

# INTEGRATION OF ARTIFICIAL NEURAL NETWORK AND THE OPTIMAL GNSS SATELLITES' CONFIGURATION FOR IMPROVING GNSS POSITIONING TECHNIQUES (A CASE STUDY IN EGYPT)

Mustafa K. ALEMAM<sup>1,2</sup>, Bin YONG<sup>1,3</sup>, Abubakar S. MOHAMMED<sup>4</sup>

<sup>1</sup> School of Earth Sciences and Engineering, Hohai University, Nanjin, China

<sup>2</sup> Faculty of Engineering, Assiut University, Assiut, Egypt

<sup>3</sup> State Key Laboratory of Hydrology-Water Resources and Hydraulic  
Engineering, Hohai University, Nanjing, China

<sup>4</sup> Faculty of Construction and Environment, Polytechnic University, Hong Kong

e-mail: [mustafaalemam2014@gmail.com](mailto:mustafaalemam2014@gmail.com)

**ABSTRACT.** Nowadays, the global navigation satellite system (GNSS) positioning techniques based on the International GNSS Service (IGS) products are extensively used for various precise applications. However, specific conditions such as the dual-frequency observations and the final IGS products are required. Consequently, the absence of the final IGS data and using single-frequency observations will degrade these techniques' accuracy. In this paper, two algorithms through two separated stages are formulated for improving the single-frequency GNSS observations by using one GNSS receiver based on the broadcast ephemerides in real time or close to real time. The first algorithm represents the preparation stage for the second one. It classifies the observations by separating the optimal values of position dilution of precision (PDOP) and the number of satellites (NOS), as well as the corresponding values of coordinates. The second stage includes an algorithm based on the artificial neural network (ANN) approach, which is set at the ANN variables that produce the best precision through the applied tests at the present study. Binary numbers, log sigmoid-Purelin, cascade forward net, and one hidden layer with a size of 10 neurons are the optimal variables of ANN inputs format, transfer functions constellations, feedforward net type, and the number of hidden layers (NHL) and its size, respectively. The simulation results show that the designed algorithms produce a significant improvement in the horizontal and vertical components. Lastly, an evaluation stage is performed in the case of dual-frequency observations by using broadcast ephemerides. The simulation outputs indicate that the precision at applying the proposed integration is completely enhanced compared with the outputs of IGS final data.

**Keywords:** GNSS, artificial neural network, satellites' configuration, IGS, single-frequency observations

## 1. INTRODUCTION

Global navigation satellite system (GNSS) plays an essential role in a wide range of applications such as geodetic, geodynamic, surveying, and mapping (e.g., Mohamed 2015). Numerous sources of errors (e.g., satellite ephemerides, satellite clock, ionospheric and tropospheric, and



multipath errors) affect GNSS positioning accuracy. The differential GNSS (DGNSS) technique, which basically requires a reference station (Correa Muñoz and Cerón-Calderón 2018), is widely used to eliminate or mitigate these errors. Besides, GNSS performance can be improved through various factors such as the number of satellites (NOS) (Lemmon and Gerdan 1999, Specht et al. 2015, Davidovic and Mijic 2017) and position dilution of precision (PDOP) (Nie et al. 2016, Biswas et al. 2017, Janowski and Rapinski 2016, Han et al. 2014, Teng and Wang 2015) (i.e., satellites configuration), cut-off angle (Teunissen et al. 2013), recording interval (Yousef and Alemam 2014), occupation time (Mohamed 2015, McGaughey et al. 2017), and GNSS constellations (El Manaily et al. 2017, Rabbou and El-Rabbany 2015).

Economically, DGNSS is not favorable for most users because it requires at least two receivers (i.e., one receiver as a reference station and the other as a rover); consequently, high cost and a long time in fieldwork are consumed. Furthermore, the real time kinematic (RTK) GNSS positioning technique, which works like DGNSS depending on the radio communication, has a limited distance for communication. The poor connection due to obstacles, especially in downtown areas, is an additional factor. Thus, the GNSS techniques that use only one GNSS receiver, that is, independent during fieldwork, are the priority choice for users.

The popular GNSS positioning techniques that use one GNSS receiver in the fieldwork are the precise point positioning (PPP) and the DGNSS with respect to the international GNSS service (IGS) stations (IGS DGNSS) (El-Tokhey et al. 2018). The IGS is the most considered organization for GNSS service. The IGS stations play a significant contribution in DGNSS positioning, which is used to get the high level of accuracy of another type of GNSS stations, for instance, permanent GNSS stations, campaign mode GNSS stations, climate modeling, meteorology, and space weather applications (Ansari et al. 2017).

Recently, these techniques (i.e., PPP and IGS DGNSS) have provided a high position accuracy, taking advantage of both the IGS networks and the IGS products range (e.g., precise GNSS orbits and clock products) (Martín et al. 2011, Farah 2017). It should be pointed out that several PPP or relative positioning online services have been established for providing processed data for the GNSS observations (Heßelbarth and Wanninger 2010, Jamieson and Gillins 2018). Some of the service providers include, for example, the Geoscience Australia Online GPS Processing Service (AUSPOS) (<http://www.ga.gov.au>), the Scripps Coordinate Update Tool (SCOUT), which was developed by the Scripps Orbit and Permanent Array Centre (SOPAC) (<http://sopac.ucsd.edu>), and the Canadian Spatial Reference System (CSRS)-PPP (<http://www.geod.nrcan.gc.ca>) (Ghoddousi-Fard and Dare 2005). According to the nearest IGS stations, these services provide corrections for the GNSS observations using the IGS precise orbits. Table 1 shows the IGS satellites' ephemerides and clock products (<http://www.igs.org/products>).

As shown in Table 1, the final ephemerides, which are available approximately 2 weeks after the last observations (Martín et al. 2011), have the highest quality and internal consistency compared to other products. The broadcast and ultra-rapid ephemerides, which produce the worst accuracy, are available in real time. Consequently, obtaining high accuracy in real time is considered as one of the important subjects in recent studies.

Presently, the application of the artificial neural network (ANN) as an intelligent processing tool is highly recommended for solving this type of issue; it has been widely developed and applied in different fields, including function approximation, classification, and pattern recognition (Zhou et al. 2017). Evaluating it as a suitable technique for solving most geodetic issues (e.g., enhancing the GNSS technique) has been investigated through several studies.

