data based on the results of statistic performance indicators such as mean square error (MSE), coefficient of determination (R^2), correlation coefficient (R), mean bias error (MBE), mean absolute error (MAE), noise to signal ratio (NSR), relative error correction (REC), Legates and McCabe index (LM) and three-dimensional position error. The findings showed that BPANN compared to RBFNN had the latter model more significant for converting geodetic coordinates to cartesian coordinates. Furthermore, comparison between the BPANN, RBFNN and MLR models revealed that the RBFNN produced much better results than the BPANN and MLR. Hence, this study will serve as an added contribution to existing knowledge of ANN applications in geodesy.

The paper is structured as follows: in Sect. 2 the necessary background information on ANN is first presented before narrowing it down to the steps adopted in the ANN models development. This is followed by a description of BPANN and RBFNN structures. Section 3 covers a brief overview of multiple linear regression technique. Statistical performance indices used to assess the ANN models prediction capabilities are presented in Sect. 4. Section 5 contains the application of the ANN and MLR methods of practical case, while Sect. 6 presents the concluding remarks.

2 Artificial Neural Network Methods

Artificial neural network (ANN) is a technology inspired by the adaptive, parallel computing style of the human brain and functions through a variety of theoretical concepts and computer analogies (Arbib 2003). It involves the use of computational techniques to model the structure, operation and behaviour that imitate the properties of biological neural networks. ANN consists of a number of neurons interconnected with links of variable synaptic weights that processes information fed into the network through an interaction with the environment. Thus, each neuron unit receives input information weighted by a factor which signifies the strength of the synaptic connection to produce an output. This output is then sent as a new input to another neuron by adapting new weights if the total sum of the weighted inputs is above a certain threshold. Generally, there are different types of ANN based on their architecture. However, a typical ANN architecture applied in most disciplines such as geodesy is structured into different layers. It is worth mentioning that the application of ANN in mathematical geodesy can be categorized into two aspects. That is, the type of function approximation problem to be solved and the type of training algorithm to be applied. The supervised training algorithm was adopted in this study for the ANN model development and subsequent prediction. This is because in the supervised learning, the training data comprise of training examples with each example having a pair of input vector and desired output data. The main objective here is to build a forecasting model that can produce reasonable predictions for mapping new examples. Besides, the supervised training provides the opportunity to interpret the output results based on the training values. In this study, an optimized BPANN and RBFNN models for converting geodetic coordinates into cartesian coordinates were developed. The choice of these networks was based on their frequent use as universal function approximators (Hartman et al. 1990; Hornik et al. 1989; Park and Sandberg 1991) within the geoscientific disciplines. Figure 1 shows a typical ANN architecture with inputs (X_1, X_2, …, X_N) and output (Y).

In order to develop the BPANN and RBFNN prediction models and achieve the results presented in this paper, the procedural steps adopted are described in the subsequent sections.

2.1 Data and Selection of Input Parameters

In the present study, a total of 328 GPS geodetic coordinates (φ, λ, h) collected by field measurement in Tarkwa, Ghana, located in West Africa, were used in the BPANN and RBFNN model formulation. It is well acknowledged that one of the contributing factors affecting the estimation accuracy of ANN is related to the quality of datasets used in model-building and selection of appropriate inputs parameters (Dreiseitl and Ohno-Machado 2002; Ismail et al. 2012). Therefore, to ensure that the obtained geodetic coordinate (φ, λ, h) data from the GPS receivers are reliable and accurate, several factors such as checking of overhead obstruction, observation period, observation principles and techniques as suggested by many researchers were considered (Yakubu and Kumi-Boateng 2011). In addition, all potential problems relating to GPS survey work were also considered.

The next step was the identification of the input parameters for the ANN training. It must be noted here that the input neurons act as control variables with an influence on the desired output of the neural network. Hence, the input data should represent the condition for which training of the neural network is done (Konaté et al. 2015). Consequently, the 328 GPS points measurement was first transformed into cartesian coordinates (X, Y, Z) using Eq. (1). The GPS WGS84 semi-major axis, a and semi-minor axis, b parameter values of 6,378,137.0 and 6,378,299.99899 m were implemented in Eq. (1). It is important to note that, in order for the ANN models to have consistency with Eq. (1), the radius of curvature in the prime vertical (N) and the square of the first eccentricity (e^2) values were estimated separately. With this in mind, (ϕ, λ, h, N, e^2)was used as the input layer data while (X, Y, Z) was used as the output layer data.

2.2 Data Normalization

In order to train the neural network, data set must be normalized. Generally, the original data are expressed in different units with different physical meaning. Therefore, to ensure constant variability in the ANN model, data set are usually normalized to a certain interval such as [−1, 1], [0, 1] or other scaling criterion. This data normalization improves convergence speed and doing so, reduces the chances of getting stuck in local minima. In this study, the selected input and output variables were normalized into the interval [−1, 1] using Eq. (4) (Muller and Hemond 2013; Wu et al. 2012b)

$$y_i = y_{\min} + \frac{(y_{\max} - y_{\min}) \times (x_i - x_{\min})}{(x_{\max} - x_{\min})}, \tag{4}$$

where y_i represents the normalized data, x_i is the measured coordinate values, while $x_{\min}$ and $x_{\max}$ represent the minimum and maximum value of the measured coordinates with $y_{\max}$ and $y_{\min}$ values set at 1 and −1, respectively.

2.3 ANN Architecture

The ANN models that are popular and widely used in solving most function approximation problems in geosciences are, respectively, the BPANN and RBFNN. In this study, the BPANN and RBFNN were utilized due to its frequent application. These networks have a feed forward topology consisting of input, hidden and output layers that are fully connected together. Detailed description of their structures is given in Sects. 2.5 and 2.6, respectively.

2.4 Network Training

It is well known that datasets are trained in ANNs to generate the required desired output for a particular input. Likewise, in this study, the aim is to train the ANNs to find an approximation of the functional relation between cartesian coordinates (X, Y, Z)

and selected input variables $(\phi, \lambda, h, N, e^2)$. Owing to the special property of ANN technology as a black box system with no specific function expression, the authors assumed a functional relation for the trained neural networks expressed in Eqs. (5–7) as

$$X_i = F\ (\phi, \lambda, h, N, e^2)_i, \tag{5}$$

$$Y_i = G\ (\phi, \lambda, h, N, e^2)_i, \tag{6}$$

$$Z_i = H\ (\phi, \lambda, h, N, e^2)_i, \tag{7}$$

where F, G and H are functions that relate the input vectors with corresponding output vectors while $i = 1, 2, 3, \ldots, N$ (N = number of observation points). To carry out the network training, the data set after normalization was divided into three subsets, namely training, validation and testing.

Here, 150 points were selected out of the 328 GPS point measurement as reference points set $P = (P_1, P_2, \ldots P_{150})$ and applied as the training set. Out of the remaining 178 data points, 78 were used as validation set $V = (V_1, V_2, V_3, \ldots, V_{78})$ while the remaining 100 points were used as the test set $T = (T_1, T_2, T_3, \ldots, T_{100})$. The training set served as parameterization (weight adjustment) to minimize the error function, while the validation set was used for fine-tuning the parameters of the network during learning. This provided an unbiased estimate of the generalization error. For the network training, the Levenberg-Marquardt backpropagation algorithm was used to train the BPANN and the gradient descent rule was used to train the RBFNN. During this phase, the training and validation data sets were used together. The two networks (BPANN and RBFNN) were allowed to train until no additional effective improvement occurred. As a result, if there was a significant change in terms of error between the training and validation results, then there was a possibility of overfitting occurring. In such situations, the error on the validation set typically begins to rise although at the initial phase of training, both the validation and training error were at a minimum. Such occurrences will make it difficult for the network to perform better when unseen data (testing data) are presented to it. This implies that the ANN has memorized the specific details of the training set instead of the general pattern found in all present and future data. After training the networks, the testing data which had no effect on training were applied to the trained models to provide an overall independent assessment of the network performance.

In determining the optimum BPANN and RBFNN model, the mean squared error (MSE) of all the models were monitored at each stage of training, validation and testing. In addition, the coefficient of determination (R^2) and correlation coefficient (R) were used to judge the performance of the ANN models. After several trials, the model with the lowest MSE value and highest R^2 and R values was selected as the better model. Other statistic indicators for evaluating the BPANN and RBFNN models obtained results are given in Sect. 4. It is notable to know that only the results given by the optimum performing BPANN and RBFNN models are presented in this study.

2.5 Backpropagation Artificial Neural Network

The backpropagation artificial neural network (BPANN) is one of the most widely used ANN models in geodesy (Lin and Wang 2006; Mihalache 2012; Tierra and Romero

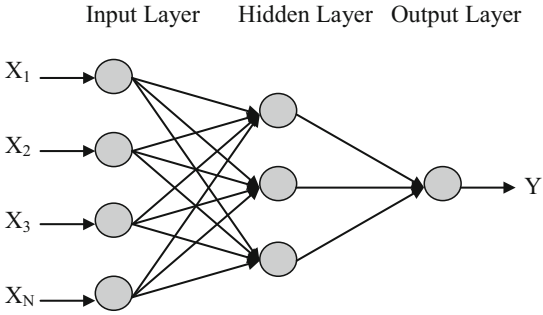

Fig. 1 Artificial neural network structure

2014; Turgut 2010; Yilmaz and Gullu 2012; Zaletnyik 2004). This network as shown in Fig. 1 contains input, hidden and output layer of processing units with each layer feeding input to the next layer in a feedforward manner through a set of connection weights (Yegnanarayana 2005). The input layer is an outlet that receives the input information, whereas the output layer gives the final results of the computation. In between these two layers is the hidden layer chamber where data transferred from the input layer are analysed and processed.

It is well known that the efficiency of BPANN model depends on the number of hidden neurons, hidden layers and type of activation functions applied. Usually, the number of hidden neurons is obtained through the sequential trial and error approach. This is partly due to (i) the type of problem at hand, (ii) the choice of neural network architecture and (iii) the proposed theoretical concepts that are yet to be universally accepted to clarify the number of hidden units needed to approximate a given function. In this study, the optimum number of neurons in the hidden layer was obtained based on the smallest mean squared error.

Determining the number of hidden layers in this work was based on Hornik et al. (1989) where it was proven that the BPANN with one hidden layer is sufficient as a universal approximator of any discrete and continuous functions. Hence, one hidden layer was used in this research. Moreover, to introduce non-linearity into the network, the hyperbolic tangent activation function was selected for the hidden units while a linear function was applied for the output nodes. The hyperbolic tangent function (Yonaba et al. 2010) is defined in Eq. (8) as

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1, \tag{8}$$

where x is the sum of the weighted inputs.

