Acta Geodaetica et Geophysica (2021) 56:271–291
https://doi.org/10.1007/s40328-021-00336-6

ORIGINAL STUDY

Prediction of geodetic point velocity using MLPNN, GRNN, and RBFNN models: a comparative study

Berkant Konakoglu[1]

Received: 28 April 2020 / Accepted: 20 December 2020 / Published online: 31 March 2021

Abstract
The prediction of an accurate geodetic point velocity has great importance in geosciences. The purpose of this work is to explore the predictive capacity of three artificial neural network (ANN) models in predicting geodetic point velocities. First, the multi-layer perceptron neural network (MLPNN) model was developed with two hidden layers. The generalized regression neural network (GRNN) model was then applied for the first time. Afterwards, the radial basis function neural network (RBFNN) model was trained and tested with the same data. Latitude (φ) and longitude (λ) were utilized as inputs and the geodetic point velocities (V_X,V_Y,V_Z) as outputs to the MLPNN, GRNN, and RBFNN models. The performances of all ANN models were evaluated using root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). The first investigation demonstrated that it was possible to predict the geodetic point velocities by using all the components as output parameters simultaneously. The other result is that all ANN models were able to predict the geodetic point velocity with satisfactory accuracy; however, the GRNN model provided better accuracy than the MLPNN and RBFNN models. For example, the RMSE and MAE values were 1.77–1.88 mm and 1.44–1.51 mm, respectively, for the GRNN model.

Keywords Geodetic point velocity · Multi–layer perceptron neural network · Generalized regression neural network · Radial basis function neural network

1 Introduction

Geodetic networking activities in Turkey began in the early 1900s. In the 1950s, Turkey National Geodetic Network (TNGN) was established in order to create a basis for the study of mapping and cadastral works. In 1954, via eight points of the Greek and Bulgarian geodetic networks, TNGN was connected to the European datum ED50. Because Turkey is linked to a major world fault line and in terms of crustal movements is an extremely active region, Turkey National Horizontal Control Network has witnessed great damage in the

✉ Berkant Konakoglu
berkantkonakoglu@amasya.edu.tr

[1] Department of Architecture and Urban Planning, Technical Sciences Vocational School, Amasya University, 05100 Amasya, Turkey

region over time. Highly sensitive measurement and calculation methods were needed to identify deformations. On the other hand, due to rapid population growth and urbanisation, the need for infrastructure services had increased. This made it necessary to complete Turkey's digital cadastral situation accurately and as soon as possible. Therefore, there was a need for a fundamental geodetic network based on the Global Positioning System (GPS) and consisting of points of specified accuracy that sufficiently covered the country's surface. With the work carried out between 1997 and 1999 by the General Command of Mapping to realise the said objectives, the Turkish National Fundamental GPS Network–1999 (TNFGN–99) was established, consisting of nearly 600 points. In the newly formed geodetic network, three coordinate values (X, Y, Z) and their velocities (V_X, V_Y, V_Z) were calculated in the three–dimensional geocentric coordinate system at a given time. The TNFGN is within the ITRF (International Terrestrial Reference Frame) coordinate system, with a relative sensitivity of 0.1–0.01 ppm, and point position sensitivity levels of 1–3 cm. Large-scale earthquakes in 1999–2003 (e.g.: Mw = 7.5/İzmit, Mw = 7.2/Düzce, Mw = 6.1/Çankırı-Çerkeş, Mw = 6.5/Sultandağı, and Mw = 6.4/Bingöl) caused important changes in TNFGN point locations in the earthquake zones. For this reason, TNFGN was updated by carrying out GPS measurements.

Turkey is located in the Alpine–Himalayan earthquake zone and in the collision zone with the Eurasian tectonic plate of the African and Arabian plates (McClusky et al. 2000). Considering the magnitude of the deformation induced and the active tectonic structure of Turkey and its periphery, it is of great importance for the survival and improvement of the four-dimensional (X, Y, Z, and time) designed TNFGN to determine the time-dependent coordinate changes of the on-site points with high accuracy. Time-dependent changes in point coordinates generally caused by tectonic plate movements include pre-earthquake (inter-seismic), earthquake moment (co-seismic), and post-seismic effects (Demir and Açıkgöz 2000). Point velocities must be determined with high accuracy in order to establish these time-dependent effects. For the survival and improvement of TNFGN, GPS measurements are carried out periodically and these measurements are evaluated and combined into a specified reference epoch. The TNFGN coordinates and velocities are calculated in the ITRF system (Aktuğ et al. 2011). The velocities of TNFGN points are calculated by estimating two or more repetitive GPS measurements or, in cases without repeated GPS measurements, the speeds of other TNFGN points (Kurt and Deniz 2010).

In order to ensure the standard unity of large-scale mapping, the regulation for the construction of large-scale maps was published in 1988. However, in parallel with the increase of GPS usage in geodetic applications and large-scale map activities, it was updated in 2005 and the system of calculating geodetic point velocities was adopted. In order to compress the TNFGN, satellite positioning techniques were used to shift the coordinates of the points in the measurement epoch to a specified reference epoch. The point velocities to be used for this process are calculated by interpolation from TNFGN or high-grade compression point velocities, but the method used is not specified. The Kriging (KRIG) interpolation method is the best-known and most used technique. This method has been used in geodetic velocity modelling by several researchers (Majdański 2012; Kierulf et al. 2013; Bogusz et al. 2014; Ching and Chen 2015; Li et al. 2019). Many studies have been also conducted to investigate the geodetic velocity fields for different regions using GPS observations (Aktuğ et al. 2013; Gülal et al. 2013; Müller et al. 2013; Farolfi and Del Ventisette 2016; Ito et al. 2019; Poyraz et al. 2019).

Artificial neural networks (ANNs) belong to the family of artificial intelligence (AI) techniques widely used in many fields of geoscience. The ANN is utilized for modelling, optimization and estimation in order to solve complicated problems and generally presents satisfactory results compared to other conventional techniques. In recent years, the use of ANNs in the field of geodesy has been widely adopted and implemented. Its suitability as an alternative technique to traditional methods in solving most geodetic problems has been investigated. Many researchers have used the ANN in the past for solving different geodetic problems of coordinate transformation (Zaletnyik 2004; Lin and Wang 2006; Tierra et al. 2008; Gullu 2010; Tierra and Romero 2014; Konakoğlu and Gökalp 2016; Konakoglu et al. 2016; Ziggah et al. 2016a, b, 2019; Elshambaky et al. 2018; Cakir and Konakoglu 2019), modelling ionospheric TEC (Hernandez-Pajares et al. 1997; Cander 1998; Habarulema et al. 2007; Maruyama 2008; Akhoondzadeh 2014; Huang and Yuan 2014; Tebabal et al. 2018; Inyurt and Sekertekin 2019), geoid determination (Hu et al. 2004; Kavzoglu and Saka 2005; Stopar et al. 2006; Lin 2007; Veronez et al. 2011; Erol and Erol 2013; Cakir and Yilmaz 2014; Kaloop et al. 2018a), earth orientation parameter determination (Schuh et al. 2002; Wang et al. 2008; Liao et al. 2012; Lei et al. 2015), gravity anomaly prediction (Hajian et al. 2011; Tierra and De Freitas 2005; Pereira et al. 2012), and noise reduction in GNSS signals (Mosavi 2006; Kaloop and Hu 2015; Kaloop et al. 2018b).

In addition, one of the geodetic problems is the prediction of the geodetic point velocities. A few studies utilizing the ANN in this context have been conducted in the past with successful results. For example, Güllü et al. (2011) developed the multi-layer perceptron neural network (MLPNN) model to solve this problem using the TNFGN dataset. The geodetic point velocities have also been estimated using different interpolation methods. On the basis of statistical analysis, the MLPNN model is highly preferred compared to traditional interpolation models. Yilmaz and Gullu (2014) applied two different types of ANN models for geodetic point velocity estimation. They constructed ANN models using the MLPNN and the radial basis function neural network (RBFNN). To evaluate the performance of the ANNs, the Kriging (KRIG) method was used to interpolate the velocities. The optimal ANN was developed with two neurons in the input layer and one neuron in the output layer. The input parameters were the geographic coordinates (latitude and longitude), and the velocity component of the geodetic point (V_X or V_Y or V_Z) was selected separately as the output parameter. A single hidden layer was used in the ANN models. The MLPNN and RBFNN models of the ANN successfully estimated the geodetic point velocities for regional geodetic networks. The performance results predicted using the MLPNN model were better than those obtained using the KRIG and RBFNN models. A literature review indicates that an ANN model can be used as an alternative to conventional interpolation methods for predicting geodetic point velocities. However, only the MLPNN and RBFNN models were used for this purpose. Application of the generalized regression neural network (GRNN) model was not conducted in the past. In this present work, in addition to the MLPNN and RBFNN models, the GRNN model was used to predict geodetic point velocities and to find the appropriate model on the basis of a comparative study. This is the novelty of this research. To achieve this objective, 238 points and corresponding velocities belonging to the TNFGN were utilized. In order to predict the geodetic point velocities, latitude (φ) and longitude (λ) were selected as input parameters, and all geodetic point velocity components (V_X,V_Y, and V_Z) were selected simultaneously as output parameters.