**Table 1.** Characteristics of IGS satellite ephemerides and clock products (2019)

Type		Accuracy	Latency	Updates	Sample interval
<i>GPS satellite ephemerides/satellite and station clocks</i>					
Broadcast	Orbits	~100 cm	Real time	--	Daily
	Sat. clocks	~5 ns RMS ~2.5 ns SD			
Ultra-rapid (predicted half)	Orbits	~5 cm	Real time	Four times/day	15 min
	Sat. clocks	~3 ns RMS ~1.5 ns SD			
Ultra-rapid (observed half)	Orbits	~3 cm	3–9 h	Four times/day	15 min
	Sat. clocks	~150 ps RMS ~50 ps SD			
Rapid	Orbits	~2.5 cm	17–41 h	One time/day	15 min
	Sat. and stn clocks	~75 ps RMS ~25 ps SD			5 min
Final	Orbits	~2.5 cm	12–18 days	One time/week	15 min
	Sat. and stn clocks	~75 ps RMS ~20 ps SD			Sat.: 30 s Stn.: 5 min
<i>GLONASS satellite ephemerides</i>					
Final	Orbits	~3 cm	12–18 days	Every Thursday	15 min
where: ns = nanosecond, ps = picosecond, UTC =Universal Coordinated Time, RMS = Root Mean Square errors, SD = standard deviation					

For instance, Yang et al. (2015), Gera et al. (2011), and Mosavi and Mohammadi (2002) improved GNSS positioning accuracy. Zhou et al. (2017), Kaygısız et al. (2007), and Malleswaran et al. (2014) enhanced performance and position accuracy of the global positioning system (GPS)/inertial navigation system (INS) integrated navigation. Ordóñez Galán et al. (2013) studied the influence of forestry environments on the accuracy of GPS. Besides, Mohammed et al. (2015), Wang et al. (2016), Kim and Kim (2015), and Barrile et al. (2016) applied ANN to predict GPS positioning, GPS satellite clock bias, IGS real time service corrections, and displacement in tectonically active areas, respectively. Furthermore, it has been used for geodetic coordinates' transformation (Elshambaky et al. 2018, Ziggah et al. 2016b), a corrector surface determination for global geopotential model (Elshambaky 2018), planimetric coordinates' transformation (Ziggah et al. 2016a), and Geometrical Dilution of Precision (GDOP) approximation for improving GPS accuracy (Tafazoli and Mosavi 2011, Azami et al. 2012, Chien-Sheng and Szu-Lin 2010, Tafazoli et al. 2011, Jwo and Lai 2006, Azami and Sanei 2013, Azarbad et al. 2014, Azami et al. 2016). The general insight gathered from these previous works indicate that ANN is suitable for numerous geodetic applications.

As is well known, low-cost, single-frequency GNSS receivers provide an accuracy less than that provided by using high-cost, dual-frequency GNSS receivers. As a result, single-frequency GNSS observations have gained a wide research interest for applications that need high accuracy with less cost, especially in developing countries such as Egypt. Consequently, this investigation was carried out in Egypt, mainly to improve single-frequency GNSS observations

in real time by employing the ANN approach. The GNSS positioning technique selected in the present study is the IGS DGNSS. However, this study's outputs are applicable to both differential and PPP methods that utilize IGS final products. It is noteworthy, to our best knowledge, that ANN has not been applied in Egypt for improving GNSS positioning techniques' accuracy.

This study, basically, focuses on an algorithm formulated based on the ANN approach. The ANN is optimized to provide the best performance. In addition, the satellites' configuration that represents one of the significant effective factors on GNSS positioning accuracy was taken into account by improving the input data to ANN. Simultaneously, an algorithm was designed for selecting the best values of the input data according to the optimal satellites' configuration (NOS and PDOP). Generally, the ANN in this study focuses on the dynamics of numbers as well as using the binary format, which represents an essential role for achieving the required accuracy, instead of decimal numbers for input data. Finally, the output data is filtered, and the final outputs are then calculated using the least square method (LSM).

The main objectives of this study are to (i) test the affected parameters in the ANN and determine the best values that produce an acceptable precision, (ii) improve single-frequency GNSS observations in real time applications, and (iii) test the final simulation in the case of dual-frequency GNSS observations by comparing its outputs using broadcast ephemerides with the data processed using the final IGS orbits.

## 2. ARTIFICIAL NEURAL NETWORK

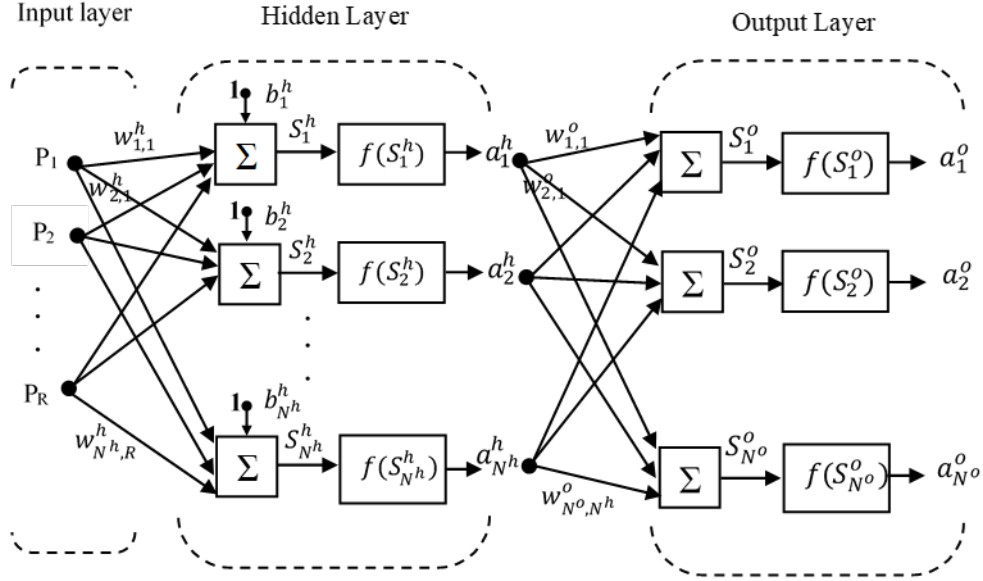
ANN is a technology inspired by biological neurons' structure to model the human brain's behavior while performing a particular function. Generally, it consists of a series of layers, namely, the input layer, hidden layer, and the output layer (Fig. 1).