It is important to note that the BPANN training can be characterized as a non-linear optimization problem, w^* (Konaté et al. 2015), given by Eq. (9)

$$w^* = \arg\min E(w), \tag{9}$$

where w is the weight matrix and $E(w)$ is the error function. The purpose of the network training is to find the optimum weight connection (w^*) that minimizes $E(w)$

such that the estimated outputs from the BPANN will be in good agreement with the target data. This $E(w)$ (Konaté et al. 2015) is evaluated at any point of w shown in Eq. (10) as

$$E(w) = \sum_{n} E_n(w), \tag{10}$$

where n is the number of training examples and $E_n(w)$ is the output error for each example n. $E_n(w)$ (Konaté et al. 2015) is mathematically defined by Eq. (11)

$$E_n(w) = \frac{1}{2} \sum_{j} (d_{nj} - y_{nj}(w))^2, \tag{11}$$

where d_{nj} and $y_{nj}(w)$ are desired network outputs and estimated values of the jth output neuron for the nth example, respectively. Therefore, substituting Eq. (11) into Eq. (10) gives the objective function to be minimized expressed in Eq. (12) (Konaté et al. 2015) as

$$E(w) = \frac{1}{2} \sum_{n} \sum_{j} (d_{nj} - y_{nj}(w))^2. \tag{12}$$

The training process continues by adjusting the weight of the output neurons and then proceeds towards the input data until the error function reaches an acceptable value. There exist several numerical optimization algorithms to perform this weight adaptation (Kecman 2001; Nocedal and Wright 2006). In this study, Levenberg-Marquardt algorithm (LMA) was chosen to train the BPANN because it is faster and has more stable convergence compared to the popular gradient descent algorithm. This was proven in the works of Hagan and Menhaj (1994) and Wang (2009). The LMA can be thought of as a combination of steepest descent and the Gauss–Newton method. The algorithm behaves like steepest descent method when the current solution is far from the correct one, thus slow but guaranteed to converge. On the other hand, when the algorithm approaches the correct solution, it becomes a Gauss–Newton method. Detailed mathematical theory of LMA can be found in (Lourakis 2005; Nocedal and Wright 2006).

2.6 Radial Basis Function Neural Network

The radial basis function neural network (RBFNN) has a feedforward topology embedded in three layers: input, hidden and output layers that are completely linked together. An example is as shown in Fig. 1. This network is basically used for supervised training.

The RBFNN input layer receives information into the network and transfer into the hidden layer space by means of unweighted connections. The hidden layer then transforms the input data by means of a non-linear function. Within the hidden layer, each neuron calculates a Euclidean norm that shows the distance between the input

to the network and the position of the neuron called the centre. This is then inserted into a radial basis activation function which calculates and outputs the activation of the neuron (Deyfrus 2005). The present study applied the Gaussian activation function (Gurney 2005) expressed in Eq. (13) as

$$\theta_j(X) = \exp\left[-\frac{\|X-\mu_i\|^2}{2\sigma_j^2}\right], \tag{13}$$

where X is the input vector, μ_i is the centre of the Gaussian function and σ_j is the spread parameter of the Gaussian bells and $\|\|$ is the Euclidean norm. The output layer contains the linear function and uses the weighted sum of the hidden layer as propagation function.

Let $y_{ki}^{(H)}$ be the output of the kth radial basis function on the ith sample. The output of each target node j is computed using the weights w_{jk} (Michie et al. 1994) expressed in Eq. (14) as

$$y_{ji} = \sum_k w_{jk} y_{ki}^{(H)}. \tag{14}$$

Let the target output for sample i on target node j be Y_{ji}. The error function $E(w)$ (Michie et al. 1994) is written in Eq. (15) as

$$E(w) = \frac{1}{2}\sum_{ji}\left(\sum_k w_{jk} y_{ki}^{(H)} - Y_{ji}\right)^2, \tag{15}$$

which has its minimum where the derivative (Eq. (16))

$$\frac{\mathrm{d}E}{\mathrm{d}w_{rs}} = \sum_k \sum_i w_{rk} y_{ki}^{(H)} y_{ji}^{(H)} - \sum_i Y_{ri} y_{si}^{(H)}, \tag{16}$$

vanishes. Let R be the correlation matrix of the radial basis function outputs given by Eq. (17) as follows

$$R_{jk} = \sum_i y_{ki}^{(H)} y_{ji}^{(H)}. \tag{17}$$

The weight matrix w^* (Eq. (18)) which minimises E lies where the gradient vanishes

$$w_{jk}^* = \sum_r \sum_i Y_{ji} y_{ri}^{(H)} (R^{-1})_{rk}. \tag{18}$$

Thus, the problem is solved by inverting the square H × H matrix R, where H is the number of radial basis function. This matrix inversion can be solved using the singular value decomposition (SVD) approach whereby an approximate inverse is provided by

diagonalising the matrix. The eigenvalues exceeding zero are inverted by a parameter-specified margin and transformed back to the original coordinates. This provides an optimal minimum-norm approximation to the inverse in the least-mean-squares sense (Michie et al. 1994). This training process continues until the network error reaches an acceptable value.

3 Multiple Linear Regression

The multiple linear regression (MLR) is an extensively used technique for articulating the dependence of a response variable on several explanatory variables. It fits a linear combination of the components of multiple input parameters x_i to a single output parameter y defined in Eq. (19) (Ghorbani et al. 2015) as

$$y = a_0 + \sum_{i=1}^{N} a_i x_i, \tag{19}$$

where a_0 is the intercept (values when all the independent variables are zero) while a_i values denote the regression coefficients which were determined in this study using the least square approach. Here, i in Eq. (19) is an integer varying from 1 to N, where N is the total number of observations. Since there are several variables that can be used as candidates for predictor variables in the MLR model formulation, it would be demanding having to try every possible combination of variables. Therefore, in this study, Pearson correlation analysis was applied as a criterion for predictor selection in a stepwise manner.

4 Assessment of Model Quality

The BPANN and RBFNN models' performance for training, validation and testing data was evaluated. Thus, by examining the discrepancies between the measured training, validation and testing data to those predicted by the BPANN and RBFNN models. In this study, mean squared error (MSE) (Ali and Abustan 2014) was used as a criterion to determine the optimum BPANN and RBFNN structures. The correlation coefficient (R) (Banarjee et al. 2011) and coefficient of determination (R^2) (Krause et al. 2005), on the other hand, were used to judge the ANN models performance. Their respective mathematical representations are given by Eqs. (20–22), respectively, as

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (O_i - P_i)^2, \tag{20}$$

$$R = \left(\frac{\sum_{i=1}^{N} (O_i - \overline{O})(P_i - \overline{P})}{\sqrt{\sum_{i=1}^{N} (O_i - \overline{O})^2} \times \sqrt{\sum_{i=1}^{N} (P_i - \overline{P})^2}} \right), \tag{21}$$

$$R^2 = \left(\frac{\sum_{i=1}^{N} (O_i - \overline{O})(P_i - \overline{P})}{\sqrt{\sum_{i=1}^{N} (O_i - \overline{O})^2} \times \sqrt{\sum_{i=1}^{N} (P_i - \overline{P})^2}} \right)^2. \tag{22}$$

Additionally, to produce a more comprehensive model performance analysis consistent with the training, validation and testing results, dimensioned error statistic indicators such as mean bias error (MBE), mean absolute error (MAE) and noise to signal ratio (NSR) were utilized in the model valuation. Their mathematical expressions (Ali and Abustan 2014; Banarjee et al. 2011; Krause et al. 2005) are given by Eqs. (23–25) as

$$\mathrm{MBE} = \frac{1}{N}\sum_{i=1}^{N}(O_i - P_i), \tag{23}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|O_i - P_i|, \tag{24}$$

$$\mathrm{NSR} = \frac{\left[\frac{1}{N-1}\sum_{i=1}^{N}(O_i - P_i)^2\right]^{1.2}}{\overline{O}}. \tag{25}$$

Furthermore, two model efficiency-based statistics, Legates and McCabe index (LM) and relative error correction (REC), were also implemented. They are expressed (Legates and McCabe 1999; Hu et al. 2014) in Eqs. (26–27) as

$$\mathrm{LM} = 1 - \frac{\sum_{i=1}^{N} Abs(O_i - P_i)}{\sum_{i=1}^{N} Abs(O_i - \bar{O})}, \tag{26}$$

$$\mathrm{REC} = 100 - \left[\frac{1}{N}\sum_{i=1}^{N}\frac{|O_i - P_i|}{O_i} \times 100\right]. \tag{27}$$

With reference to Eqs. (20–27), $\mathrm{O_i}$ is the measured cartesian coordinates and $\bar{O}$ is the mean of the measured cartesian coordinates. While $\mathrm{P_i}$ is the predicted cartesian coordinates and $\overline{P}$ is the mean of the predicted cartesian coordinates. i is an integer varying from 1 to N where N is the total number of observations.

5 Application

5.1 ANN Models Developed

The main aim of this study was to convert geodetic coordinates (φ, λ, h) into cartesian coordinates (X, Y, Z) using artificial neural network (ANN) technology to develop prediction models. The proposed ANN models (BPANN and RBFNN) accepted for converting geodetic coordinates (ϕ, λ, h) into cartesian coordinates (X, Y, Z) consist of three layers: input layer, hidden layer and output layer. For the ANN model formulation, the data set was split into three divisions: training, validation and testing sets, as described in Sect. 2.4. The supervised learning procedure was applied to the BPANN and RBFNN with their corresponding input (ϕ, λ, h, N, e^2) and target (X, Y, Z) data

Table 1 The results from the training, testing and validation data sets for the ANN models

Data set	BPANN			RBFNN		
	X	*Y*	*Z*	*X*	*Y*	*Z*
Training						
MSE (m)	1.0368E−08	3.1508E−06	1.5570E−08	2.1155E−10	2.5308E−10	1.7889E−10
R^2	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
R	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
Validation						
MSE (m)	7.5631E−08	3.3258E−03	2.2799E−06	6.3710E−08	4.9676E−07	6.3794E−08
R^2	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
R	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
Testing						
MSE (m)	4.0335E−06	2.7128E−02	1.7799E−06	2.7898E−08	5.1881E−07	6.5851E−08
R^2	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
R	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999

normalized between −1 and 1. In this study, the BPANN structure used is made up of five inputs with a hyperbolic tangent function from the input layer to a single hidden layer, and a linear transfer function to the output layer. The network was trained for 1000 epochs using the Levenberg–Marquardt backpropagation algorithm with a learning rate of 0.03 and a momentum coefficient of 0.7. It has been established that working with the momentum term makes optimization more robust with respect to the choice of learning rate (Kecman 2001), thereby improving convergence speed in the network. The five input data were trained and tested by the BPANN, and the optimal number of neurons in the hidden layer was identified based on a sequential trial and error procedure. The network that yielded the best results with the lowest MSE, largest R^2 and R values from the testing dataset was selected as the best BPANN scheme. After several trials, the optimum structure of the BPANN for converting geodetic coordinates into X, Y and Z cartesian coordinates was $[5-8-1]$, $[5-4-1]$ and $[5-7-1]$, respectively. Thus, for X output vector, there are five inputs with eight hidden neurons. Y and Z output vector had four and seven hidden neurons with five inputs. Table 1 shows the optimum performance results of BPANN in training, validation and testing period.