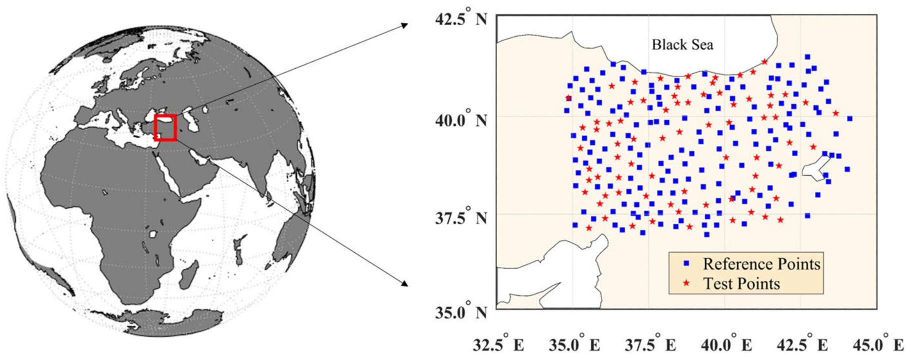

Fig. 1 Location of the study area and distribution of the reference and test points

Table 1 Statistical parameters of the dataset

Data	Unit	Average	Minimum	Maximum	Standard deviation
Latitude	°	39.27002177	36.97123797	41.47965261	1.24861387
Longitude	°	38.91313954	34.81386392	44.12488748	2.54054889
V_X	(mm/year)	–22.9	–38.1	–7.9	6.2
V_Y	(mm/year)	2.7	–37.6	16.7	7.7
V_Z	(mm/year)	10.8	–1.8	34.4	5.1

° degree, mm/year millimetre per year

2 Study area and applied data

This study was conducted in central and eastern Anatolian regions of Turkey in an area located at 34° 48′ E–44° 7′ E longitudes and 36° 58′ N–41° 25′ N latitudes (Fig. 2). The study area included a total of 238 points belonging to TNFGN. Out of the 238 points, 166 (~70%) were randomly selected as the references and the remaining 72 points (~30%) were used as the test points to evaluate the performance of the ANN models. The distribution of the reference and test points within the study area is shown in Fig. 1. Here, the blue squares and red asterisks denote the reference points and test points, respectively.

The statistical characteristics of the dataset related to the studied points including the average, minimum, maximum, and standard deviation values are listed in Table 1.

To improve the computational speed and obtain more accurate results, the input and output sample data must be normalized to a range of [– 1,1], [0,1], or another scaling criterion before developing the ANN models. In this study, the data was normalized between – 1 and 1 using Eq. (1).

$$Y_{\text{normalized}} = Low_{value} + \left(High_{value} - Low_{value}\right)\frac{Y_i - Y_{min}}{Y_{max} - Y_{min}} \tag{1}$$

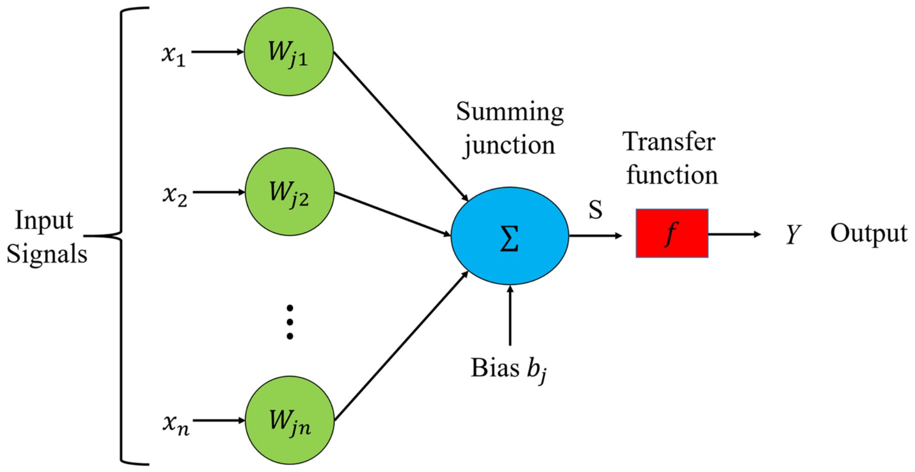

Fig. 2 Structure of the multi-layer perceptron neural network (MLPNN)

3 Artificial neural network (ANN)

McCulloch and Pitts (1943) proposed the artificial neural network (ANN) for the first time. The ANN is a computational paradigm which behaves like the human brain, and has been successfully used to find solutions to complex problems that are difficult to solve by other known methods. This is the main advantage of the ANN method. Many types of ANN models have been introduced and successfully applied to date. In this study, the three types of ANN models employed to predict geodetic point velocities were the MLPNN, GRNN, and RBFNN. The following sections present a brief description of these ANN models.

3.1 Multi-layer perceptron neural network (MLPNN)

The MLPNN is known as the most common type of ANN (Haykin 1999) and its structure can be represented as in Fig. 2. The MLPNN has three layers: the input layer, the hidden layer, and the output layer. Each neuron receives the information from other neurons and transmits it to the following layers. Interconnected processing neurons combine together and form an ANN. The output of each neuron is the product of weighted inputs. The sum of the weighted inputs formed by neurons is as given in Eq. (2).

$$X = \left(\sum_{i=1}^{n} w_{ij} x_i \right) + b_j \tag{2}$$

where w_{ij} are the interconnecting weights of input data x_i, b_j is the bias for the neuron, and n is the number of input data. This sum X then passes through transfer function F, which generates the output, as shown in Eq. (3).

$$Y = F(X) = F\left[\left(\sum_{i=1}^{n} w_{ij} x_i \right) + b_j \right] \tag{3}$$

Hidden and output layers generally have either a linear or non–linear transfer function. The commonly used nonlinear transfer function, as expressed in Eq. (4), is the sigmoid function, whose output lies between 0 and 1.

$$F(X) = \frac{1}{1 + e^{-x}} \tag{4}$$

If the input and output layers have negative values, then the tansig transfer function is used, which is expressed as Eq. (5).

$$F(X) = \frac{1 - e^{-2X}}{1 + e^{-2X}} \tag{5}$$

3.2 Generalized regression neural network (GRNN)

The GRNN was proposed by Specht (1991) as a type of radial basis function neural network (RBFNN) and a universal approximator for smooth functions that can solve any smooth function approximation problem (Park and Sandberg, 1991). A schematic of the GRNN is depicted in Fig. 3.