As shown in Fig. 1, the input layer includes input  $\mathbf{P}$ , which has a single vector of  $\mathbf{R}$  elements that are connected with the hidden layer by synaptic weights,  $\mathbf{W}_{N^h, \mathbf{R}}^h$ . The synaptic weights  $\mathbf{W}_{N^o, N^h}^o$  make connections between the hidden and output layers. The superscripts  $\mathbf{h}$  and  $\mathbf{o}$  indicate the hidden and output layer, respectively. The first index indicates the specific neuron destination for that weight; the second index denotes the source of the signal fed to the neuron; while  $N^h, N^o$  represent the number of neurons (NON) in the hidden and output layers, respectively. These neurons represent the processing units in ANN, where each neuron has a bias  $\mathbf{b}$ , summation  $\mathbf{S}$ , activation function  $\mathbf{f}$ , and output  $\mathbf{a}$ . The hidden layer's output  $\mathbf{a}^h$  is the input for the output layer, which produces an output  $\mathbf{a}^o$ . The bias  $\mathbf{b}$  is like a weight, but it has a constant input of 1. The summation  $\mathbf{S}$  can be calculated in the hidden and output layers using Equations (1) and (2):

$$S_j^h = \sum_{i=1}^R W_{ji}^h P_i + b_j^h \quad (1)$$

$$S_k^o = \sum_{j=1}^{N^h} W_{kj}^o a_j^h + b_k^o \quad (2)$$

where  $\mathbf{S}_j^h$  and  $\mathbf{S}_k^o$  represent the weighted inputs summed with the bias at hidden neuron  $\mathbf{j}$  and output neuron  $\mathbf{k}$ , respectively.



**Figure 1.** ANN architecture

This summation is fed to the activation function (i.e., transfer function), which squashes its inputs to specific values depending on the activation function type. There are three activation functions that are most commonly used (Azami and Sanei 2013, Kim and Kim 2015). The first function (Equation 3) is a linear transfer function (Purelin), which produces an output equal to its input, where

$$f(S) = S \quad (3)$$

The second function (Equation 4) is the log sigmoid (Logsig); this transfer function takes the input which lies between plus and minus infinity and squashes the output into the range 0–1, using the expression

$$f(S) = \frac{1}{1+e^{-S}} \quad (4)$$

Equation (5) presents the third function, which is the hyperbolic tangent sigmoid (Tansig); this function is like the log sigmoid function, but the output here is squashed into the range –1 to 1 using the expression

$$f(S) = \frac{e^S - e^{-S}}{e^S + e^{-S}} \quad (5)$$

However, it is important to mention that ANN may have one or more hidden layers (i.e., multilayer network). These hidden layers may have different transfer functions and different NON. There are various types of ANN. The multilayer feedforward network that applied the backpropagation (BP) algorithm for training the network is one of the most commonly used in ANN applications.

## 2.1. BP algorithm

The mean square error (MSE) represents the network performance index in the BP algorithm, which can be obtained by comparing each network's output with its corresponding target. In doing so, the approximate steepest descent rule is used for updating the weights and biases to minimize the MSE. Network training by the BP algorithm is accomplished through two different passes of computations, namely, forward pass and backward pass.

### 2.1.1. Forward pass computations

Based on the notations in Fig. 1, the computations in the forward pass can be expressed as indicated in Equations (6) and (7), where Equations (1) and (2) of the summations at neuron  $j$  and  $k$  are modified to

$$S_j^h(n) = \sum_{i=0}^{\dot{R}} W_{ji}^h(n) P_i(n) \quad (6)$$

$$S_k^o(n) = \sum_{j=0}^{\dot{N}^h} W_{kj}^o(n) a_j^h(n) \quad (7)$$

for simplifying the mathematical operations, where  $\dot{R}$  and  $\dot{N}^h$  are the inputs (including the constant 1) for hidden and output layers, respectively. The synaptic weights  $W_{j0}^h$  and  $W_{k0}^o$  are  $b_j^h$  and  $b_k^o$  at neuron  $j$  and  $k$ , respectively, and  $n$  denotes the iteration number. This iteration (epoch) indicates the recurrence number of updating weights until the acceptable error is achieved.

$a_j^h$ , which represents the output of neuron  $j$  in the hidden layer, can be calculated by applying the associated activation function  $f$  as follows:

$$a_j^h(n) = f_j(S_j^h(n)) \quad (8)$$

This activation function is selected based on the problem needed to be solved. Likewise,  $a_k^o$ , which denotes the output of neuron  $k$  in the output layer and represents the network's output, can be calculated by the equation

$$a_k^o(n) = f_k(S_k^o(n)) \quad (9)$$

Let  $y_k = a_k^o$ ; correspondingly, the error signal produced at the output of neuron  $k$  is defined by the equation

$$e_k(n) = d_k(n) - y_k(n) \quad (10)$$

where  $d_k$  is the  $i$ th desired (i.e., target) output for the  $i$ th element of the vector input  $R$ .

Now, the computations will be carried out in the opposite direction to update the weights for minimizing the produced error.

### 2.1.2. Backward pass computations

The steepest descent algorithm (i.e., delta rule) for the approximate MSE is expressed in Equation (11) as

$$\Delta w_{kj}(n) = -\eta \frac{\partial E(n)}{\partial w_{kj}(n)} \quad (11)$$

where  $\Delta w_{kj}$  is the applied correction to the synaptic weight  $w_{kj}$ . Moreover,  $\eta$  denotes the learning rate.

$E$  represents the error energy of neuron  $k$ , that is, MSE, which can be written as

$$E_k(n) = \frac{1}{2} e_k^2(n) \quad (12)$$

according to the least mean square (LMS) algorithm.

Apparently, the error is an indirect function of the weights. Therefore, the chain rule of the calculus was proposed to calculate the derivatives.