The RBFNN model, on the other hand, comprises of five inputs, one hidden layer using Gaussian function as the non-linear activation and an output layer containing linear activation function. The network was trained using the gradient descent learning algorithm in which the weights are adapted in part to the deviation between the predicted output and target output. In determining the best RBFNN structure, the MSE, R^2 and R of all the trained models were examined at each phase of training, validation and testing. The model that gave the smallest MSE, largest R^2 and R in the testing dataset was selected as the optimum RBFNN architecture. The optimum RBFNN structure selected for converting geodetic coordinates into cartesian coordinates was five inputs with one hidden layer of 30 neurons for each output vector

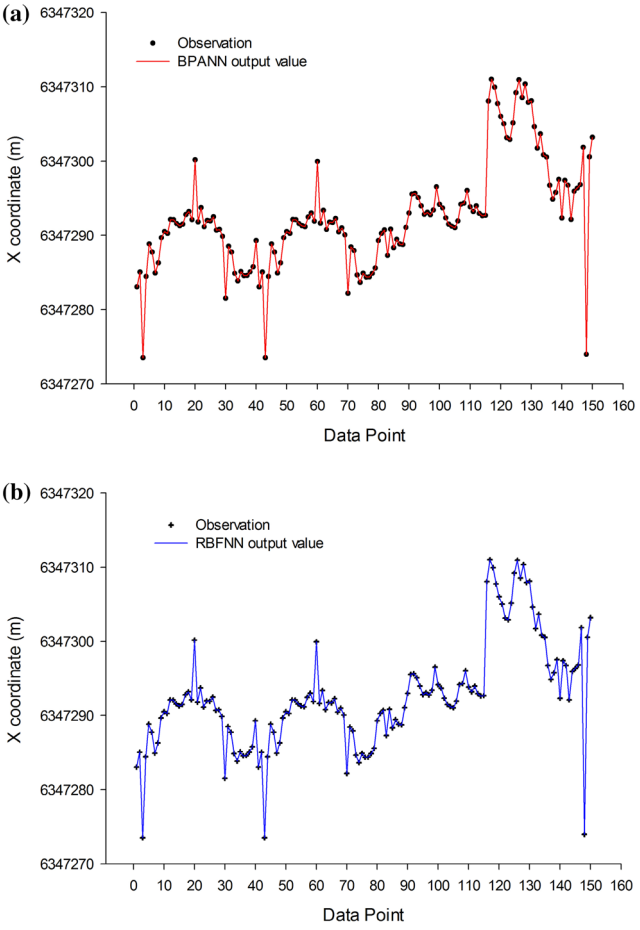

Fig. 2 Training data (X coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

(X, Y, Z), that is, $[5-30-1]$. Their respective width parameters (σ) determining the extent of each radial basis function utilized in the RBFNN structures for predicting X, Y, and Z cartesian coordinates were 13, 22, and 26. Table 1 shows the optimal performance results obtained in training, validation and testing stages of the RBFNN approach.

It is pertinent to note that the MSE utilized in this study was acting as the optimality criterion to aid in selecting the best BPANN and RBFNN structures for converting geodetic coordinates into cartesian coordinates. On the basis of the results, the MSE values (Table 1) obtained for the training data in both BPANN and RBFNN structures considering the X, Y and Z coordinates depict the closeness of the models fit to the measured training data. Figures 2a, b, 3, 4a, b depict this closeness of fit of the BPANN and RBFNN predictions to the measured cartesian coordinates based on the training data.

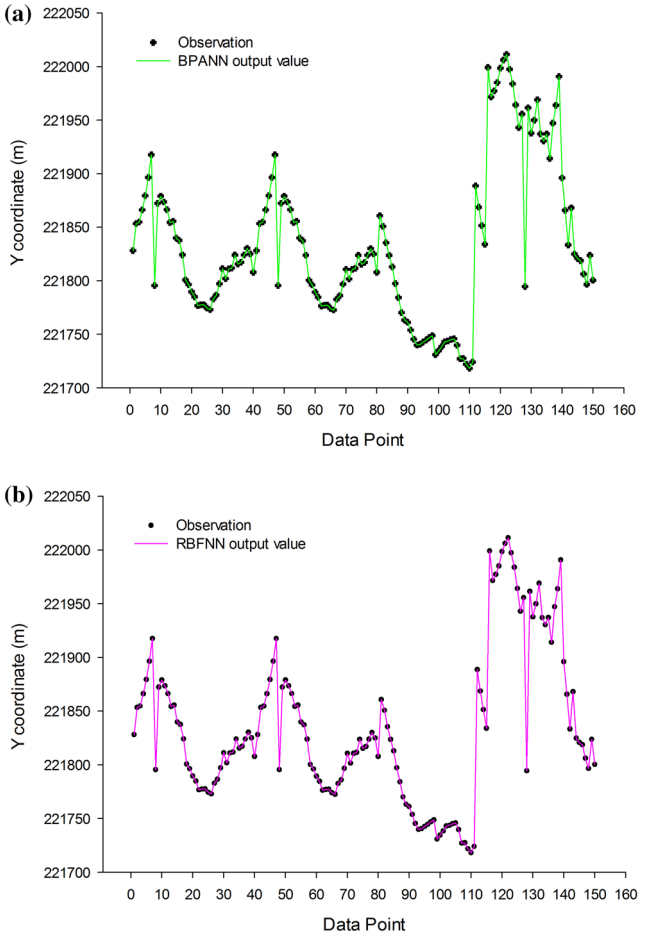

Fig. 3 Training data (Y coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

Moreover, the high R^2 and R values (Table 1) obtained in training further confirmed the quality of training performances in both BPANN and RBFNN structures. Here, the R^2 values attained for the training data indicate the tolerability of the BPANN and RBFNN prediction values. Thus, 99 % changes in the measured cartesian coordinates (training target values) are explained by the variation in training predicted output values. The R findings, on the other hand, show the strength and direction of linear dependency existing between the training targets and the predicted training outputs. Correspondingly, the MSE, R^2 and R values at the training stage also imply that the calibration capability of the BPANN and RBFNN structures are better for the given training data. Judging from the outcomes in Table 1, it can be inferred that both BPANN and RBFNN structures achieved satisfactory training performance based on the MSE values, hence indicating that in the case of this study, the BPANN and RBFNN have both demonstrated greater learning abilities. Specifically, the results

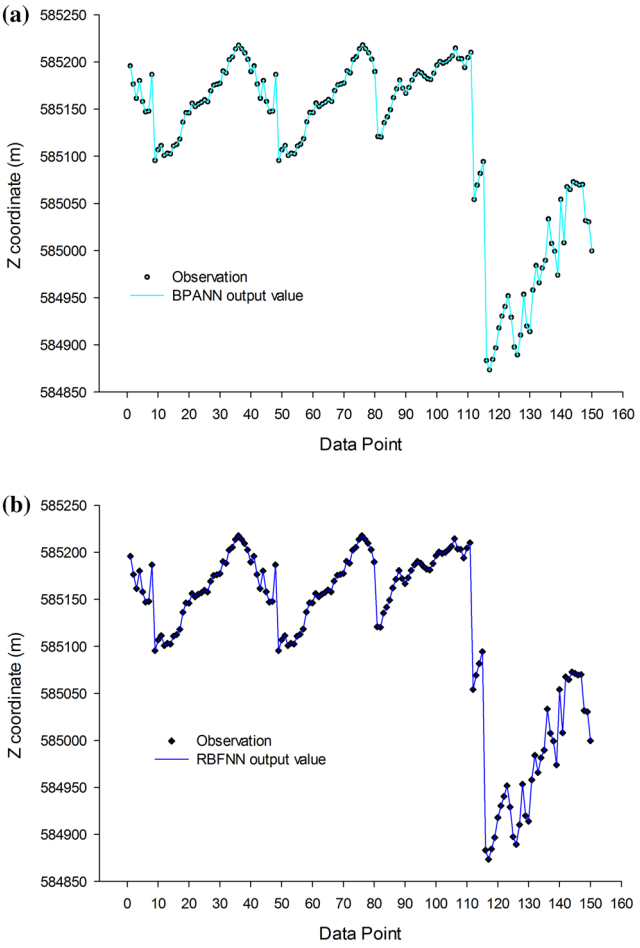

Fig. 4 Training data (Z coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

from Table 1 revealed superiority in RBFNN training performance, since RBFNN structure produced the lowest MSE compared to BPANN although identical R^2 and R values were achieved by the two ANNs.

In the validation stage, performances of the various ANNs prediction model that were created based on the training dataset were compared thus to estimate how well the BPANN and RBFNN structures had been trained. It is interesting to know that in Table 1, the best performance of the trained BPANN and RBFNN models based on the MSE results obtained from the validation data are in close agreement with the MSE values obtained in training. The MSE results indicate a good statistical estimation measure of the residuals generated by the BPANN and RBFNN from the validation data. Besides, the high values of R^2 and R attained were also in proximity to that from the training data. The inference to be made here is that the validation results did not show any significant overfitting scenario of the trained BPANN and RBFNN

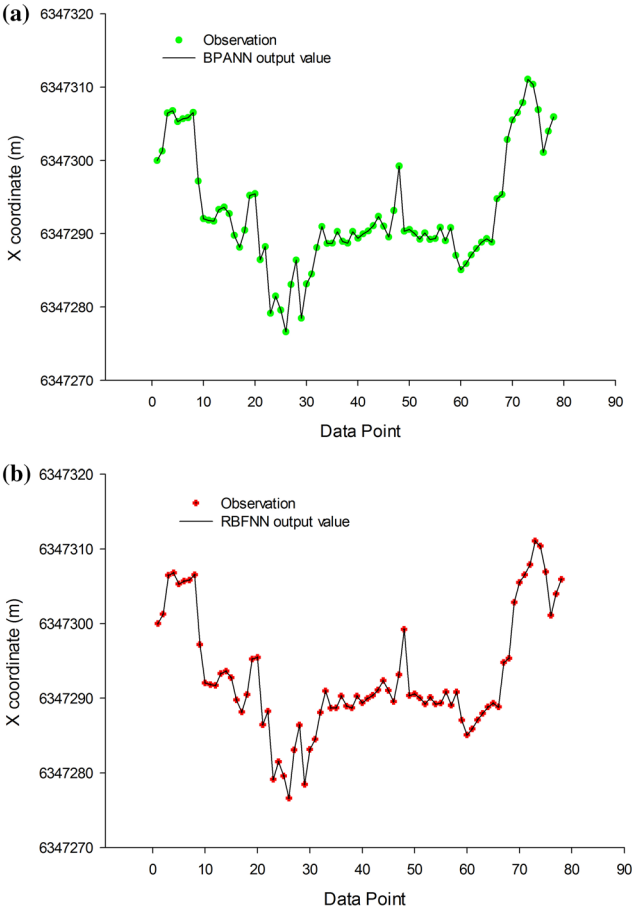

Fig. 5 Validation data (X coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

structures. Figures 5a, b, 6 and 7a, b show the plot of the predicted validation output values from the trained BPANN and RBFNN models using the validation data.