As seen from Fig. 3, the GRNN consists of four layers (Patterson 1996): the input layer, pattern layer, summation layer and output layer. The first layer, referred to as input, picks up information and conveys it to the pattern layer. The pattern layer is the second layer, which is connected to the summation layer. The output of the pattern layer passes through the summation layer. This layer consists of two summations, namely,

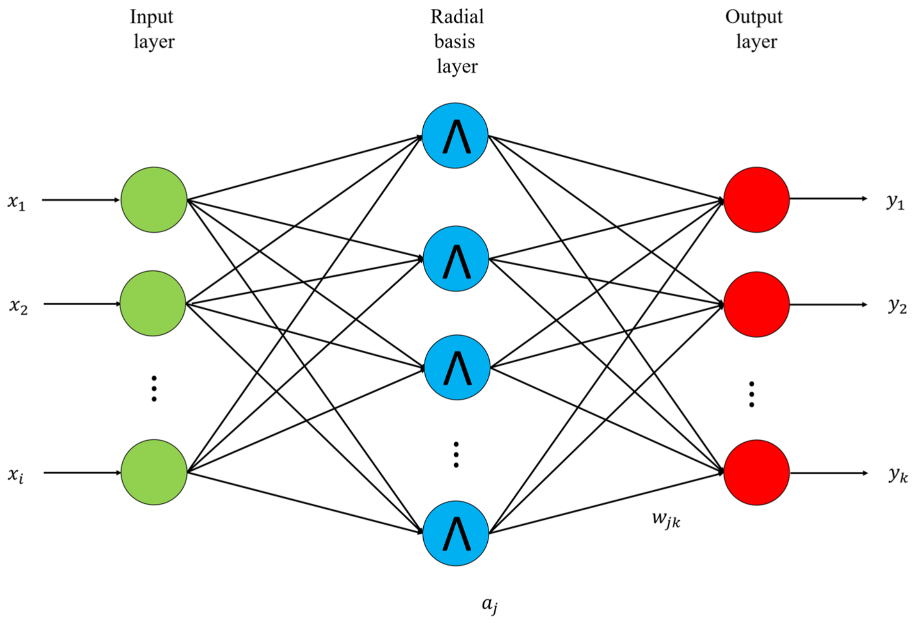

Fig. 3 Basic structure of GRNN

S-summation and D-summation neurons, respectively. The sum of the weighted pattern outputs is calculated using Eq. (6).

$$S = \sum_{i=1}^{n} Y_i exp\left(-\frac{\left(X - X_i\right)^T \left(X - X_i\right)}{2\sigma^2} \right) \tag{6}$$

where Y_i is the weight connecting the i th neuron in the pattern layer to the summation layer. The unweighted pattern output is calculated using Eq. (7).

$$D = \sum_{i=1}^{n} exp\left(-\frac{\left(X - X_i\right)^T \left(X - X_i\right)}{2\sigma^2} \right) \tag{7}$$

The results of the sum calculated in the summation layer are then transmitted to the output layer. The output Y, can be derived from the function in Eq. (8).

$$Y(X) = \frac{S}{D} = \frac{\sum_{i=1}^{n} Y_i exp\left(-\frac{(X-X_i)^T (X-X_i)}{2\sigma^2} \right)}{\sum_{i=1}^{n} exp\left(-\frac{(X-X_i)^T (X-X_i)}{2\sigma^2} \right)} \tag{8}$$

where σ is the spread parameter (also called a smoothing parameter). This is the only network parameter to be regulated and it determines the generalization performance of the GRNN. The trial and error procedure was generally used to determine the optimal spread parameter. The main advantage of this ANN compared to the other existing ANNs is that it is does not require an iterative training procedure. It needs only one–pass learning to achieve optimal prediction performance.

3.3 Radial basis function neural network (RBFNN)

The RBFNN is a type of feed-forward neural network that was addressed by Broomhead and Lowe (1988). The architecture of the RBFNN consists of three layers, namely, the input layer, hidden layer, and output layer, as presented schematically in Fig. 4. The RBFNN has the advantage of not suffering from local minima and having a simple structure and fast training procedure compared to the MLPNN.

The hidden layer of the RBFNN model is a radial basis transfer function and the linear function is used at the output layer. Among many radial basis functions, the Gaussian function is preferred as the transfer function in the hidden layer, and is defined as in Eq. (9) (Haykin 1999).

$$a_j(x) = exp\left(\frac{\|x_i - c_j\|}{2\sigma_j^2} \right) \tag{9}$$

where $a_j(x)$ is the centre of the basis function, σ_j is the spread of the j_{th} neuron in the hidden layer, $\|x_i - c_j\|$ is the radial distance between x_i and the centre of the RBF unit. The operation of the output layer is linear, as given in Eq. (10).

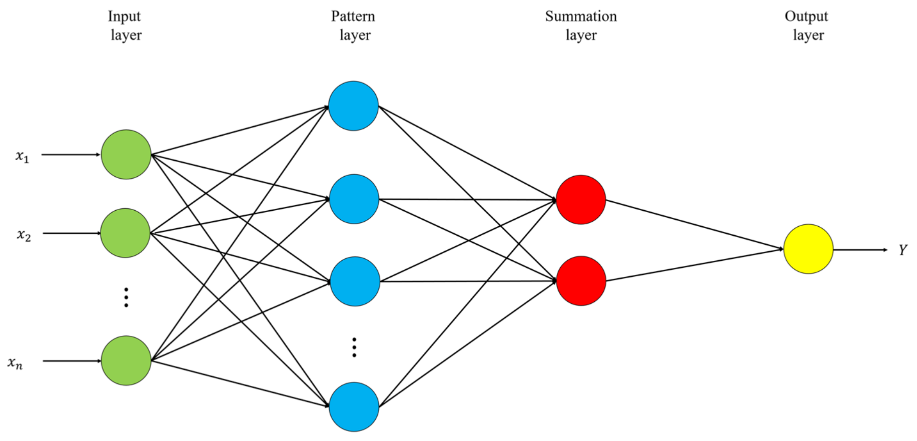

Fig. 4 Schematic diagram of RBFNN

$$y_k(x) = \sum_{j=1}^{n} w_{jk} a_j(x) + b_k \tag{10}$$

where y_k is the k_{th} output unit for the input vector x, w_{jk} is the weight connection between the k_{th} output unit and the j_{th} hidden layer unit, and b_k is the bias.

3.4 Performance metrics

The outcomes of velocity prediction using the MLPNN, GRNN, and RBFNN models were evaluated based on the statistical metrics of root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2).

$$\mathrm{RMSE} = \left(\frac{1}{n} \sum_{i=1}^{n} \left(X_{O,i} - X_{P,i} \right)^2 \right)^{1/2} \tag{11}$$

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| X_{O,i} - X_{P,i} \right| \tag{12}$$

$$\mathrm{R}^2 = \left(\sum_{i=1}^{n} \left(X_{O,i} - \bar{X}_O \right) \left(X_{P,i} - \bar{X}_P \right) / \sqrt{ \sum_{i=1}^{n} \left(X_{O,i} - \bar{X}_O \right)^2 \sum_{i=1}^{n} \left(X_{P,i} - \bar{X}_P \right)^2 } \right)^2 \tag{13}$$

where n is the number of data, $X_{O,i}$ and $X_{P,i}$ are the observed and predicted values, respectively, and $\bar{X}_O$ is the mean observed value.

4 Results and discussion

In the current study, the MLPNN, GRNN, and RBFNN models were developed in a MATLAB software environment. In this section, the obtained results are presented separately.

4.1 MLPNN model development and results

The choice of the number of hidden layers is an important step in the operation of the ANN and for that there is no theoretical guidance. It has been proven that a single hidden layer is sufficient to approximate any continuous function (Hornik et al. 1989). Moreover, using more than one hidden layer can lead to a large number of local minima and make training difficult. However, the MLPNN that used a single (one) layer did not give accurate training results. Thus, in this study, double (two) hidden layers were utilized to construct the MLPNN model. Due to the low accuracy of the model, the results derived with a single hidden layer are not given in this article.

At the start of the training phase, both weights and biases were initialized with random values. Hence, each MLPNN model was retrained 100 times with the same training dataset to avoid the effects of different initial weights and biases. The mean of the output values from these models is then assumed as the final value of the MLPNN model. The hyperbolic tangent sigmoid function (tansig) and logarithmic sigmoid function (logsig) were used in the hidden layers, whereas the pure linear function (purelin) was used in the output layer. Before the training phase, the most appropriate training function must be chosen for the problem. Different back-propagation training algorithms including the Levenberg–Marquardt (LM), Scaled Conjugate Gradient (SCG), Bayesian Regularization (BR), and Gradient Descent with Momentum and Adaptive Learning Rate (GDX) were used for training the MLPNN. Among the four training algorithms for geodetic point velocity prediction, the MLPNN trained with the BR training algorithm provided the best performance in the training phase, with RMSE about 4 mm. The MLPNN trained with LM and SCG training algorithms yielded lower performance in training phase, with RMSE of about 6 mm and 7 mm, respectively, for all velocity components. The weakest model is the MPLNN trained with the GDX training algorithm with an RMSE of about 8 mm for all velocity components. To have a complete conclusion, the models' performances were assessed on the testing dataset. Similar results to the training phases were obtained. The MLPNN trained with BR training algorithm was the best model in a comparison of the other models. The corresponding performance values of the MLPNN trained with BR training algorithm also found with an RMSE of about 2 mm for all velocity components. Whereas, MLPNN trained with LM training algorithm, MLPNN trained with SCG training algorithm, and MLPNN trained with GDX training algorithm proved lower performances, with RMSE 4 mm, 5 mm, and 6 mm respectively, for all velocity components. From the analysis of the results, it is obvious that for MLPNN, BR training algorithm is suitable for predicting the geodetic point velocities. The detail results about the MLPNN trained with BR training algorithm are given in Table 1. The BR is a training algorithm that updates the weight and bias values according to Levenberg–Marquardt optimization. It minimizes a combination of squared weights and biases, and then determines the correct combination so as to produce a network that generalizes well. To improve generalization ability of the network, the regularized training objective function $S(w)$ is denoted as:

$$S(w) = \alpha E_W + \beta E_D; \quad E_W = \sum_{i=1}^{m} w_i^2; \quad E_D = \sum_{i=1}^{n} \left(Y_i - T_i\right)^2 \tag{14}$$

where α and β are objective function parameters (regularization parameters), E_W is the sum of squared network weights, E_D is the sum of squared network errors,m is the number of weights,

n is the number of input and output examples of the training dataset D, and T is the target value.