Let  $S_k^o = v_k$ ; the partial derivative  $\partial E(n) / (\partial w_{kj}(n))$ , according to the chain rule, can be written as

$$\frac{\partial E(n)}{\partial w_{kj}(n)} = \frac{\partial E(n)}{\partial e_k(n)} \frac{\partial e_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial w_{kj}(n)} \quad (13)$$

Consequently, differentiating Equations (7), (9), (10), and (12) with respect to  $w_{kj}$ ,  $v_k$ ,  $y_k$  and  $e_k$ , respectively, yields

$$\frac{\partial E(n)}{\partial w_{kj}(n)} = -e_k(n) \hat{f}_k(v_k(n)) a_j^h(n) \quad (14)$$

The use of Equation (14) in Equation (11) gives

$$\Delta w_{kj}(n) = \eta \delta_k(n) y_j(n) \quad (15)$$

$$\text{where } \delta_k \text{ (local gradient)} = e_k(n) \hat{f}_k(v_k(n)) \quad (16)$$

$$\text{and } y_j(\mathbf{n}) = a_j^h(\mathbf{n}).$$

Equation (15) has been modified by adding a momentum term as shown by the equation

$$\Delta w_{kj}(n) = \alpha \Delta w_{kj}(n-1) + (\eta \delta_k(n) y_j(n)) \quad (17)$$

in order to control the rate of learning, where  $\alpha$  is the momentum constant.

Thus, the updated weight in the output layer can be calculated using the expression

$$w_{kj}(n+1) = w_{kj}(n) + \alpha [\Delta w_{kj}(n-1)] + (\eta \delta_k(n) y_j(n)) \quad (18)$$

Since the hidden layer shares the responsibility for any error made at the output layer, Equation (18) in the hidden layer becomes

$$w_{ji}(n+1) = w_{ji}(n) + \alpha [\Delta w_{ji}(n-1)] + (\eta \delta_j(n) y_i(n)) \quad (19)$$

where

$$\delta_j = \hat{f}_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (20)$$

For more details, refer to Haykin (2009) and Beale et al. (1996).

Accordingly, the BP algorithm was applied in this study by selecting a network training function called Train Levenberg–Marquardt (LM). This function updates the values of weights and biases according to Levenberg–Marquardt optimization, which is often the fastest BP algorithm.

### 3. MATERIALS AND METHODS

#### 3.1. Study area and data sources

The study area consists of five geodetic stations in Egypt established by the Egyptian surveying authority. These stations are distributed in five cities: Baltim, Suez, Helwan, Cairo, and Assiut. Egypt is a country situated in the northeast corner of Africa, sharing borders with Libya to the west, Sudan to the south, the Mediterranean Sea to the north, and the red sea to the east. It lies between the latitudes 22° and 32° North and the longitudes 24° and 37° East (Bear et al. 1999).

The observations were recorded with different GNSS instruments (dual frequency) at different dates. Likewise, two IGS stations, DRAG and RAMO, which are nearest to the study area, have been chosen for applying the IGS DGNSS technique. CSRS-PPP and SOPAC online services were used to obtain the reference coordinates of the test points and IGS points, respectively. Fig. 2 shows the study area and the location of the test stations as well as the utilized IGS stations. Furthermore, GNSS instruments used for recording the observations, reference coordinates, and data sources are illustrated in Table 2.



**Figure 1.** Study area and IGS stations

**Table 2.** Data sources

Station	Station type	GNSS instrument	Reference coordinates			
			Longitude ° ' "	Latitude ° ' "	Ellipsoidal height, m	Source
<b>DRAG</b>	IGS	Leica GRX1200	35 23 31.46180	31 35 35.5288	31.834	SOPAC
<b>RAMO</b>	IGS	Leica RS500	34 45 47.31050	30 35 51.38602	886.829	SOPAC
<b>Baltim</b>	Test	Trimble R8	31 04 49.15000	31 35 45.42070	31.163	CSRS
<b>Suez</b>	Test	Trimble R8	32 36 22.45620	30 07 09.53080	53.827	CSRS
<b>Helwan</b>	Test	Trimble R8	31 20 37.30370	29 51 33.72150	135.055	CSRS
<b>Cairo</b>	Test	Leica GR10	31 14 16.45330	30 02 43.33490	68.268	CSRS
<b>Assiut</b>	Test	Ashtech Z-Xtreme	31 10 19.90010	27 11 12.12040	91.420	CSRS

### 3.2. Methodology

For achieving the desired accuracy of coordinates as a final output from GNSS observations as a first input, three main stages were considered using powerful packages of data processing and analyzing that are gained in MATLAB software R2018a and Excel program 2016 after using GNSS software (the Trimble Business Centre [TBC] and Ashtech solution) to post-process the observations.

The mentioned stages are illustrated in Fig. 3 and discussed in the following subsections.



### 3.2.1. GNSS software stage

Conventionally, GNSS software requires observations of a control point and an unknown point to run a proper processing operation. At this stage, processing operation through the normal way was implemented. In addition to that, the observations of a known point were processed in parallel by the same control point. These points were recorded at the same time. Thus, the IGS station DRAG was fed to TBC software as a control point and the IGS station RAMO was fed as Rover1, while a test point was fed as Rover2. Subsequently, TBC processing was performed separately for each test point at 30 s sampling time and 2 h of occupation time by applying the following scenarios:

- broadcast ephemerides with single-frequency observations,
- broadcast ephemerides with dual-frequency observations, and
- precise ephemerides with dual-frequency observations.

The TBC cannot extract the coordinates at each sampling time, PDOP, and NOS. Therefore, the Ashtech solution is employed for this operation. The GNSS software outputs were converted from geodetic coordinates (latitude, longitude, and ellipsoidal height) to geocentric coordinates, that is, Earth Centre Earth Fixed (X, Y, and Z), using MATLAB functions because the geodetic coordinates are not suitable for the computations stage. Finally, the X, Y, Z, PDOP, and NOS at each sampling time of Rover1 and Rover2 were extracted and fed to the classification stage.