When the training and validation phase was over, the testing phase was carried out. This was necessary to independently test and confirm the predictive power of the optimized BPANN and RBFNN structures. In this procedure, the optimized ANN models applied on the test data provided a realistic estimate of their performance on entirely unseen data. The statistical performance criteria (MSE, R^2 and R) achieved by the BPANN and RBFNN using the testing data are presented in Table 1. The testing data MSE values obtained further provided an independent measure of the BPANN and RBFNN models' performance. By virtue of the testing data MSE results (Table 1), it can obviously be concluded that the predicted testing outputs rendered by the trained BPANN and RBFNN models are significantly close to the desired testing target outputs after the testing data (untrained) were introduced to the neural networks. This indicates

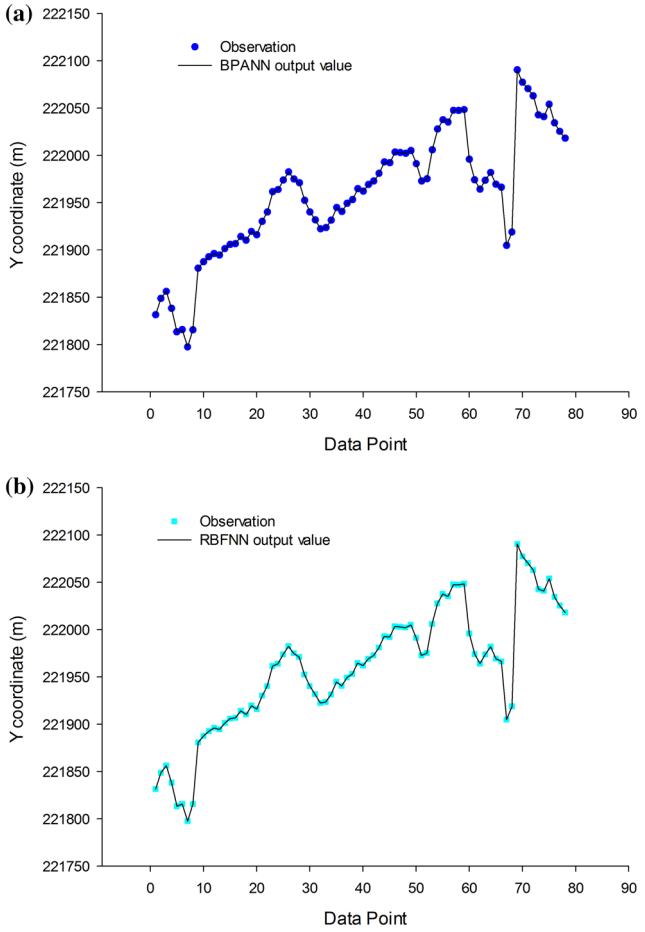

Fig. 6 Validation (Y coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

that the BPANN and RBFNN are well developed and have achieved generalization. This assertion can also be gathered from Figs. 8a, b, 9 and 10a, b correspondingly.

Again, taking into account the MSE test results (Table 1), it was uncovered that the RBFNN outperformed the BPANN since RBFNN structure produced the lowest MSE values, while BPANN suffered from higher MSE values. It can, therefore, be concluded that, in this study, the RBFNN scheme fits cartesian coordinates better than BPANN.

Furthermore, the adequacy of the BPANN and RBFNN predictions based on test data was ascertained using the R^2. Upon inspection, it was evident from Table 1 that both BPANN and RBFNN model yielded identical R^2 values (testing data). These values of R^2 are evidently showing that the predicted test data points fall closely to the line of best fit and thus, the BPANN and RBFNN models delineate discrepancy in the test data with high precision and accuracy. The testing data R values (Table 1)

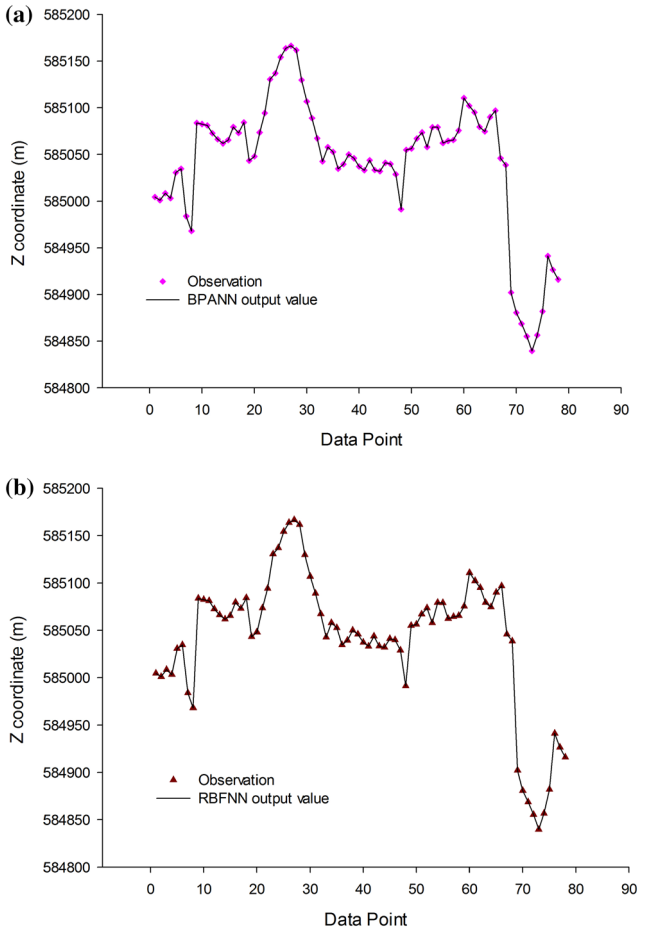

Fig. 7 Validation data (Z coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

supported this high strength of correlation between measured and predicted test coordinates. Similarly, the high R values obtained for the testing data indicate that both BPANN and RBFNN implemented in this study have demonstrated good potential in converting geodetic coordinates to cartesian coordinates.

5.2 Residual Analysis

With reference to Figs. 2a, b, 3, 4, 5, 6, 7, 8, 9, 10a, b, due to the closeness of the respective predicted outputs from the BPANN and RBFNN models, it is a challenge to visually identify the amount the predicted outcomes deviate from their corresponding measured data. Hence, it is essential that a plot depicting clearly the variation of the residuals obtained from the BPANN and RBFNN models in training, validation and

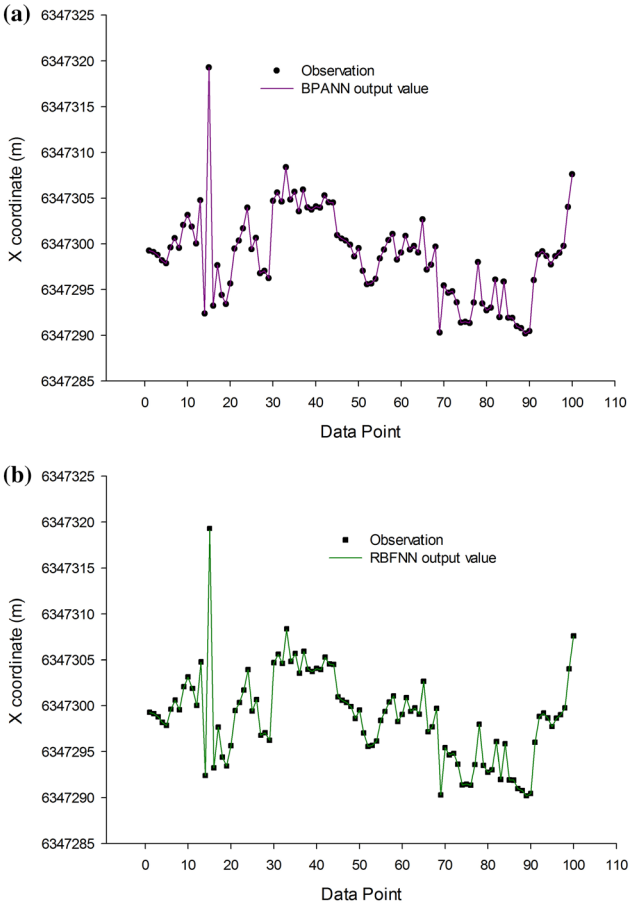

Fig. 8 Testing data (*X* coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

testing stages are shown. First, the residuals were estimated by subtracting the predicted coordinates at the various stages from its corresponding measured coordinates. The obtained residuals signify the prediction limitations of the BPANN and RBFNN models developed.

Figures 11a–c, 12, 13a, b, c show the variation in residuals with respect to the training, validation and test data points when the BPANN and RBFNN models were executed. Ideally, a functional relation model should produce zero error results. However, in real-world situations, this is almost impossible to achieve in function approximation related problems. Therefore, these graphical illustrations (Figs. 11a–c, 12, 13a–c) offer a better description of the dynamics on how much the BPANN and RBFNN models predicted coordinate values matched with the measured coordinates by way of errors along the ideal zero error (vertical axes).

A visual check in Figs. 11a–c, 12, 13a–c confirms graphical evidence that the RBFNN model was able to produce better predictions than BPANN model. In addition,

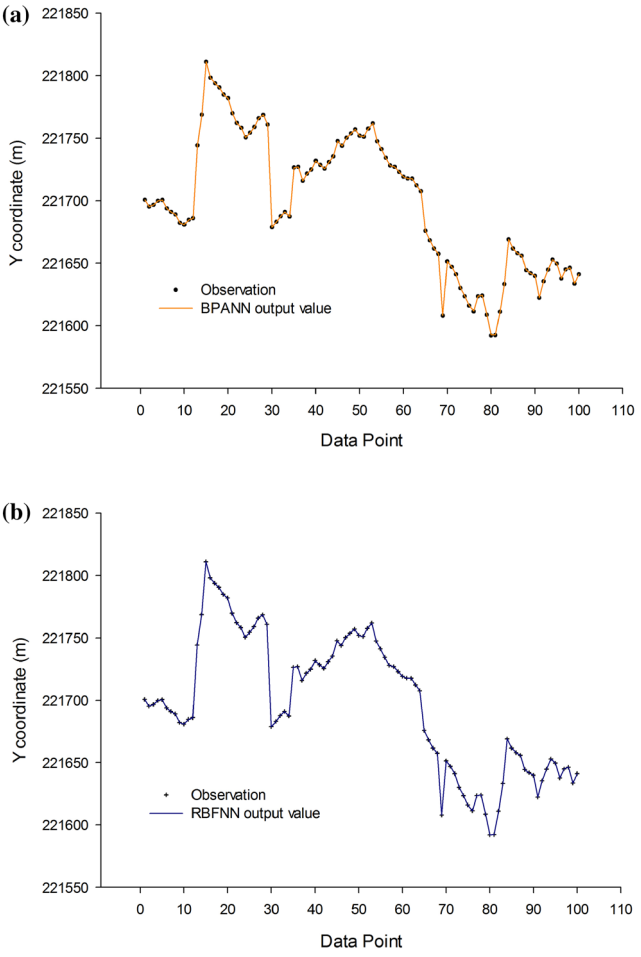

Fig. 9 Testing data (*Y* coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** radial basis function neural network (RBFNN)

a fairly steady change along the ideal zero error value was observed for RBFNN model. Although based on visual inspection, there have been slightly inconsistencies in the reduction trend of BPANN residuals; the BPANN model produced satisfactory results based on the error interval range. However, in comparison, the RBFNN was able to learn and generalize better than the BPANN.