In Bayesian framework, the weights of the network are considered to be random variables. According to Bayes' rule the probability distribution of the weights can be written as:

$$P(w|D, \alpha, \beta, M) = \frac{P(D|w,\beta,M)P(w|\alpha,M)}{P(D|\alpha,\beta,M)} \tag{15}$$

where M is the network model and architecture, and w is the vector of network weights. $P(D|w,\beta,M)$ is the likelihood function, which shows the probability of the data occurring.$P(w|\alpha,M)$ is the prior density, which represents our knowledge of the weights.$P(D|\alpha,\beta,M)$ is a normalization factor, which guarantees that the total probability is 1. If the noise in the training set data is Gaussian and that the prior distribution for the weights is Gaussian, the probability densities can be obtained by

$$P(w|\alpha, M) = \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W) \tag{16}$$

$$P(D|w, \beta, M) = \frac{1}{Z_D(\beta)} \exp(-\beta E_D) \tag{17}$$

where $Z_W(\alpha) = (\pi/\alpha)^{N/2}$ and $Z_D(\beta) = (\pi/\beta)^{n/2}$

Substituting the expressions for the prior probability and the likelihood function into (15) gives.

$$P(w|D, \alpha, \beta, M) = \frac{1}{Z_{S(w)}(\alpha,\beta)} \exp(-S(w)) \tag{18}$$

The objective function parameters α and β determine the complexity of the model M. Now we again apply Bayes' rule to.

$$P(D|\alpha, \beta, M) = \frac{P(D|\alpha,\beta,M)P(\alpha,\beta|M)}{P(D|M)} \tag{19}$$

From Eqs. (16) and (17), it follows.

$$P(\alpha, \beta|D, M) = \frac{Z_{S(w)}(\alpha,\beta)}{Z_W(\alpha)Z_D(\beta)} \tag{20}$$

In Eq. (20), we already know $Z_W(\alpha)$ and $Z_D(\beta)$. Since the objective function has the shape of a quadratic in the small area surrounding the minimum point of the posterior density w^{MP}, where the gradient is zero. Thus, we can estimate $Z_{S(w)}(\alpha,\beta)$ by Taylor series expansion. For solving the normalizing constant, we obtain

$$Z_{S(w)}(\alpha,\beta)^{\text{v}} \approx (2\pi)^{N/2}\det((H^{MP})^{-1})^{\frac{1}{2}}\exp(-S(w^{MP})) \tag{21}$$

where H is the Hessian matrix of the objective function.

$$H = \alpha\nabla^2 E_W + \beta\nabla^2 E_D \tag{22}$$

Substituting $Z_{S(w)}$ in Eq. (21) by Eq. (20), taking the derivative with respect to each of the log of Eq. (20) and set them equal to zero, we can find the optimal values for α and β at w^{MP}.

$$\alpha = \gamma/2E_W(w^{MP}) \tag{23}$$

$$\beta = (n-\gamma)/2E_D(w^{MP}) \tag{24}$$

where γ is the number of effective parameters

$$\gamma = N - 2\alpha^{MP}trace^{-1}(H^{MP}) \tag{25}$$

where N is the number of parameters in the network.

According to MacKay (1992) and Foresee and Hagan (1997), the iterative procedure is as follows: (1) Initialize α, β and the weights. (2) Take one step of the LM training algorithm to find the weights that minimize the objective function $S(w)$. (3) Compute γ using the Gauss-newton approximation to Hessian matrix in the Levenberg–Marquardt training algorithm and new values for α and β. (4) Iterate steps 2 to 3 until convergence.

In the past, many approaches were suggested for calculating the number of neurons in the hidden layer. However, determining the number of hidden neurons was achieved by the trial–and–error approach. The RMSE was used as a criterion to select the optimum number of neurons. Starting from two hidden neurons, the number of hidden neurons was increased by one in each trial until the required accuracy was achieved. The value of 7 for both hidden neurons was selected as the optimum case. Thus, the optimal MLPNN structure was determined as [2:7:7:3]. The performance of the MLPNN model for the training and testing phases in terms of the RMSE, MAE, and R^2 is given in Table 2.

Table 2 shows that all velocity components gave similar RMSE and MAE values of about 4 mm and 3 mm, respectively, for the training phase. The same table indicates that the RMSE and MAE values belonging to all velocity components are about 2.5 mm and 2 mm, respectively, for the testing phase. The MLPNN model gave a high R^2 value

Table 2 Performance statistics of the MLPNN model in the training and testing phases

Phase	V_X	V_Y	V_Z
Training			
RMSE (mm)	3.52	4.71	4.23
MAE (mm)	2.61	2.82	2.99
R^2	0.71436	0.66787	0.42005
Testing			
RMSE (mm)	2.57	2.28	2.39
MAE (mm)	1.92	1.75	1.95
R^2	0.77892	0.89700	0.61488

for the testing phase, whereas the R^2 value for training phase was found to be low. The R^2 value for the testing phase was close to 0.9 only for the V_Y component, while the R^2 values of V_X and V_Z remained below 0.9. The scatter diagrams of the velocity values predicted by the MLPNN model versus the observed velocities and the residuals produced by the MLPNN model in the testing phase is also shown in Fig. 5.

In the V_X and V_Z components, the intersection rate was far from to 0, although the slope rate was close to 1. Nevertheless, the slope and intercept rates were very close to 1 and 0 in the V_Y component. According to the obtained results, the MLPNN model