### 3.2.2. Classification stage

An algorithm was formulated for this stage (Fig. 4); it includes two main parts. In the first part, the minimum and maximum values of NOS and PDOP were calculated to determine the available NOS and PDOP. This classification was conducted through different satellite configurations with NOS representing values (NOSmin, ..., NOSmax) and PDOP as periods (PDOP1 = PDOP < 3, PDOP2 = 3 ≤ PDOP < 5, and PDOP3 = PDOP ≥ 5). For example, the coordinate error and the 3D position error at NOSmin and PDOP1 were calculated in terms of standard deviation (SD) as described in Equations (21)–(24) (Inal et al. 2018)

$$\sigma_X = \sqrt{\frac{\sum(X_t - X_i)^2}{n}} \quad (21)$$

$$\sigma_Y = \sqrt{\frac{\sum(Y_t - Y_i)^2}{n}} \quad (22)$$

$$\sigma_Z = \sqrt{\frac{\sum(Z_t - Z_i)^2}{n}} \quad (23)$$

$$\sigma_P = \sqrt{(\sigma_X)^2 + (\sigma_Y)^2 + (\sigma_Z)^2} \quad (24)$$

where

- $\sigma_X, \sigma_Y,$  and  $\sigma_Z$  are the SDs of X, Y, and Z, respectively;
- $\sigma_P$  is the 3D position error;
- $X_t, Y_t,$  and  $Z_t$  are the true values of X, Y, and Z, respectively;
- $X_i, Y_i, Z_i$  are the  $i$ th values of X, Y, and Z, respectively; and
- $n$  is the number of observations.

Consequently, the optimum values of NOS, PDOP, sampling times (which produced the best precision), and the corresponding values of X, Y, and Z were obtained. In the second part, the best coordinates values for Rover2 were selected based on the optimal values of NOS, PDOP, and sampling time at rover1.

As a result, the data obtained for Rover1 (known station) and Rover2 (unknown station) were entirely similar in the following parameters:

- control point,
- NOS,
- PDOP,
- sampling time,
- occupation time, and
- the time of collecting data in the day.

It is noteworthy that these parameters are the most influential factors for GNSS accuracy.

### 3.2.3. ANN stage

Here, the classification outputs were fed to an algorithm (Fig. 5) designed based on the ANN approach. The ANN layers were optimized by selecting the best parameters at each layer for obtaining the best accuracy. Table 3 shows the tested parameters and their values, which were used for each test. Furthermore, as mentioned above, Train LM was selected as the training function in all tests. The training parameters that occur according to the Train LM function and learning rate are shown in Table 4. The algorithm was then run for each test, and the simulation prediction errors of X, Y, and Z were calculated in terms of the SDs.

#### 3.2.3.1. Input layer

Since the inputs' accuracy has a serious effect on the final outputs, an independent stage (i.e., the classification stage) was employed to improve it. Eventually, the input layer was optimized.

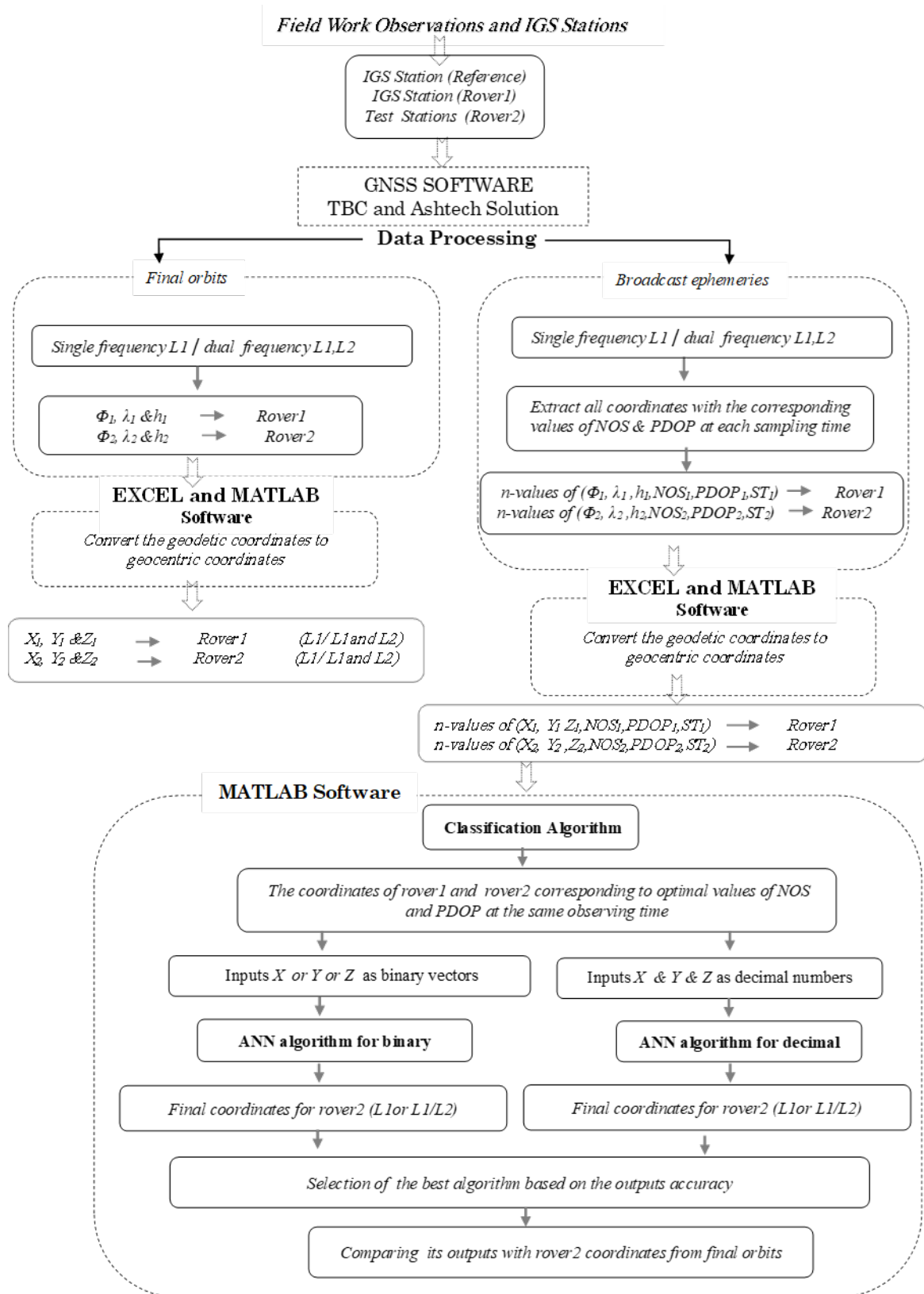
**Table 3.** The tested parameters in the ANN algorithm