It is well acknowledged that for practical applications, coordinates are usually utilized either in two-dimension or three-dimension. This means that individual coordinate points as a stand-alone do not have any physical meaning unless combined to form two-dimension or three-dimension. In view of this, the predicted X, Y, Z coordinates from the BPANN and RBFNN were considered in three-dimension. This enabled the practical application of the predicted outcomes of the BPANN and RBFNN models to be ascertained. A summary of the total error obtained using both the BPANN and

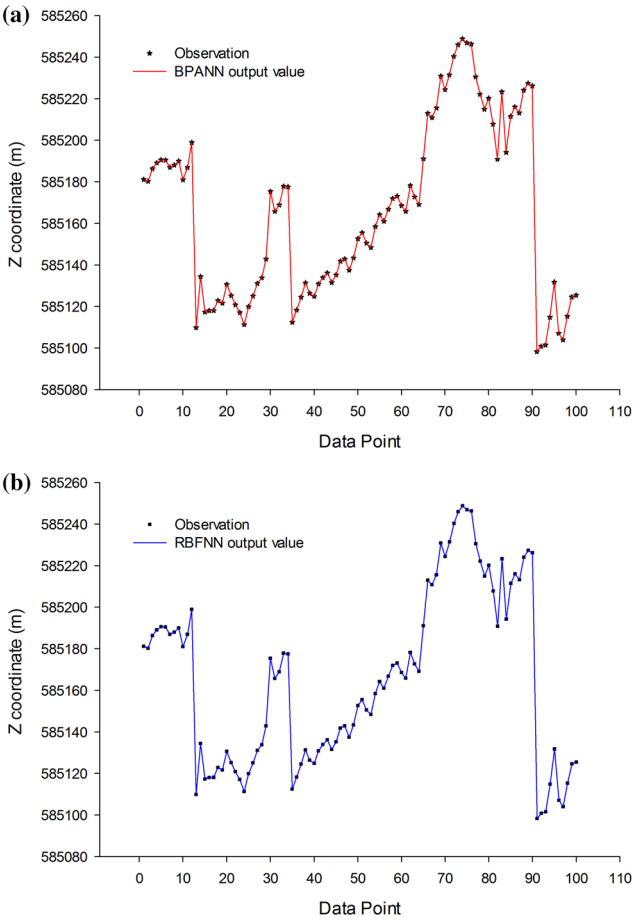

Fig. 10 Testing data (Z coordinates) prediction results for **a** backpropagation artificial neural network (BPANN) and **b** Radial basis function neural network (RBFNN)

RBFNN is presented in Table 2 by using the mean position error (MPE), standard deviation (SD), maximum and minimum coordinate differences as the error performance indices (EPI). They are given by Eqs. (28–32), respectively, as

$$\text{MPE} = \frac{1}{N}\sum_{i=1}^{N} 3\text{D PS}_i, \tag{28}$$

$$\text{SD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(3\text{D PS} - \overline{3\text{D PS}})^2}, \tag{29}$$

$$\text{Max MPE} = \max(3\text{D PS}), \tag{30}$$

$$\text{Min MPE} = \min(3\text{D PS}). \tag{31}$$

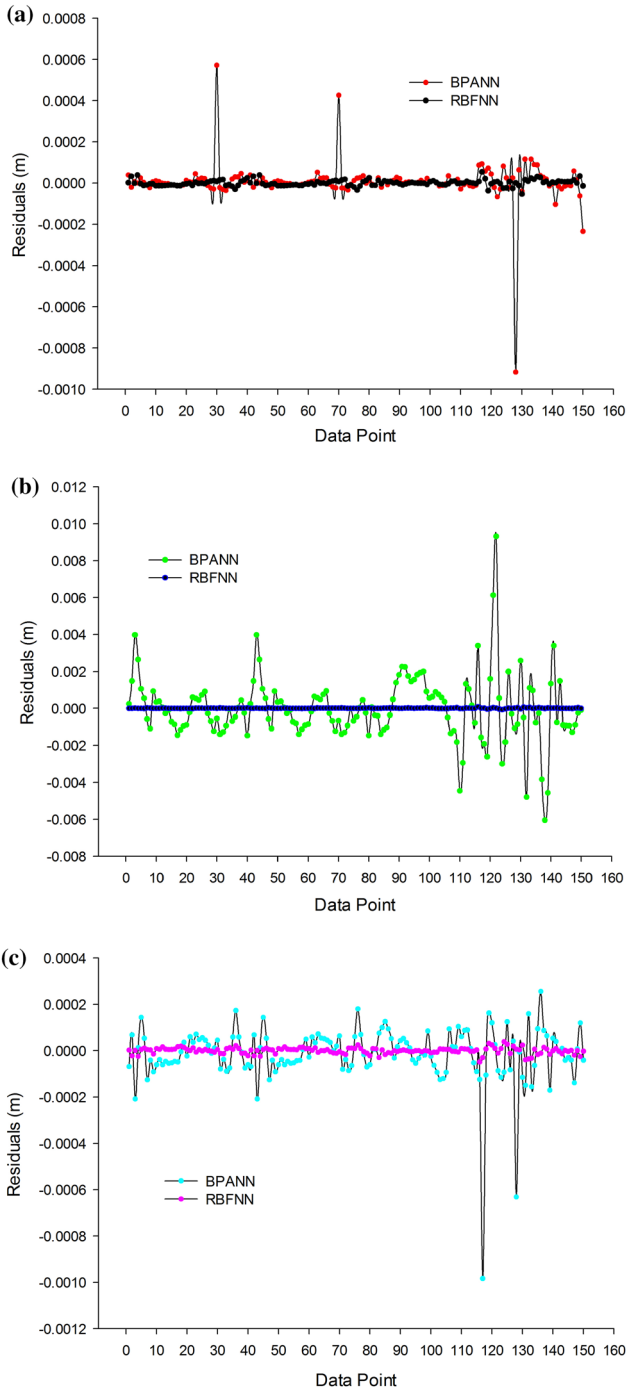

Fig. 11 Backpropagation artificial neural network (BPANN) and radial basis function neural network (RBFNN) training data descriptive error distribution for **a** X coordinates, **b** Y coordinates and **c** Z coordinates

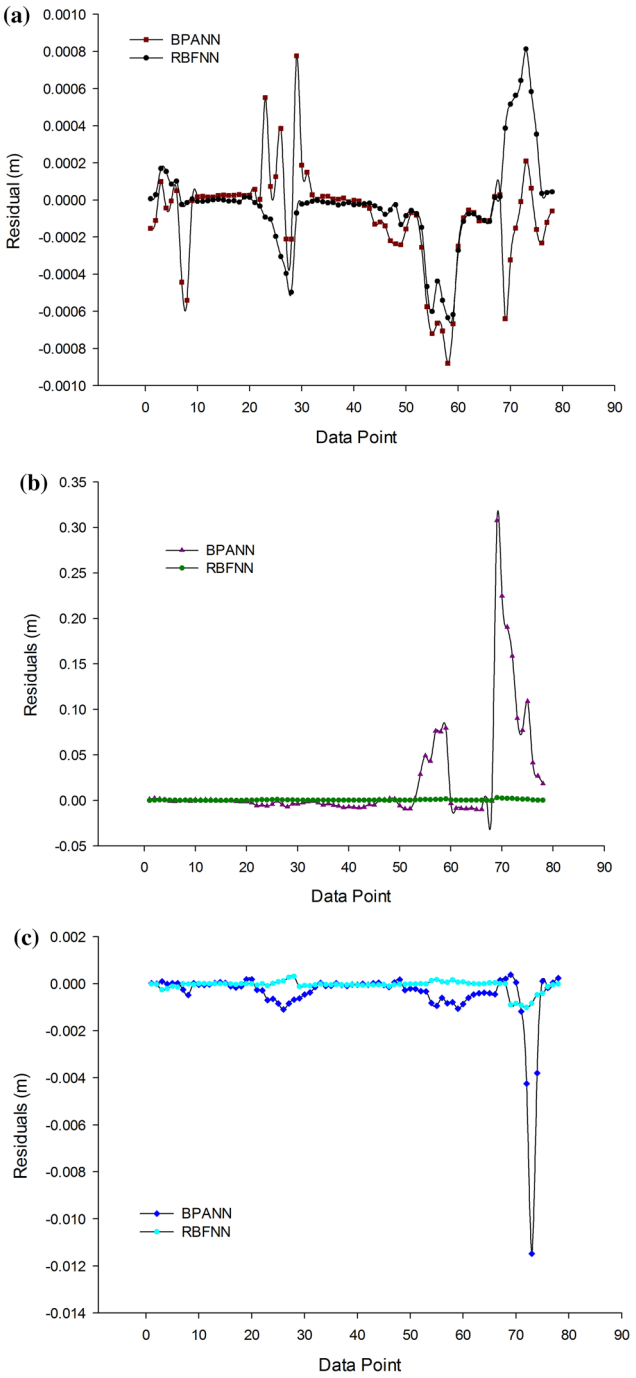

Fig. 12 Backpropagation artificial neural network (BPANN) and radial basis function neural network (RBFNN) validation data descriptive error distribution for **a** *X* coordinates, **b** *Y* coordinates and **c** *Z* coordinates

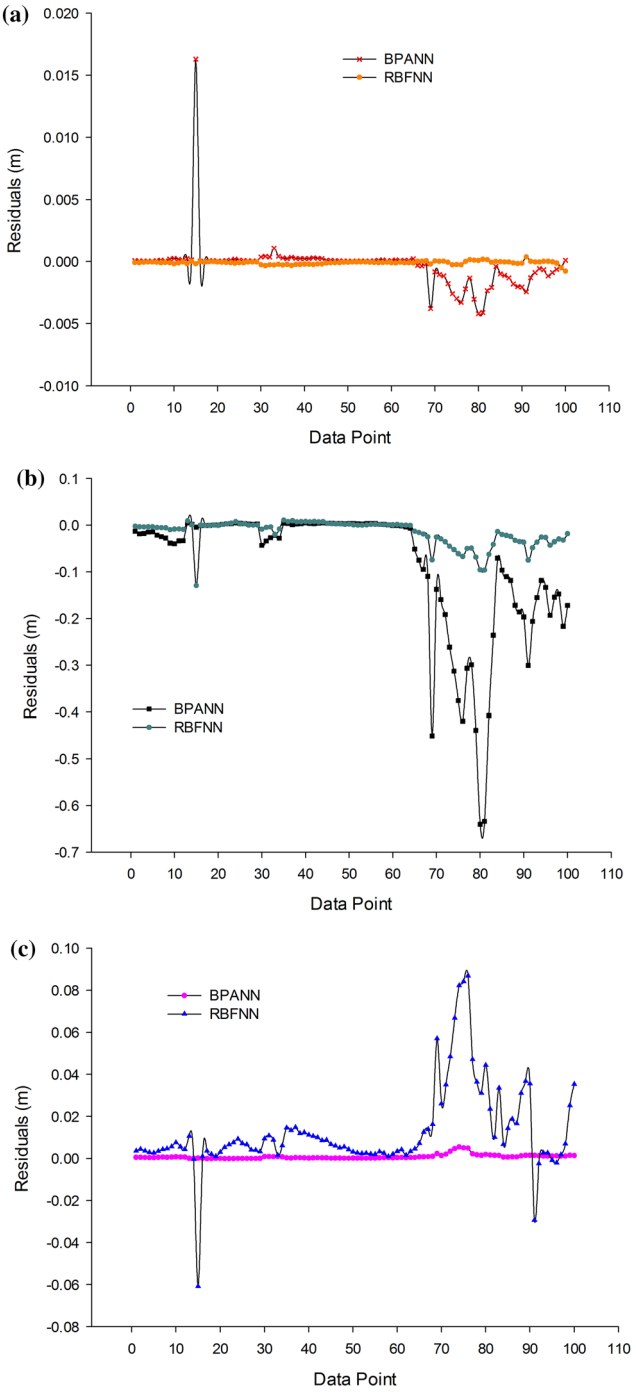

Fig. 13 Backpropagation artificial neural network (BPANN) and radial basis function neural network (RBFNN) testing data descriptive error distribution for **a** *X* coordinates, **b** *Y* coordinates and **c** *Z* coordinates