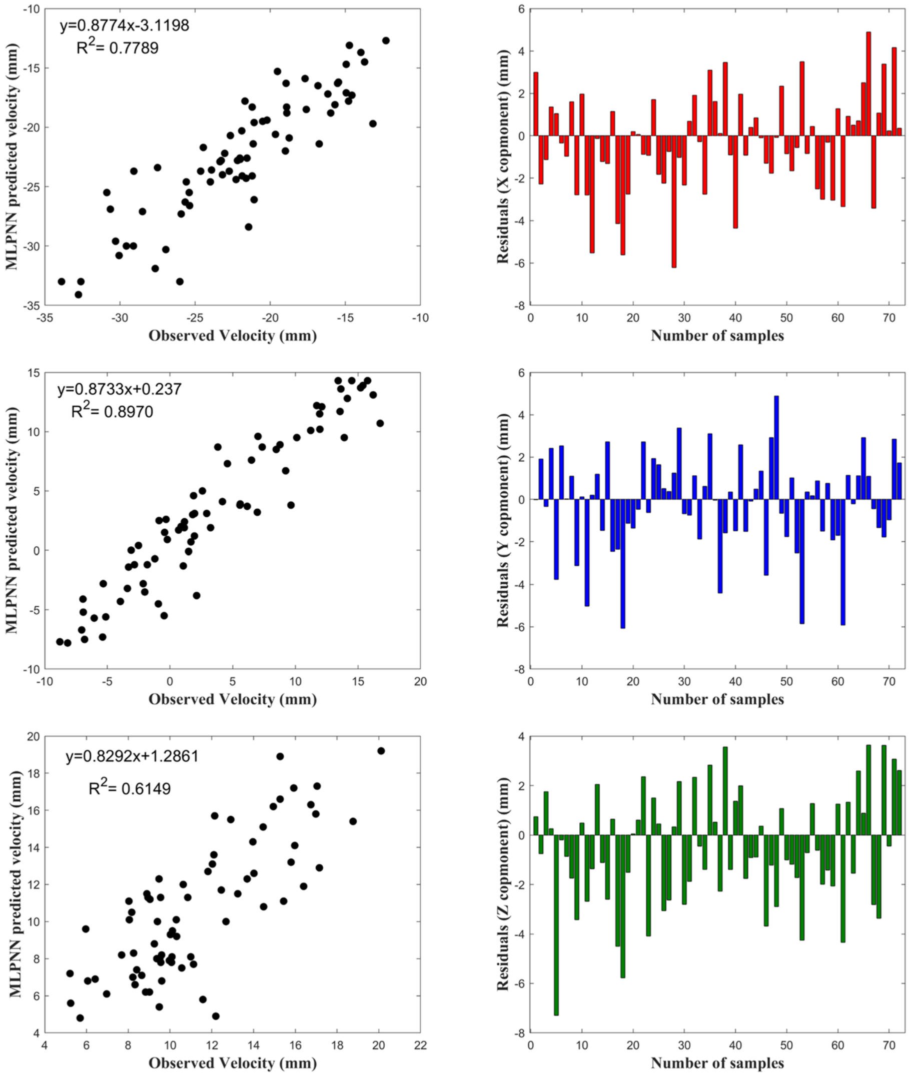

Fig. 5 Scatter diagrams of the predicted and observed velocity values (left) and residuals obtained from the developed MLPNN model for testing points (right)

developed for predicting the geodetic point velocities gave acceptable results despite the reasonable fitting of the curve.

4.2 GRNN model development and results

The GRNN was employed to construct the second ANN model. One of the most important steps in training a GRNN model is to select the best possible spread parameter because this value influences the efficacy of the developed GRNN. The GRNN was trained with different spread values ranging from 0.01 to 1. The optimum value of the spread parameter was determined according to the RMSE. The point where the error decreased and began to increase again was chosen as the most appropriate spread parameter. Because of the lowest RMSE at a spread parameter of 0.18, the predicted results were satisfactory. In order to reveal the performance of the GRNN model, the training and testing results in terms of the RMSE, MAE, and R^2 are tabulated in Table 3.

According to the derived results of the GRNN model, based on the training dataset, the RMSE for velocity components ranged from 3.47 to 4.22 mm, and the MAE value obtained was about 3 mm per component. Considering the testing results, the GRNN model provided higher accuracies for all velocity components, with RMSE values of 1.88 mm, 1.81 mm, and 1.77 mm, respectively. The MAE ranged from 1.44 to 1.51 mm with regard to testing results. The GRNN model was similar to the MLPNN model, based on the testing dataset, and yielded a high R^2 value, whereas the R^2 value was low for the training phase. The R^2 value for the testing phase was about 0.9 for the V_x and V_Y velocity components. For the V_Z component, the R^2 value was found as 0.8. Figure 6 shows the scatter diagrams of the velocity values predicted by the GRNN model versus the observed velocities and the residuals produced by the GRNN model in the training phase.

For the V_Z component, the slope rate was close to 1, whereas the intersection rate was about 2. The rates of slope and intercept were very close to 1 and 0 for components V_X and V_Y. That is to say, the predicted velocity values were near to the real velocity values of the cleaning width. Similar to the MLPNN model results, the fitting curves of the V_Y and V_Z components gave less sensitive results compared to the V_X component. In general, based on the obtained results, the GRNN model developed for predicting the geodetic point velocities is reasonably good.

Table 3 Performance statistics of the GRNN model in the training and testing phases

Phase	V_X	V_Y	V_Z
Training			
RMSE (mm)	3.47	4.74	3.92
MAE (mm)	2.80	3.04	2.93
R^2	0.73208	0.67801	0.52686
Testing			
RMSE (mm)	1.88	1.81	1.77
MAE (mm)	1.51	1.47	1.44
R^2	0.87061	0.93299	0.78032

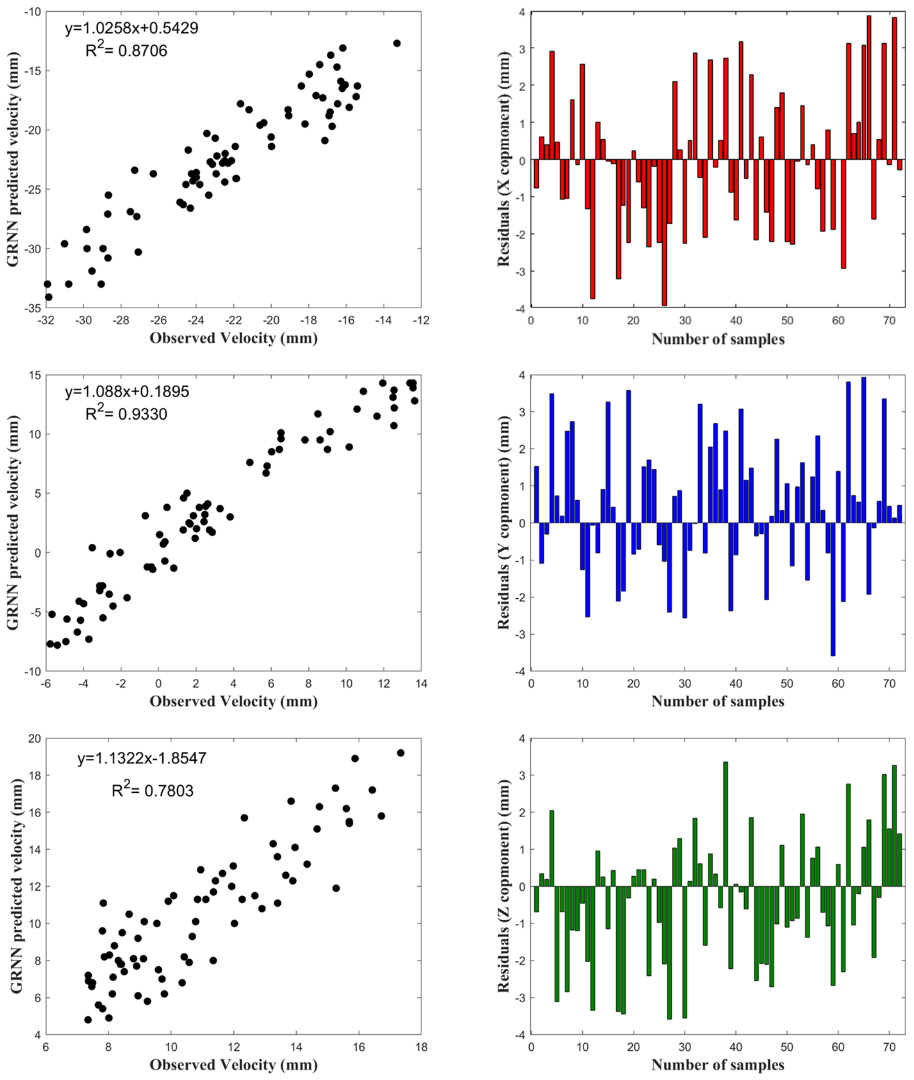

Fig. 6 Scatter diagrams of the predicted and observed velocity values and residuals obtained from the developed GRNN model for testing points

4.3 RBFNN model development and results

The RBFNN was employed to construct the third ANN model and the same parameters were used. The spread parameter was varied to achieve a lower RMSE value. After several trials, it was found that the RBFNN model showed good results at a spread of 1.5. To achieve good prediction, 60 neurons were taken in the hidden layer of the RBFNN model. The prediction results of RMSE, MAE, and R^2 for the training and testing phases are given in Table 4.

Table 4 Performance statistics of the RBFNN model in the training and testing phases

Phase	V_X	V_Y	V_Z
Training			
RMSE (mm)	3.57	4.78	4.11
MAE (mm)	2.83	3.00	3.05
R^2	0.70488	0.65500	0.44845
Testing			
RMSE (mm)	2.33	2.34	2.34
MAE (mm)	1.83	1.90	1.83
R^2	0.82344	0.88553	0.62057

According to the derived results of the RBFNN model, based on the training dataset, the RMSE for all velocity components ranged from 3.57 to 4.78 mm, while the RMSE for all velocity components was about 2.34 mm based on the testing dataset. Based on the training dataset, the MAE value was about 3 mm per component, while the MAE value was about 2 mm per component for the testing dataset. Similar to the MLPNN and GRNN models, the RBFNN model gave a low R^2 value for the testing dataset, whereas a high R^2 value was found for the training dataset. The R^2 value for the testing phase was about 0.8 for both the V_X and V_Y components. The R^2 value was about 0.6 for component V_Z. Figure 7 shows the scatter diagrams of the predicted values of velocities via the RBFNN model versus the observed velocities and the residuals produced by the RFBNN model in the training phase.