Test no.		Transfer function	ANN type	NHL	NON	NOI
1	Transfer function	X	Fit net	1	10	100
2	ANN type	The optimal at test 1	X	1	10	100
3	NHL	The optimal at test 1	The optimal at test 2	X	10	100
4	NON	The optimal at test 1	The optimal at test 2	The optimal at test 3	X	100
5	NOI	The optimal at test 1	The optimal at test 2	The optimal at test 3	The optimal at test 4	X

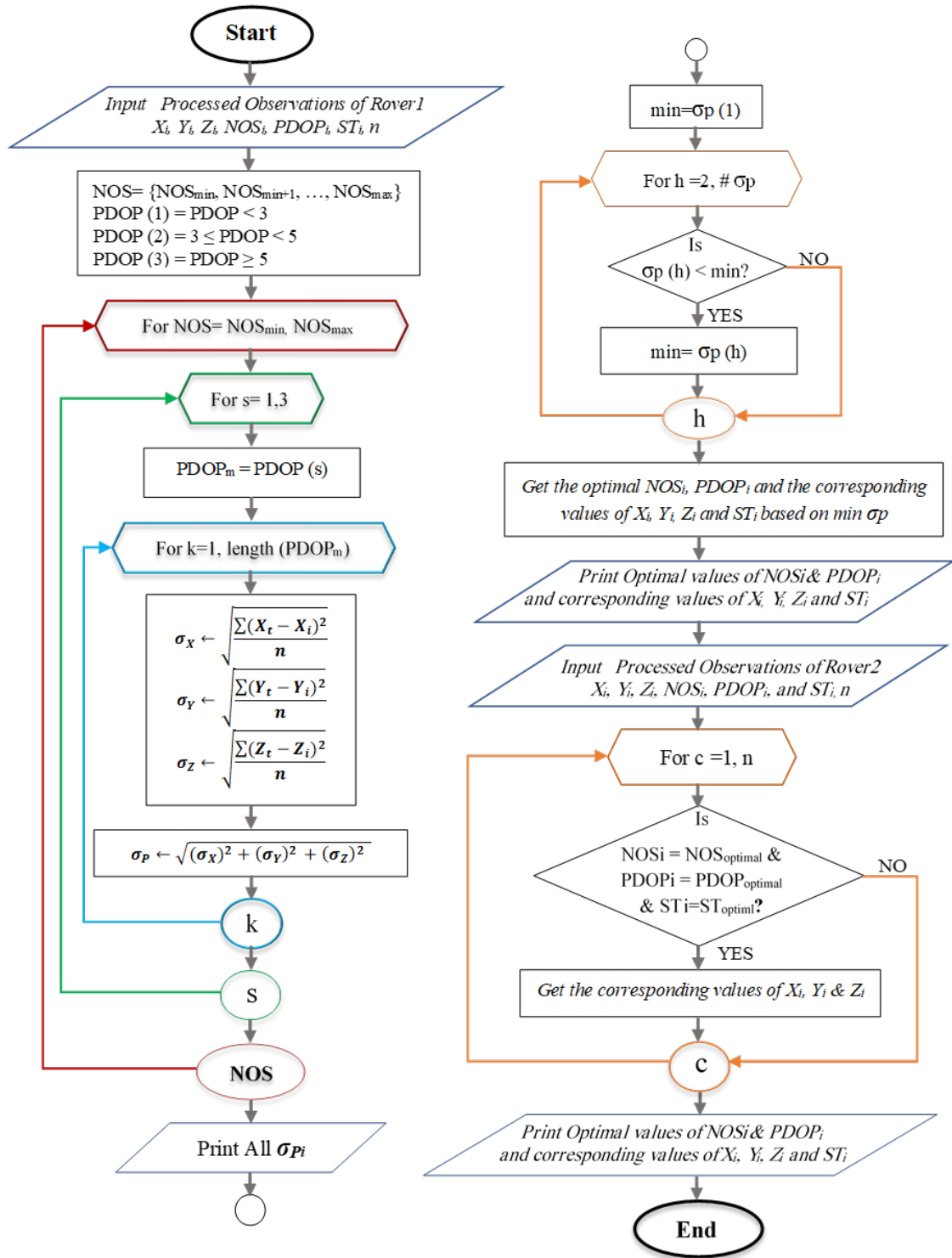
X: The selected values for testing

**Table 4.** The parameters involved in the ANN algorithm

Epoch	Goal	Max_fail	Min_fail	Mu	Learning rate
1000	0	6	1e-7	0.001	0.01

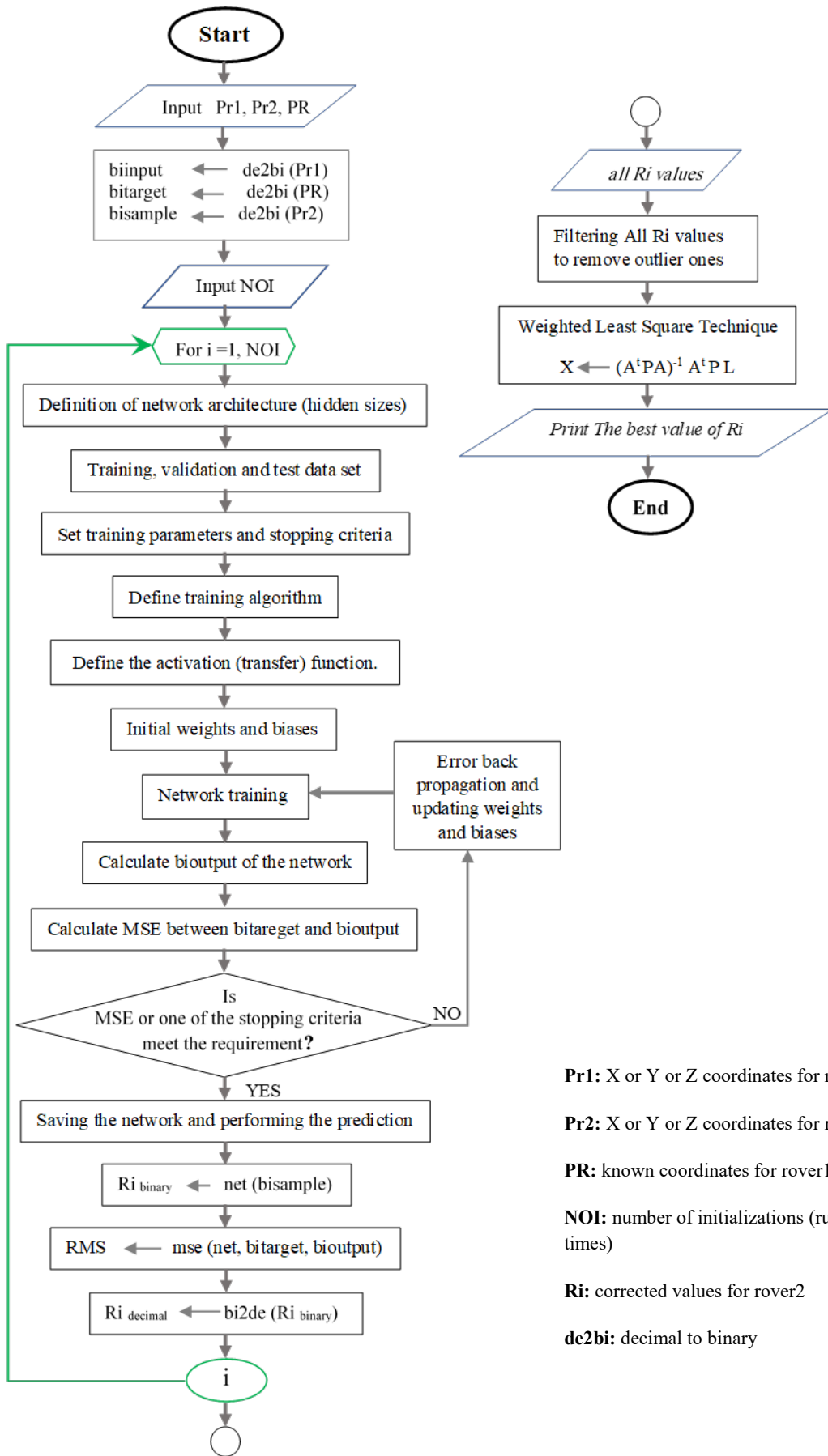


**Figure 3.** Methodology flowchart



**Figure 4.** Algorithm steps of the classification stage

$X_i, Y_i, Z_i$ : coordinates values at each NOS, PDOP and ST  
 $n$ : Number of the processed observations



**Pr1:** X or Y or Z coordinates for rover1  
**Pr2:** X or Y or Z coordinates for rover2  
**PR:** known coordinates for rover1  
**NOI:** number of initializations (running times)  
**Ri:** corrected values for rover2  
**de2bi:** decimal to binary

**Figure 5.** Algorithm steps of the ANN stage (binary numbers)

### 3.2.3.2. Hidden layer

The parameters in the hidden layer have been optimized through five tests. Test 1 includes studying the effect of transfer function on accuracy. Different constellations among transfer functions (Tansig, Logsig, and Purelin), which are usually used in ANN at hidden and output layers, were studied. These constellations were based on pairing each function with itself in the first attempt and with the other functions subsequently; for instance, Purelin at the hidden layer and Purelin at the output layer for the first constellation; the next constellation was Purelin at the hidden layer and Logsig at the output layer, in that order.