Table 2 Total error of the three-dimensional coordinate differences with the ANN models

EPI	BPANN			RBFNN		
	Training (m)	Validation (m)	Testing (m)	Training (m)	Validation (m)	Testing (m)
MPE	1.228E−03	0.023	0.090	1.976E−05	4.493E−04	4.726E−04
SD	1.290E−03	0.053	0.1390	1.597E−05	6.542E−04	6.270E−04
Min MPE	5.123E−05	8.249E−05	2.928E−04	5.110E−07	6.006E−06	1.647E−05
Max MPE	9.319E−03	0.308	0.640	9.127E−05	0.003	0.004

Here, 3D PS is the three-dimensional position error expressed in Eq. (32)

$$\text{3D PS} = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}, \tag{32}$$

where ΔX, ΔY, ΔZ is the individual coordinate differences between the predicted and measured in relation to the data set (training, validation and testing) under consideration.

Table 2 shows that in our case of the forward conversion from geodetic coordinates to cartesian coordinates, the RBFNN model is dominant over the BPANN model. This is in line with the RBFNN attaining the least mean three-dimensional position error (MPE) in training, validation and testing data, respectively (Table 2), thus indicating a massive improvement in three-dimensional positional accuracy of the converted coordinates given by the RBFNN model compared to the BPANN model. This assertion conforms to Fig. 14a–c where the three-dimensional position error in training, validation and testing data is displayed.

The estimated SD values for the three-dimensional position error signify a practical expression for the precision of the predicted training, validation and test outputs from the two ANN models. In Table 2, it can be seen that the RBFNN was again superior to the BPANN because, RBFNN had the least SD values which indicate the limit of the error bound by which every value within the predicted training, validation and testing data sets varies from its mean value. On the strength of the SD values obtained by the RBFNN model, it can be stated that its predicted outcomes are more precise and accurate than the BPANN model. Moreover, on the basis of the results in Table 2, it can also be seen that the RBFNN demonstrated good generalization capability in the testing data than the BPANN.

The inference made in line with the maximum and minimum values (Table 2) is that the RBFNN model predicted outputs in training, validation and testing differed by not more than 0.004 m, whereas 0.64 m was realized by the BPANN model. This additionally gives a better indication about the accuracy range of the two ANN models in terms of their practicality. In conformance to the maximum and minimum values (Table 2), the RBFNN model can thus serve as a better substitute for the standard forward transformation equation (Eq. 1) in converting geodetic coordinates to cartesian coordinates compared to the BPANN model.

Considerably, these errors incurred in this study during the BPANN and RBFNN models' formulation and subsequent application could be attributed to two factors. The

Table 3 ANN models performance assessment

Dataset	BPANN			RBFNN		
	X (m)	Y (m)	Z (m)	X (m)	Y (m)	Z (m)
Training						
MBE	5.33E−06	−1.75E−05	−1.45E−05	5.64E−09	1.84E−07	−1.02E−07
MAE	3.56E−05	1.22E−03	7.63E−05	1.06E−05	1.04E−05	9.98E−06
NSR	1.61E−11	2.81E−10	1.97E−11	2.30E−12	2.51E−12	2.11E−12
Validation						
MBE	−9.06E−05	0.017852	−4.77E−04	2.99E−05	3.93 E−04	−7.54E−05
MAE	1.70E−04	0.02337	5.36E−04	1.48E−04	3.93E−04	1.19E−04
NSR	4.36E−11	9.14E−09	2.39E−10	4.00E−11	1.12E−10	4.01E−11
Testing						
MBE	−3.03E−04	−0.08705	7.89E−04	−9.44E−05	−1.60E−04	1.29E0−4
MAE	8.20E−04	0.089347	8.12E−04	1.19E−04	3.94E−04	1.59E−04
NSR	3.18E−10	2.61E−08	2.11E−10	2.64E−11	1.14E−10	4.06E−11

first are subject to the issue that ANN methods are approximate functions. Second, random errors of the measured data (X, Y, Z) applied for the model formulations have an influence on the outcome of the estimation. Nonetheless, ANNs could still produce sub-meter accuracy predictions as was the case observed in the present study.

5.3 Dimensioned Error Statistic

In order to make further objective assessments of BPANN and RBFNN results attained, dimensioned error statistics (DES), namely MBE, MAE and NSR as stated in Sect. (4) were utilized. The choice of these indices was based on the recommendation in Fox (1981) that these indices are able to quantify model performance in terms of prediction accuracy, thus informing the modeller and the reader about the actual size of the error produced by the model. The closer these statistical indices are to zero the better the prediction abilities of the BPANN and RBFNN models. Table 3 presents the statistical results of the ANN models.

With reference to Table 3, the estimated MBE values in training, validation and testing data emphasize the degree of the average over-estimation (positive value) or under-estimation (negative value) by the BPANN and RBFNN models. On account of the MBE results (Table 3), it can be conveyed that the overall maximum over- and under-estimation by the RBFNN in X, Y and Z cartesian coordinates considering together the training, validation and testing data is approximately 3.93E−04 and −1.60E−04 m, respectively. The BPANN instead achieved maximum over- and under-estimation values of approximately 0.018 and −0.087 m, respectively, thus signifying that the RBFNN predicted outcomes deviated marginally from the measured data and thus they are more practically applicable to convert geodetic coordinates into cartesian coordinates than the BPANN.

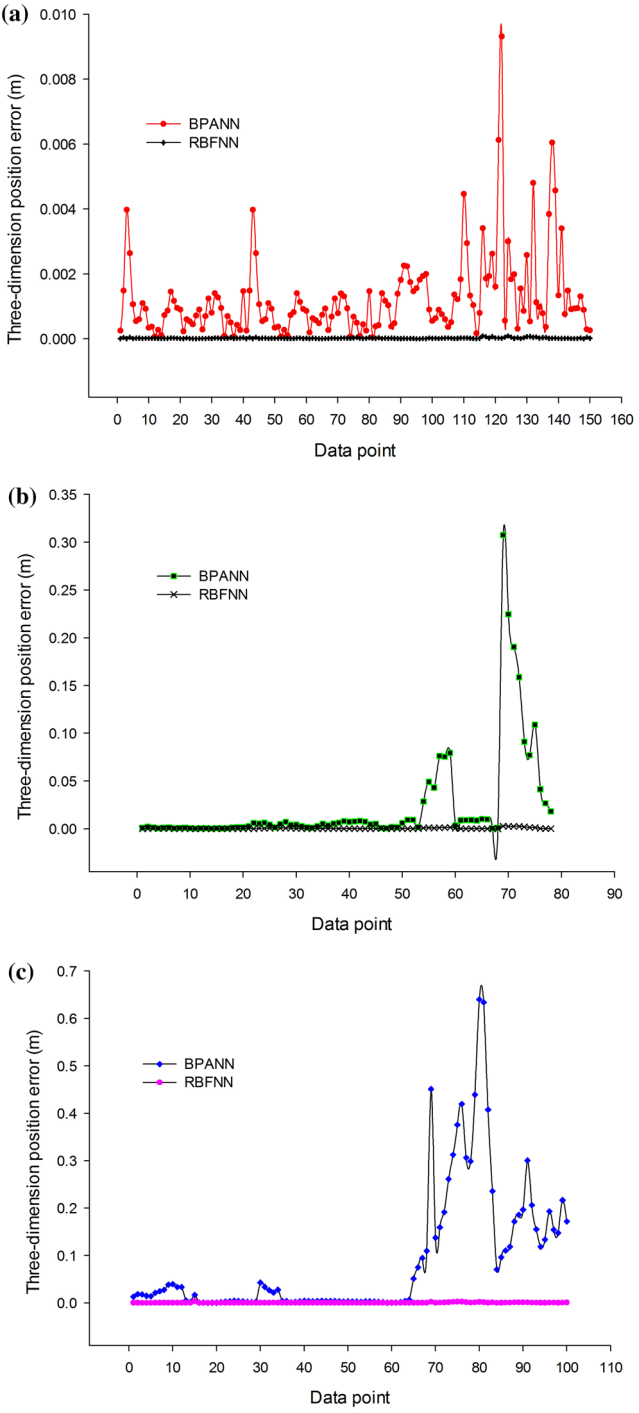

Fig. 14 Three-dimensional position error for **a** training, **b** validation and **c** testing data

Table 4 Model efficiency criteria

Dataset	BPANN			RBFNN		
	X (%)	*Y* (%)	*Z* (%)	*X* (%)	*Y* (%)	*Z* (%)
Training						
REC	99.99999	99.99994	99.99999	99.99999	99.99999	99.99999
LM	99.99931	99.99784	99.99989	99.99979	99.99998	99.99998
Validation						
REC	99.99999	99.9989	99.9999	99.9999	99.9999	99.9999
LM	99.99733	99.9556	99.9989	99.9977	99.99925	99.9998
Testing						
REC	99.99999	99.9959	99.9999	99.9999	99.9999	99.9999
LM	99.97756	99.8083	99.9978	99.99674	99.9992	99.9996

The MAE (Table 3), on the other hand, was used to identify the variation in error by providing a measure on how close the BPANN and RBFNN predictions in training, validation and testing data are to the measured data. In conformance with the MAE results in Table 3, it can be stated that the RBFNN produced promising related outputs than the BPANN. The computed NSR (Table 3) for the BPANN and RBFNN models indicate acceptable outcome signifying the occurrence of less random errors in predicting cartesian coordinates. Consequently, model performances with respect to computing errors were found to be satisfactory.

5.4 Model Efficiency Based Statistics

Model efficiency (ME)-based indicators of Legates and McCabe index (LM) and relative error correction (REC) (Sect. 4) were also utilized. The LM provides the degree to which the BPANN and RBFNN predictions are error free, by evaluating the accuracy of the predicted value with respect to the measured data. From the LM results (Table 4), it was discovered that 99 % of the potential error has been explained by the BPANN and RBFNN models developed for converting geodetic coordinates to cartesian coordinates. Similarly, the REC values show that the BPANN and RBFNN could predict about 99 % of the cartesian coordinates from geodetic coordinates. Therefore, both BPANN and RBFNN could practically be considered useful for surveying and mapping applications. However, in comparison, the RBFNN is the better model.