The RBFNN gave slope and intersection rates very close to the MLPNN results (Fig. 7). In components V_X and V_Z, the slope rate was close to 1, whereas the intersection rate was about 3. As with both the MLPNN and GRNN methods, the slope and intercept rates for the V_Y component were very close to 1 and 0. As a result, it was understood that the RBFNN was likely capable of reasonably predicting the velocity values within a broad range of data despite the poor fitting of the curve.

4.4 Comparisons of models

Performances of the MLPNN, GRNN, and RBFNN models were compared with each other and the appropriate ANN model was found for predicting the geodetic point velocities. The comparison of results from the three different ANN models, based on the RMSE, MAE, and R^2, is given in Table 5. This table shows that the RMSE values of the MLPNN, GRNN, and RBFNN models were approximately 2.41 mm, 1.82 mm, and 2.34 mm, respectively. According to the RMSE results of the GRNN model considering all components of geodetic point velocities, the RMSE values were below 2 mm. Similarly, the MAE values were below 2 mm, at approximately 1.87 mm, 1.47 mm, and 1.85 mm, respectively. The R^2 values of the MLPNN, GRNN, and RBFNN models were approximately 0.76360, 0.86131, and 0.77651, respectively. In light of these comparisons, it is clear that the GRNN model produced the best accuracy in all the geodetic point velocity components with respect to RMSE, MAE, and R^2. As can be observed from Table 5, after the GRNN model, the RBFNN model performed better than the MLPNN model.The results shown in bold in the Table 5 indicate the best values.

The comparisons of the observed results with the predicted results of the MLPNN, GRNN, and RBFNN models, and the velocity residuals obtained from all the ANN

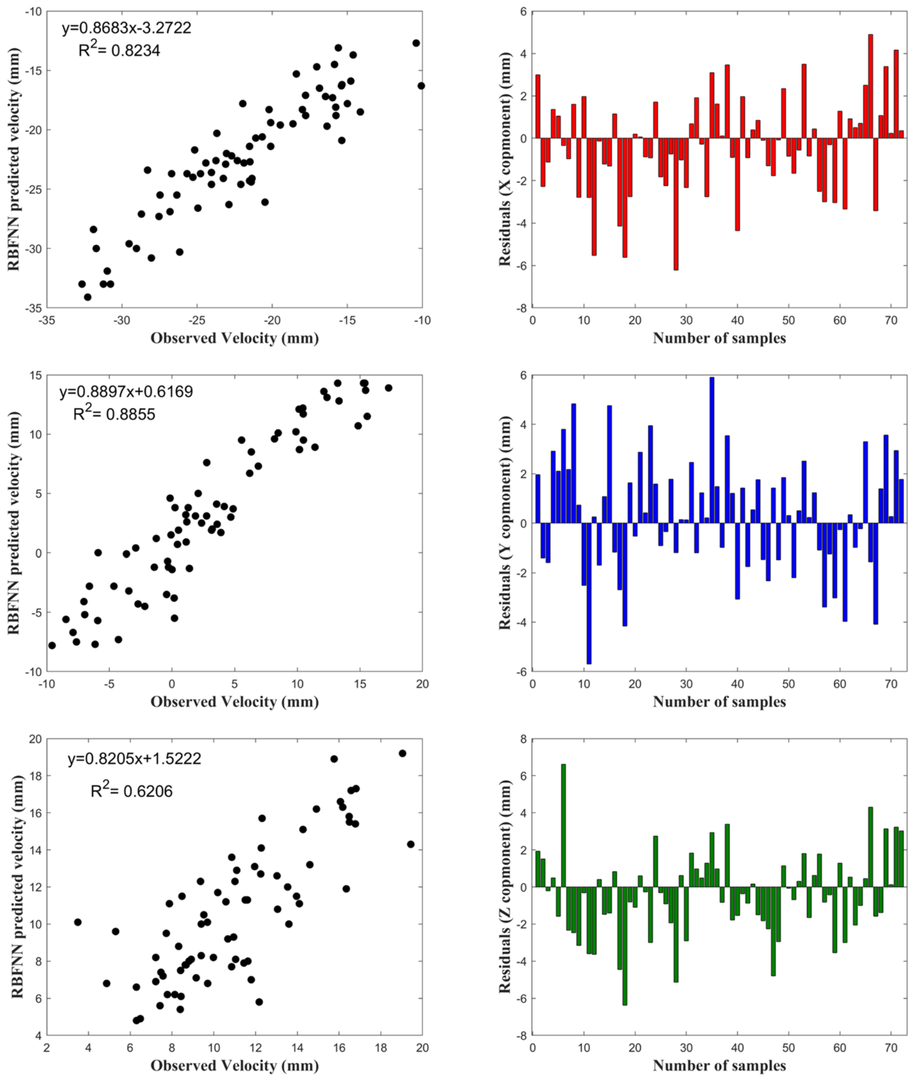

Fig. 7 Scatter diagrams of the predicted and observed velocity values and residuals obtained from the developed RBFNN model for testing points

models in the testing phase are shown in Fig. 8. For the MLPNN, GRNN, and RBFNN models, the velocity residual ranges were –6.99 mm–5.37 mm, –3.94 mm–3.87 mm, and –6.22 mm–4.89 mm for V_X; –6.08 mm–4.88 mm, –3.59 mm–3.93 mm, and –5.70 mm–5.90 mm for V_Y; and –7.29 mm–3.64 mm, –3.59 mm–3.36 mm, and –6.38 mm–6.61 mm for V_Z. As a result of the prediction made using the GRNN model, the velocity residuals did not exceed 4 mm for any of the geodetic point velocity components. The extreme velocity residuals were found for the V_Z geodetic point velocity component.

It is clear that the velocity residual values of the GRNN model were lower than either the MLPNN or RBFNN model.

Table 5 Comparisons of MLPNN, GRNN, and RBFNN models for each geodetic point velocity component

Types of model	Velocity	RMSE (mm)	MAE (mm)	R^2
MLPNN	V_X	2.57	1.92	0.77892
	V_Y	2.28	1.75	0.89700
	V_Z	2.39	1.95	0.61488
GRNN	V_X	**1.88**	**1.51**	**0.87061**
	V_Y	**1.81**	**1.47**	**0.93299**
	V_Z	**1.77**	**1.44**	**0.78032**
RBFNN	V_X	2.33	1.83	0.82344
	V_Y	2.34	1.90	0.88553
	V_Z	2.34	1.83	0.62057

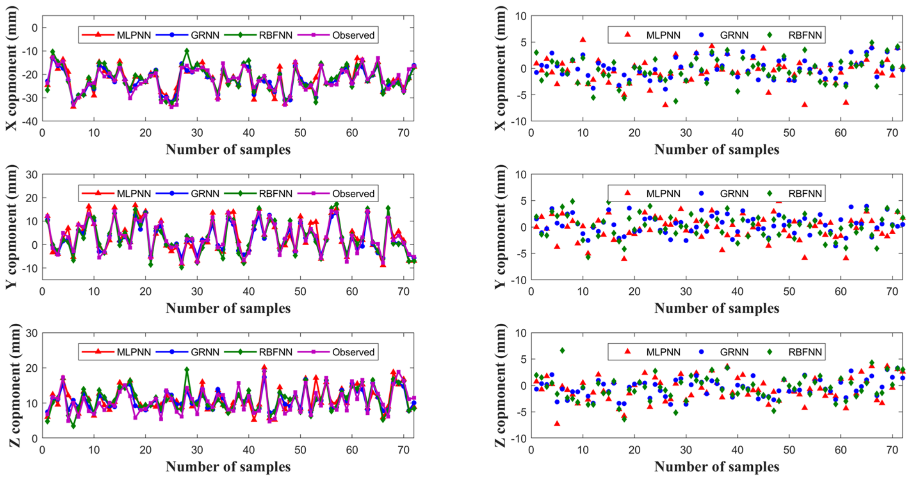

Fig. 8 Comparison between observed velocities and predicted velocities for each velocity component using MLPNN, GRNN, and RBFNN models

5 Conclusions

The applicability of the two ANN models, the MLPNN and RBFNN, in the prediction of geodetic point velocities has been investigated in previous studies. The MLPNN model was proposed as an alternative to classical interpolation methods in the prediction of geodetic point velocities (Güllü et al. 2011). In addition to the MLPNN, the RBFNN was applied and tested for prediction of geodetic point velocity (Yilmaz and Gullu 2014). However, the GRNN model has never been used for this purpose. This study investigated the potential of the GRNN in predicting the geodetic point velocities as compared to the MLPNN and RBFNN. All ANN model performances were evaluated and compared through the statistical parameters RMSE, MAE and R^2 for each model. The following conclusions can be made from the results of this study:

- All geodetic point velocity components (V_X, V_Y, V_Z) can be predicted used as output parameters simultaneously.