Notably, when a trial was performed for the test, only the simulation outputs of Purelin–Purelin constellation produced an acceptable coordinate value. Conversely, constellations that have Logsig or Tansig produced inconsistent values. It is well known that the outputs of sigmoid functions are squashed into the range (0–1) and (-1 to 1) with Logsig and Tansig, respectively (Haykin 2009). Consequently, it was proposed to convert the input coordinates into (1,0), that is, binary format, and test it with the coordinates in decimal numbers.

The decimal numbers were converted to binary ones through the proposed algorithm of this stage and then were reversed at the output layer for performing the necessary calculations and analysis. It is important to note that binary numbers occupy many cells in the worksheet, but the decimals need one cell for each coordinate. Therefore, to guarantee more clearance and accurate outputs, each coordinate was fed separately in the case of binary format.

Fit net, pattern net, and cascade forward net are the specialized versions of feedforward networks, which can be used for any kind of input to output mapping (Zhou et al. 2017, Kaygısız et al. 2007, Zhang 2000). Hence, these networks were selected for studying the ANN types in test 2. Test 3 studies the effect of changing the number of hidden layers (NHL) on output accuracy. A range from 1 to 10 was selected for testing. Additionally, the NON, which represents the hidden layer's size, was investigated in test 4. The values 1, 2, 3, 4, 5, 10, 15, 20, 30, 40, and 50 were selected for testing.

Because of the random set of the initial weights and biases, the outputs may be changed at each initialization. Thus, the effects of the number of initializations (NOI), that is, the algorithm's running times, on the output accuracy were studied in test 5. A range from 1 to 10,000 was divided into four groups (with intervals of 1, 10, 100, and 1000) and proposed for testing. One of the advantages of the designed algorithm is its ability to reinitialize the weights and biases at a specific value or period (group) within the NOI, in addition to calculating the mean value.

Fig. 6 shows how the algorithm behaves while calculating the outputs of the NOI as a group.