5.5 Comparing ANN and Multiple Linear Regression

Multiple linear regression (MLR) technique was performed to convert geodetic coordinates to cartesian coordinates. It is known that five geodetic parameters are considered as inputs when applying the standard forward transformation equation (Eq. 1). They are geodetic longitude (ϕ), geodetic latitude (λ), geodetic height (h), radius of curvature in the prime vertical (N) and first eccentricity squared (e^2). Although the results are not

Table 5 Statistical performance of BPANN, RBFNN and MLR models using all the data

Model	MSE			R		
	X (m)	Y (m)	Z (m)	X	Y	Z
BPANN	1.2524E−06	9.0393E−03	1.5358E−04	0.9999	0.9999	0.9999
RBFNN	2.3753E−08	2.7642E−07	3.5329E−08	0.9999	0.9999	0.9999
MLR	7.2977E−05	3.3367E−08	1.6482E−08	0.9999	0.9999	0.9999

included here, Pearson correlation analysis at 0.01 significance level (two-tailed) was performed in a step wise manner using all the 328 data sets in both X, Y and Z coordinates. This was necessary to select the most appropriate input parameters suitable for developing the MLR models. The results showed that the relationships existing between ϕ, λ, h, N, e^2 and measured X, Y, Z coordinates are statistically significant with $p \leq 0.01$. This means that the measured data sets provide enough evidence to reject the null hypothesis and accept the alternative hypothesis that "the population correlation coefficient is different from zero" (i.e., $\rho \neq 0$). Therefore, in this study ϕ, λ, h, N, e^2 were selected as input parameters in the MLR models' formulation. The MLR models developed are represented by Eqs. (33) to (35) given as follows

$$X_P = 5422381306199.340 - 3871.664\ \lambda + 890.141\ \phi + 0.995h, \\ -269.899N - 809731969723261\ e^2 \tag{33}$$

$$Y_P = 110561056524.072 + 110781.164\ \lambda - 1431.822\ \phi + 0.035\ h, \\ +15.687\ N - 16530449241434.2\ e^2 \tag{34}$$

$$Z_P = -76234661419.438 - 0.0097\ \lambda + 110982.793\ \phi + 0.092\ h \\ -12.691N + 11399951149870.5e^2. \tag{35}$$

Here, X_P, Y_P, Z_P are the predicted cartesian coordinates and ϕ, λ, h, N, e^2 are the independent variables. It is noteworthy that all the 328 data were used to form the various MLR models. Consequently, to make effective comparison with the ANN models, the 328 data sets were implemented in the already determined optimum RBFNN and BPANN structures. Moreover, this will further show how well the developed ANN models will generalize using the whole data.

From Table 5, it was uncovered that closely identical MSE values were achieved by both MLR and RBFNN models. These MSE results (Table 5) clearly depict close association with the respective prediction values of MLR and RBFNN models compared to the BPANN model. The identical R values attained by BPANN, RBFNN and MLR models show a strong positive relationship between predicted cartesian coordinates and measured cartesian coordinates. This means that BPANN, RBFNN and MLR models fit lines closely coincide with the ideal line and thus the three models have satisfactorily predicted cartesian coordinates.

In order to check the practicality of the three models, a summary of the total error obtained using BPANN, RBFNN and MLR models in its three-dimensional application is presented in Table 6. It is important to note that Eqs. (28–32) were used in obtaining the results in Table 6.

Table 6 Three-dimensional position deviation from the measured data using all the data

Model	Mean three-dimensional position error (m)	SD (m)	Minimum (m)	Maximum (m)
BPANN	0.036	0.089	1.57E−04	0.627
RBFNN	2.599E−04	5.184E−04	5.11E−07	0.004
MLR	8.486E−03	1.008E−03	5.119E−03	0.011

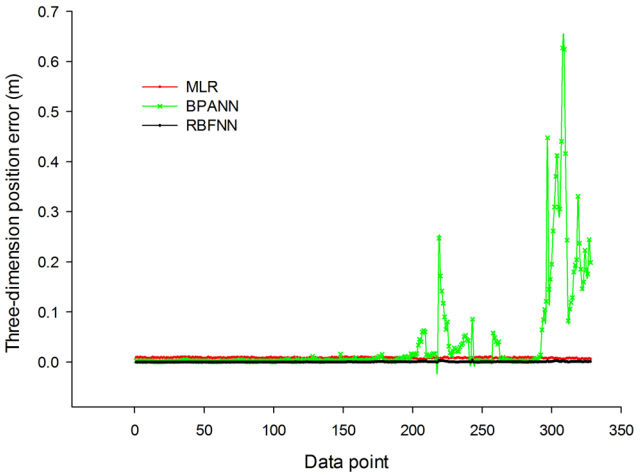

Fig. 15 Three-dimensional position error for BPANN, RBFNN and MLR models using all the data

The results in Table 6 clearly depict that the RBFNN model surpasses both the MLR and BPANN models. This is because RBFNN model produced the least mean three-dimensional positional error of 2.599E−04 m and standard deviation (SD) value of 5.184E−04 m (Table 2). Additionally, based on the maximum position error (Table 6), it can be interpreted that when the RBFNN model is applied within the study area to convert geodetic coordinates to cartesian coordinates a maximum error of 0.004 m could be achieved, whereas MLR model will give 0.011 m and BPANN model attaining 0.627 m. Therefore, in this study, it is obvious that the RBFNN structure appears to perform much better in fitting cartesian coordinate than the BPANN and MLR models. Figure 15 intuitively confirms this assertion.

6 Conclusions

Transformation of geodetic coordinates into cartesian coordinates system has been a common practice in geodesy for solving majority of astronomic, geodetic, cartographic, navigation and datum-related problems. This procedure is generally carried out using the standard forward transformation equation. However, literature has shown that little or no alternative technique has been tested to serve as a substitute for the standard forward transformation equation. The main contribution in this study is to

explore the capability of ANNs as a realistic alternative technology for converting geodetic coordinates to cartesian coordinates.

To this end, backpropagation artificial neural network (BPANN) and radial basis function neural network (RBFNN) based on the supervised learning technique as well as the multiple linear regression (MLR) technique have been presented. The findings revealed that the BPANN, RBFNN and MLR offered satisfactory prediction of cartesian coordinates. However, the RBFNN compared to BPANN and MLR showed superior stability and more accurate prediction results. It can, therefore, be proposed that the RBFNN should be used instead of the BPANN and MLR within the study area in the forward conversion of geodetic coordinates to cartesian coordinates.

On the basis of the analysis, it has been demonstrated that geodetic longitude, geodetic latitude, geodetic height, radius of curvature in the prime vertical and first eccentricity squared, combined into RBFNN could produce accurate estimates of cartesian coordinates. Therefore, this study does not only have a localized significance but will also open up more scientific discourse into the applications of ANN in geodesy within the geoscientific community.

Acknowledgments The authors would like to express their profound gratitude to the anonymous reviewers for their helpful comments.

References

Ali MH, Abustan I (2014) A new novel index for evaluating model performance. J Nat Resour Dev 4:1–9

Andrei OC (2006) 3D affine coordinate transformations. Masters of Science Thesis in Geodesy No. 3091 TRITA-GIT EX 06-004, School of Architecture and the Built Environment, Royal Institute of Technology (KTH), 100 44 Stockholm

Arbib MA (2003) The handbook of brain theory and neural networks, 2nd edn. A Bradford Book, The MIT Press, Massachusetts

Banarjee T, Singh SB, Srivastava RK (2011) Development and performance evaluation of statistical models correlating air pollutants and meteorological variables at Pantnagar, India. Atmos Res 99:505–517

Bao H, Zhao D, Fu Z, Zhu J, Gao Z (2011) Application of genetic-algorithm improved BP neural network in automated deformation monitoring. In: 7th International conference on natural computation, Shanghai-China IEEE. doi:10.1109/ICNC.2011.6022149

Cai G, Chen BM, Lee TH (2011) Unmanned rotorcraft systems. Advances in industrial control. Springer, London. doi:10.1007/978-0-85729-635-1_2

Civicioglu P (2012) Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Comput Geosci 46:229–247

Dawod GM, Mirza NM, Al-Ghamdi AK (2010) Simple precise coordinate transformations for geomatics applications in Makkah metropolitan area, Saudi-Arabia. Bridging the gap between cultures FIG working week, Marrakech Morocco, pp 18-22

Deyfrus G (2005) Neural networks: methodology and applications. Springer, Berlin

Dreiseitl S, Ohno-Machado L (2002) Logistic regression and artificial neural network classification models: a methodology review. J Biomed Inf 35(5–6):352–359

Du S, Zhang J, Deng Z, Li J (2014a) A new approach of geological disasters forecasting using meteorological factors based on genetic algorithm optimized BP neural network. Elektronika IR Elektrotechnika 20(4):57–62

Du S, Zhang J, Deng Z, Li J (2014b) A neural network based intelligent method for mine slope surface deformation prediction considering the meteorological factors. TELKOMNIKA Indones J Elect Eng 12(4):2882–2889

Featherstone WE (1997) A comparison of existing co-ordinate transformation models and parameters in Australia. Cartogr 26(1):13–26