- The MLPNN, GRNN, and RBFNN models offered satisfactory prediction of geodetic velocity (at mm accuracy).
- The GRNN model was found to be superior to the other ANN models (MLPNN and RBFNN) since it achieved the lowest RMSE and MAE, and the highest R^2 values. In this regard, the GRNN can be said to be able to predict geodetic point velocity.
- After the GRNN, the RBFNN model was found to perform better than the MLPNN model.

Although the R^2 performance criteria results obtained from the ANN models were generally not very good, especially in the training phase, the RMSE and MAE results indicated that sufficient accuracy was achieved in both the training and testing phases. It should be noted that the R^2 should not be applied alone as a performance criterion (Legates and McCabe 1999) and that being equal to 1 does not guarantee that a model captures the behaviour of the investigated parameter (Kisi 2008). The reason for the low R^2 is thought to have been the distribution of the training and testing datasets. In this current study, to predict the velocities of the geodetic points, the training and testing datasets were selected randomly. Incorrect data partition could have led to reduced accuracy of the results on the predictive performance of the ANN model. To overcome this problem, the use of cross-validation (i.e. k-fold) technique has been recommended (Reitermanová 2010). Cross–validation is an assessment method used to improve the flexibility of a model, thus the performance of the proposed model; and then further statistical analysis will generalize onto an individual dataset. In further studies, the potential of the ANN with different cross-validation methods can be investigated for prediction of geodetic point velocity. Otherwise, the GRNN model should be applied and tested at different point densities. Thus, different datasets can be utilized to assess the impact of point density on the GRNN velocity prediction results. The GRNN-predicted outcomes can significantly deviate from the geodetically derived velocities. In this study, the GRNN can provide reasonable predictions, so it can be a promising tool for this purpose. The velocity predicting framework of the GRNN model needs further discussion and research. To assess the effectiveness of the GRNN, more studies should be conducted using different data sets in future works.

Acknowledgements The author would like to thank anonymous reviewers for their valuable comments which helped to improve this paper.

References

Akhoondzadeh M (2014) Investigation of GPS-TEC measurements using ANN method indicating seismo-ionospheric anomalies around the time of the Chile (Mw= 8.2) earthquake of 01 April 2014. Adv Space Res 54(9):1768–1772. https://doi.org/10.1016/j.asr.2014.07.013

Aktuğ B, Parmaksız E, Kurt M, Lenk O, Kılıçoğlu A, Gürdal MA, Özdemir S (2013) Deformation of central anatolia: GPS implications. J Geodyn 67:78–96. https://doi.org/10.1016/j.jog.2012.05.008

Aktuğ B, Sezer S, Özdemir S, Lenk O, Kılıçoğlu A (2011) Türkiye ulusal temel GPS ağı güncel koordinat ve hızlarının hesaplanması. Harita Dergisi 145:1–14 ((**in Turkish**))

Bogusz J, Kłos A, Grzempowski P, Kontny B (2014) Modelling the velocity field in a regular grid in the area of Poland on the basis of the velocities of European permanent stations. Pure Appl Geophys 171(6):809–833. https://doi.org/10.1007/s00024-013-0645-2

Broomhead DS, Lowe D (1988) Multivariable functional interpolation and adaptive networks. Complex Syst 2:321–355

Cakir L, Konakoglu B (2019) The impact of data normalization on 2D coordinate transformation using GRNN. Geod Vestnik 63(4):541–553. https://doi.org/10.15292/geodetski-vestnik.2019.04.541-553

Cakir L, Yilmaz N (2014) Polynomials, radial basis functions and multilayer perceptron neural network methods in local geoid determination with GPS/levelling. Measurement 57:148–153. https://doi.org/10.1016/j.measurement.2014.08.003

Cander LR (1998) Artificial neural network applications in ionospheric studies. Ann Geophys 41(5–6):757–766. https://doi.org/10.4401/ag-3817

Ching KE, Chen KH (2015) Tectonic effect for establishing a semi-dynamic datum in Southwest Taiwan. Earth Planets Space 67(1):207

Demir C, Açıkgöz M (2000) Türkiye ulusal temel GPS ağı noktalarındaki uzun peryotlu koordinat değişimlerinin (sküler hızların) kestirilmesi. Harita Dergisi, 1–19 (**in Turkish**)

Elshambaky HT, Kaloop MR, Hu JW (2018) A novel three-direction datum transformation of geodetic coordinates for Egypt using artificial neural network approach. Arab J Geosci 11(6):110. https://doi.org/10.1007/s12517-018-3441-6

Erol B, Erol S (2013) Learning-based computing techniques in geoid modeling for precise height transformation. Comput Geosci 52:95–107. https://doi.org/10.1016/j.cageo.2012.09.010

Farolfi G, Del Ventisette C (2016) Contemporary crustal velocity field in Alpine Mediterranean area of Italy from new geodetic data. GPS Solut 20(4):715–722. https://doi.org/10.1007/s10291-015-0481-1

Foresee FD, Hagan MT (1997) Gauss-Newton approximation to Bayesian learning. In: Proceedings of international conference on neural networks (ICNN'97), vol 3, pp 1930–1935. https://doi.org/10.1109/ICNN.1997.614194

Gülal E, Tiryakioğlu İ, Erdoğan S, Aykut NO, Baybura T, Akpinar B, Telli AK, Ata E, Gümüş K, Taktak F, Yilmaz İ, Öcalan T, Kalyoncuoğlu ÜY, Dolmaz MN, Elitok Ö, Erdoğan H, Soycan M (2013) Tectonic activity inferred from velocity field of GNSS measurements in southwest of Turkey. Acta Geod Geophys 48:109–121. https://doi.org/10.1007/s40328-012-0005-1

Gullu M (2010) Coordinate transformation by radial basis function neural network. Sci Res Essays 5:3141–3146

Güllü M, Yilmaz İ, Yilmaz M, Turgut B (2011) An alternative method for estimating densification point velocity based on back propagation artificial neural networks. Stud Geophys Geod 55(1):73–86. https://doi.org/10.1007/s11200-011-0005-6

Habarulema JB, McKinnell LA, Cilliers PJ (2007) Prediction of global positioning system total electron content using neural networks over South Africa. J Atmos Sol Terr Phys 69(15):1842–1850. https://doi.org/10.1016/j.jastp.2007.09.002

Hajian A, Ardestani EV, Lucas C (2011) Depth estimation of gravity anomalies using hopfield neural networks. J Earth Space Phys 37(2):1–9. https://doi.org/10.3997/2214-4609.20146872

Haykin S (1999) Neural networks: a comprehensive foundation. Prentice Hall, p. 842

Hernandez-Pajares M, Juan JM, Sanz J (1997) Neural network modeling of the ionospheric electron content at global scale using GPS data. Radio Sci 32(3):1081–1089

Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hu W, Sha Y, Kuang S (2004) New method for transforming global positioning system height into normal height based on neural network. J Surv Eng 130(1):36–39. https://doi.org/10.1061/(ASCE)0733-9453(2004)130:1(36)

Huang Z, Yuan H (2014) Ionospheric single-station TEC short-term forecast using RBF neural network. Radio Sci 49(4):283–292. https://doi.org/10.1002/2013RS005247

Inyurt S, Sekertekin A (2019) Modeling and predicting seasonal ionospheric variations in Turkey using artificial neural network (ANN). Astrophys Space Sci 364:62. https://doi.org/10.1007/s10509-019-3545-9

Ito C, Takahashi H, Ohzono M (2019) Estimation of convergence boundary location and velocity between tectonic plates in northern Hokkaido inferred by GNSS velocity data. Earth Planets Space 71(1):86. https://doi.org/10.1186/s40623-019-1065-z