Feltens J (2007) Vector methods to compute azimuth, elevation, ellipsoidal normal, and the cartesian (X, Y, Z) to geodetic (φ, λ, h) transformation. J Geod 82(8):493–504
Feltens J (2009) Vector method to compute the cartesian (X, Y, Z) to geodetic (φ, λ, h) transformation on a triaxial ellipsoid. J Geod 83(2):129–137
Fok SH, Iz HB (2003) A comparative analysis of the performance of iterative and non-iterative solutions to the cartesian to geodetic coordinate transformation. J Geospatial Eng 5(2):61–74
Fox D (1981) Judging air quality model performance. Bull Am Meteor Soc 62:599–609. doi:10.1175/1520-0477(1981)062<0599:JAQMP>2.0.CO;2
Fu B, Liu X (2014) Application of artificial neural network in GPS height transformation. Appl Mech Mater 501–504:2162–2165
Gao CY, Cui XM, Hong XQ (2014) Study on the applications of neural networks for processing deformation monitoring data. Appl Mech Mater 501–504:2149–2153
Ge Y, Yuan Y, Jia N (2013) More efficient methods among commonly used robust estimation methods for GPS coordinate transformation. Surv Rev 45(330):229–234
Gerdan GP, Deakin RE (1999) Transforming cartesian coordinates (x, y, z) to geographical coordinates (φ, λ, h). Aust Surv 44(1):55–63
Ghorbani MA, Khatibi R, FazeliFard MH, Naghipour L, Makarynskyy O (2015) Short-term wind speed predictions with machine learning techniques. Meteorol Atmos Phys. doi:10.1007/s00703-015-0398-9
Gullu M (2010) Coordinate transformation by radial basis function neural network. Sci Res Essays 5(20):3141–3146
Gullu M, Yilmaz M, Yilmaz I, Turgut B (2011) Datum transformation by artificial neural networks for geographic information systems applications. In: International symposium on environmental protection and planning: geographic information systems (GIS) and remote sensing (RS) applications (ISEPP), Izmir-Turkey, pp 13-19
Gurney K (2005) An introduction to neural networks. Taylor and Francis, London
Hagan MT, Menhaj MB (1994) Training feed forward techniques with the Marquardt algorithm. IEEE Trans Neural Netw 5(6):989–993
Hajian A, Ardestani EV, Lucas C (2011) Depth estimation of gravity anomalies using Hopfield neural networks. J Earth Space Phys 37(2):1–9
Hamid RS, Mohammad RS (2013) Neural network and least squares method (ANN-LS) for depth estimation of subsurface cavities case studies: Gardaneh Rokh Tunnel. Iran J Appl Sci Agric 8(3):164–171
Hartman EJ, Keeler JD, Kowalski JM (1990) Layered neural networks with Gaussian hidden units as universal approximations. Neural Comput 2(2):210–215
He-Sheng W (2006) Precise GPS orbit determination and prediction using H_∞ neural network. J Chin Inst Eng 29(2):211–219
Hoar GJ (1982) Satellite surveying. Magnavox Advanced Products and Systems Company, 2829 Maricopa Street. Torrance, California
Hornik K, Stinchcombe M, White H (1989) Multilayer feed forward networks are universal approximators. Neural Netw 2:359–366
Hu WS, Zheng DY, Nie WF (2014) Research on methods of regional ionospheric delay correction based on neural network technology. Surv Rev 46(336):167–174
Ismail S, Shabri A, Samsudin R (2012) A hybrid model of self organizing maps and least square support vector machine for river flow forecasting. Hydrol Earth Syst Sci 16:4417–4433
Kavzoglu T, Saka MH (2005) Modelling local GPS/levelling geoid undulations using artificial neural networks. J Geod 78:520–527. doi:10.1007/s00190-004-0420-3
Kecman V (2001) Learning and soft computing: a bradford book. The MIT Press, Massachusetts
Konaté AA, Pan H, Khan N, Ziggah YY (2015) Prediction of porosity in crystalline rocks using artificial neural networks: an example from the Chinese continental scientific drilling main hole. Stud Geophys Geod 59(1):113–136
Krause P, Boyle DP, Base F (2005) Comparison of different efficiency criteria for hydrological model assessment. Adv Geosci 5:89–97
Legates DR, McCabe GJ Jr (1999) Evaluating the use of "goodness of fit" measures in hydrologic and hydroclimatic model validation. Water Resour Res 35(1):233–241
Lei W, Qi X (2010) The application of BP neural network in GPS elevation fitting. In: International conference on intelligent computation technology and automation, Changsha-China, IEEE. doi:10.1109/ICICTA.2010.162
Leick A (2004) GPS satellite surveying. Wiley, Hoboken, NJ

Li X, Zhou J, Guo R (2014) High-precision orbit prediction and error control techniques for COMPASS navigation satellite. Chin Sci Bull 59(23):2841–2849
Liao DC, Wang QJ, Zhou YH, Liao XH, Huang CL (2012) Long-term prediction of the earth orientation parameters by the artificial neural network technique. J Geodyn 62:87–92
Ligas M, Banasik P (2011) Conversion between cartesian and geodetic coordinates on a rotational ellipsoid by solving a system of nonlinear equations. Geod Cartogr 60(2):145–159
Lin LS, Wang YJ (2006) A study on cadastral coordinate transformation using artificial neural network. In: Proceedings of the 27th Asian conference on remote sensing, Ulaanbaatar, Mongolia
Liu S, Li J, Wang S (2011) A hybrid gps height conversion approach considering of neural network and topographic correction. In: International conference on computer science and network technology, China IEEE. doi:10.1109/ICCSNT.2011.6182386
Lourakis MIA (2005) A brief description of the Levenberg-Marquardt algorithm implemented by levmar. Technical Report, Institute of Computer Science, Foundation for Research and Technology-Hellas
Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River, NJ, USA
Mihalache RM (2012) Coordinate transformation for integrating map information in the new geocentric European system using artificial neural networks. GeoCAD pp 1-9
Muller VA, Hemond FH (2013) Extended artificial neural networks: incorporation of a priori chemical knowledge enables use of ion selective electrodes for in-situ measurement of ions at environmentally relevant levels. Talanta 117:112–118
Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Springer, New York
Odutola AC, Beiping W, Ziggah YY (2013) Testing simple regression model for coordinate transformation by comparing its predictive result for two regions. Acad Res Int 4(6):540–550
Oil and Gas Producers (OGP) (2012) Coordinate conversions and transformations including formulas. Geomatics Guidance Note Number 7, part-2, p 5
Pan G, Zhou Y, Sun H, Guo W (2015) Linear observation based total least squares. Surv Rev 47(340):18–27
Pantazis G, Eleni-Georgia A (2013) The use of artificial neural networks in predicting vertical displacements of structures. Int J Appl Sci Technol 3(5):1–7
Park J, Sandberg IW (1991) Universal approximation using radial basis function networks. Neural Comput 3(2):246–257
Pikridas C, Fotiou A, Katsougiannopoulos S, Rossikopoulos D (2011) Estimation and evaluation of GPS geoid heights using an artificial neural network model. Appl Geomat 3:183–187. doi:10.1007/s12518-011-0052-2
Schofield W (2001) Engineering surveying: theory and examination problems for students, 5th edn. Butterworth-Heinemann, Linacre House, Jordan Hill, Oxford OX2 8DP, UK
Schuh H, Ulrich M, Egger D, Muller J, Schwegmann W (2002) Prediction of earth orientation parameters by artificial neural networks. J Geod 76:247–258
Shu C, Li F (2010) An iterative algorithm to compute geodetic coordinates. Comput Geosci 36:1145–1149
Sickle JV (2010) Basic GIS coordinates, 2nd edn. CRC Press, Taylor and Francis Group, New York
Solomon M (2013), Determination of transformation parameters for Montserrado County, Republic of Liberia. Masters Thesis, Faculty of Civil and Geomatic Engineering, College of Engineering, KNUST, Kumasi, Ghana
Sorkhabi OM (2015) Geoid determination based on log sigmoid function of artificial neural networks: a case study, Iran. J Artif Intell Electr Eng 3(12):18–24
Stopar B, Ambrožič T, Kuhar M, Turk G (2006) GPS-derived geoid using artificial neural network and least squares collocation. Surv Rev 38(300):513–524
Tieding L, Shijian Z, Xijiang C (2010) A number of issues about converting GPS height by BP neural network. In: International conference on biomedical engineering and computer science (ICBECS), Wuhan-China, IEEE doi:10.1109/ICBECS.2010.5462426
Tierra A, Dalazoana R, De Freitas S (2008) Using an artificial neural network to improve the transformation of coordinates between classical geodetic reference frames. Comput Geosci 34:181–189
Tierra AR, De Freitas SRC (2005) Artificial neural network: a powerful tool for predicting gravity anomaly from sparse data. Gravity, Geoid and Space Missions, International Association of Geodesy Symposia. Springer, Berlin Heidelberg DA. doi:10.1007/3-540-26932-0_36
Tierra AR, De Freitas SRC, Guevara PM (2009) Using an artificial neural network to transformation of coordinates from PSAD56 to SIRGAS95. Geodetic reference frames, international association of geodesy symposia. Springer 134:173-178

Tierra A, Romero R (2014) Planes coordinates transformation between PSAD56 to SIRGAS using a multilayer artificial neural network. Geod Cartogr 63(2):199–209
Turgut B (2010) A back-propagation artificial neural network approach for three-dimensional coordinate transformation. Sci Res Essays 5(21):3330–3335
Vanicek P, Steeves RR (1996) Transformation of coordinates between two horizontal geodetic datums. J Geod 70:740–745
Veronez MR, De Souza GC, Matsuoka TM, Reinhardt A, Da Silva RM (2011) Regional mapping of the geoid using GNSS (GPS) measurements and an artificial neural network. Remote Sensing 3:668–683. doi:10.3390/rs3040668
Veronez MR, Thum BA, De Souza GC (2006) A new method for obtaining geoidal undulations through artificial neural networks. In: 7th International symposium on spatial accuracy assessment in natural resources and environmental sciences, pp 306-316
Wang X (2009) Application of artificial neural network to predict short-term capital flow. In: International conference on research challenges in computer science. doi:10.1109/ICRCCS.2009.39
Wu LC, Tang X, Zhang S (2012) The application of genetic neural network in the GPS height transformation. In: IEEE 4th international conference on computational and information sciences, Chongqing-China. doi:10.1109/ICCIS.2012.317
Wu P, Yi X, Jin K (2012) A study on Chinese output of timber prediction model based on PSO-SVM. Adv Inf Sci Serv Sci (AISS) 4(2):227–233. doi:10.4156/AISS.vol4.issue2.28
Yakubu I, Kumi-Boateng B (2011) Control position fix using single frequency global positioning system receiver technique-a case study. Res J Environ Earth Sci 3(1):32–37
Yegnanarayana B (2005) Artificial neural networks. Prentice-Hall of India Private Limited, Delhi
Yilmaz I, Gullu M (2012) Georeferencing of historical maps using back propagation artificial neural network. Exp Tech 36(5):15–19
Yilmaz M (2013) Artificial neural networks pruning approach for geodetic velocity field determination. Bol Ciênc Geod 19(4):558–573
Yilmaz M, Gullu M (2014) A comparative study for the estimation of geodetic point velocity by artificial neural networks. J Earth Syst Sci 123(4):791–808
Yonaba H, Anctil F, Fortin V (2010) Comparing sigmoid transfer functions for neural network multistep ahead stream flow forecasting. J Hydrol Eng 15(4):275–283
Yu L, Danning Z, Cai H (2015) Prediction of length-of-day-using extreme learning machine. Geod Geodyn 6(2):151–159
Zaletnyik P (2004) Coordinate transformation with neural networks and with polynomials in Hungary. In: International symposium on modern technologies, education and professional practice in geodesy and related fields, Sofia, Bulgaria, pp 471-479
Zeng HE (2014) Geodetic datum transformation and inverse transformation. Appl Mech Mater 501–504: 2154–2157
Zeng H (2015) Analytical algorithm of weighted 3D datum transformation using the constraint of orthonormal matrix. Earth Planets Space 67:1–10
Zhu J (1994) Conversion of earth-centered earth-fixed coordinates to geodetic coordinates. IEEE Trans Aerosp Electron Syst 30(3):957–961
Ziggah YY, Hu Y, Odutola AC (2012) Regression models for 2-dimensional cartesian coordinates prediction: a case study at University of Mines and Technology (UMaT), Tarkwa-Ghana. Int J Comput Sci Eng Surv 3(6):61–79
Ziggah YY, Youjian H, Odutola AC, Fan DL (2013) Determination of GPS coordinate transformation parameters of geodetic data between reference datums: a case study of ghana geodetic reference network. Int J Eng Sci Res Technol 2(4):2277–9655