Kaloop MR, Hu JW (2015) Optimizing the de-noise neural network model for GPS time-series monitoring of structures. Sensors 15(9):24428–24444. https://doi.org/10.3390/s150924428

Kaloop MR, Rabah M, Hu JW, Zaki A (2018a) Using advanced soft computing techniques for regional shoreline geoid model estimation and evaluation. Mar Georesour Geotechnol 36(6):688–697. https://doi.org/10.1080/1064119X.2017.1370622

Kaloop MR, Yigit CO, Hu JW (2018b) Analysis of the dynamic behavior of structures using the high-rate GNSS-PPP method combined with a wavelet-neural model: numerical simulation and experimental tests. Adv Space Res 61(6):1512–1524. https://doi.org/10.1016/j.asr.2018.01.005

Kavzoglu T, Saka M (2005) Modelling local GPS/levelling geoid undulations using artificial neural networks. J Geod 78:520–527. https://doi.org/10.1007/s00190-004-0420-3

Kierulf HP, Ouassou M, Simpson MJR, Vestøl O (2013) A continuous velocity field for Norway. J Geod 87(4):337–349. https://doi.org/10.1007/s00190-012-0603-2

Kisi Ö (2008) Constructing neural network sediment estimation models using a data-driven algorithm. Math Comput Simul 79(1):94–103. https://doi.org/10.1016/j.matcom.2007.10.005

Konakoglu B, Cakır L, Gökalp E (2016) 2D coordinates transformation using artificial neural networks. In: Geo advances 2016: ISPRS workshop on multi-dimensional and multi-scale spatial data modeling, At Mimar Sinan Fine Arts University/Istanbul, Volume XLII-2/W1: 3rd international geoadvances workshop. https://doi.org/10.5194/isprs-archives-XLII-2-W1-183-2016

Konakoğlu B, Gökalp E (2016) A Study on 2D similarity transformation using multilayer perceptron neural networks and a performance comparison with conventional and robust outlier detection methods. Acta Montan Slovaca 21(4):324–332

Kurt Aİ, Deniz R (2010) Deformasyon hızlarının iyileştirilmesinde sabit GPS istasyonları zaman serileri analizinden yararlanılması. Harita Dergisi 144:20–28 ((**in Turkish**))

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. https://doi.org/10.1029/1998WR900018

Lei Y, Zhao D, Cai H (2015) Prediction of length-of-day using extreme learning machine. Geod Geodyn 6(2):151–159. https://doi.org/10.1016/j.geog.2014.12.007

Li CK, Ching KE, Chen KH (2019) The ongoing modernization of the Taiwan semi-dynamic datum based on the surface horizontal deformation model using GNSS data from 2000 to 2016. J Geod 93(9):1543–1558. https://doi.org/10.1007/s00190-019-01267-5

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. https://doi.org/10.1016/j.jog.2011.12.004

Lin LS (2007) Application of a back-propagation artificial neural network to regional grid-based geoid model generation using GPS and leveling data. J Surv Eng 133(2):81–89. https://doi.org/10.1061/(ASCE)0733-9453(2007)133:2(81)

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.

MacKay DJ (1992) Bayesian interpolation. Neural Comput 4(3):415–447

Majdański M (2012) The structure of the crust in TESZ area by kriging interpolation. Acta Geophys 60(1):59–75. https://doi.org/10.2478/s11600-011-0058-5

Maruyama T (2008) Regional reference total electron content model over Japan based on neural network mapping techniques. Ann Geophys 25(12):2609–2614. https://doi.org/10.5194/angeo-25-2609-2007

McClusky S, Balassanian S, Barka A, Demir C, Ergintav S, Georgiev I, Gurkan O, Hamburger M, Hurst K, Kahle H, Kastens K, Kekelidze G, King R, Kotzev V, Lenk O, Mahmoud S, Mishin A, Nadariya M, Ouzounis A, Paradissis D, Peter Y, Prilepin M, Reilinger R, Sanli I, Seeger H, Teableb A, Toksöz MN, Veis G (2000) GPS constraints on crustal movements and deformations for plate dynamics. J Geophys Res 105:5695–5720. https://doi.org/10.1029/1999JB900351

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133. https://doi.org/10.1007/BF02478259

Mosavi MR (2006) A practical approach for accurate positioning with L1 GPS receivers using neural networks. J Intell Fuzzy Syst 17(2):159–171

Müller MD, Geiger A, Kahle HG, Veis G, Billiris H, Paradissis D, Felekis S (2013) Velocity and deformation fields in the North Aegean domain, Greece, and implications for fault kinematics, derived from GPS data 1993–2009. Tectonophysics 597–598:34–49. https://doi.org/10.1016/j.tecto.2012.08.003

Park J, Sandberg IW (1991) Universal approximation using radial basis function networks. Neural Comput 3(2):246–257. https://doi.org/10.1162/neco.1991.3.2.246

Patterson DW (1996) Artificial neural networks theory and applications. Prentice Hall, Upper Saddle River, p 477

Pereira RAD, De Freitas SRC, Ferreira VG, Faggion PL, dos Santos DP, Luz RT, Tierra Criollo AR, Del Cogliano D (2012) Evaluation of a few interpolation techniques of gravity values in the border region of Brazil and Argentina. In: Geodesy for planet Earth. Springer, Berlin, pp 909–915

Poyraz F, Hastaoğlu KO, Koçbulut F, Tiryakioğlu I, Tatar O, Demirel M, Duman H, Aydin C, Ciğer AF, Gursoy O, Turk T, Sigirci R (2019) Determination of the block movements in the eastern section of the Gediz Graben (Turkey) from GNSS measurements. J Geodyn 123:38–48. https://doi.org/10.1016/j.jog.2018.11.001

Reitermanová Z (2010) Data splitting. In: Safránková J, Pavlu J (eds) In: WDS 2010 proceedings of contributed papers, part I: mathematics and computer sciences. Matfyzpress, Prague, pp 31–36

Schuh H, Ulrich M, Egger D, Müller J, Schwegmann W (2002) Prediction of Earth orientation parameters by artificial neural networks. J Geod 76(5):247–258. https://doi.org/10.1007/s00190-001-0242-5

Specht DF (1991) A general regression neural network. IEEE Trans Neural Netw 2(6):568–576. https://doi.org/10.1109/72.97934

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. https://doi.org/10.1179/sre.2006.38.300.513

Tebabal A, Radicella SM, Nigussie M, Damtie B, Nava B, Yizengaw E (2018) Local TEC modelling and forecasting using neural networks. J Atmos Sol Terr Phys 172:143–151. https://doi.org/10.1016/j.jastp.2018.03.004

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(3):181–189. https://doi.org/10.1016/j.cageo.2007.03.011

Tierra A, Romero R (2014) Planes coordinates transformation between PSAD56 to SIRGAS using a multilayer artificial neural network. Geod Cartogr 63(2):199–209. https://doi.org/10.2478/geocart-2014-0014

Tierra AR, De Freitas SRC (2005) Artificial neural network: a powerful tool for predicting gravity anomaly from sparse data. In Gravity, geoid and space missions. Springer, Berlin, pp 208–213

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 Sens 3:668–683. https://doi.org/10.3390/rs3040668

Wang Q, Liao D, Zhou Y (2008) Real-time rapid prediction of variations of Earth's rotational rate. Chin Sci Bull 53(7):969–973. https://doi.org/10.1007/s11434-008-0047-5

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. https://doi.org/10.1007/s12040-014-0411-6

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

Ziggah YY, Youjian H, Tierra A, Konaté AA, Hui Z (2016a) Performance evaluation of artificial neural networks for planimetric coordinate transformation—a case study, Ghana. Arab J Geosci 9:698–714. https://doi.org/10.1007/s12517-016-2729-7

Ziggah YY, Youjian H, Tierra AR, Laari PB (2019) Coordinate transformation between global and local data based on artificial neural network with K-fold cross-validation in Ghana. Earth Sci Res J 23(1):67–77. https://doi.org/10.15446/esrj.v23n1.63860

Ziggah YY, Youjian H, Xianyu Yu, Basommi LP (2016b) Capability of artificial neural network for forward conversion of geodetic coordinates (ϕ, λ, h) to cartesian coordinates (X, Y, Z). Math Geosci 48:687–721. https://doi.org/10.1007/s11004-016-9638-x