### 3.2.3.3. Output layer

The output data have been filtered by applying an outlier function which removes the outlier elements based on the  $3\sigma$  rule. After obtaining the filtered data, the LSM was applied to estimate the data sets' optimal value. It can minimize the weighted residuals ( $v$ ) by applying the normal equation

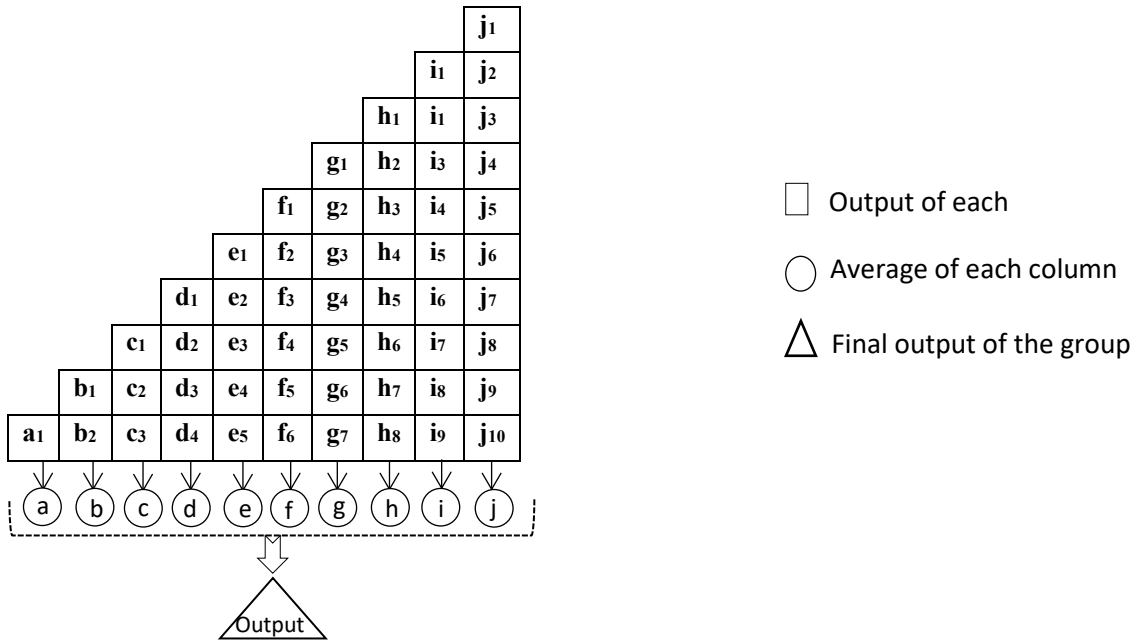
$$L + v = AX \quad (25)$$

and the solution model

$$X = (A^T P A)^{-1} A^T P L \quad (26)$$

on the output data, where  $L$  is the output matrix,  $A$  is the design matrix,  $X$  is the unknown matrix, and  $P$  is the weights matrix (El-naggar 2011, Olyazadeh et al. 2011). The weights matrix can be calculated through the SD ( $\sigma$ ) and the number of outputs that passed from the filtering step.

These outputs are directly proportional to the weights ( $p$ ), which are inversely proportional to the variance ( $\sigma^2$ ) (Chandra 2005).



**Figure 6.** A scheme displays the algorithm steps for producing reinitializing operation outputs (e.g., group 1)

### 3.2.4. Application stage

After optimizing the parameters in the ANN layers, the simulation is ready to produce improved accuracy. The classification outputs were fed to the final simulation for evaluating this work, and the differences between the predicted and fixed coordinates were determined. Moreover, the two stages (the classification and optimized ANN) were applied to dual-frequency processed data. Then, the differences between the coordinates of Rover2 and its known data were calculated. These differences were compared with the calculated ones between the known station and the output coordinates by applying IGS final orbits.

## 4. RESULTS AND DISCUSSION

The known coordinates for all the tested stations are supposed to be free of errors. Therefore, the comparisons carried out herein are based on the precision, not the accuracy, concept.

Results of test 1 for the five test stations are illustrated in Table 5a–e. The table shows the ANN performance in terms of MSE and also provide the SDs for X, Y, and Z coordinates that represent the prediction errors of ANN simulation. The number of times at which the ANN failed in each prediction is labeled number of fails (NOF). As shown in Table 5a–e, in the case of decimal numbers, the constellation Purelin–Purelin was the only one which provided an acceptable output, whereas the other constellations provided not a number (NaN), which means the network produced no outputs. On the other hand, there were many constellations for binary numbers providing reasonable outputs, where their best values in all test stations are highlighted by bold font. Accordingly, it can be noted that the constellation Tansig–Tansig produced better precision in  $\sigma_z$ ,  $MSE_z$ ,  $MSE_x$ , and  $MSE_y$  at Baltim, Helwan, Cairo, and Assiut stations, respectively, than the other constellations. As well, the best value of  $\sigma_x$  at Helwan station was attained by Tansig–Purelin constellation. In a general insight through the table of test 1, it can be observed that the applied binary numbers in the constellation Logsig–Purelin achieved the

best precision at all stations. Consequently, it was selected as the optimum constellation. This confirms the findings in previous investigations conducted by Zhou et al. (2017), Ordóñez Galán et al. (2013), Memarian Sorkhabi (2015), and Li (2008), where this constellation was applied as the best choice in their studies.

Table 6 shows the results of test 2 based on the SDs, 3D position error, and the elapsed time Central Processing Unit (CPU) time. The results show that the best precision in all five stations was obtained by applying the cascade forward net. Moreover, it took less time to execute the algorithm than for fit net and pattern net. This superiority could be interpreted as the strength in the connections among layers because it has additional connections from the input to every layer and from each layer to all following layers. This means there are more initialized and updated weights through the iteration number, thereby meeting the requirements in fewer epochs. Also, more equations in the BP algorithm for minimizing the errors mean the precision will be improved.

The results of test 3 are shown in Fig. 7, which represents the relation between the 3D position error and NHL at all test stations. This relation demonstrates no consistent behavior for the position error, that is, it is independent of NHL. Therefore, it could be stated that changing the NHL has no effect on the precision, and the value of NHL = 1 is suitable as an optimum value for getting an acceptable precision. Moreover, many studies have mentioned that two-layer neural networks with one hidden layer can theoretically achieve any nonlinear mapping (Wang et al. 2016, Chien-Sheng and Szu-Lin 2010, Ziggah et al. 2016b).

Fig. 8 shows the results of test 4; it indicates that the 3D position error decreases with increasing NON (hidden layer size). Nevertheless, the consistent decrease in error got fluctuated at some point in time, as the NON increased. This was realized after 20 neurons were considered for Assiut, 15 neurons for Cairo, and 10 neurons for the rest of the stations. Hence, to obtain the best precision without involving these disturbance areas, 10 neurons were chosen as the optimum value. This has been confirmed by Azami et al. (2012) and Azami and Sanei (2013), who showed that the ANN might produce unacceptable results for complex data if the NON is too low. Also, too many neurons increase the training time and may reduce the ANN performance.

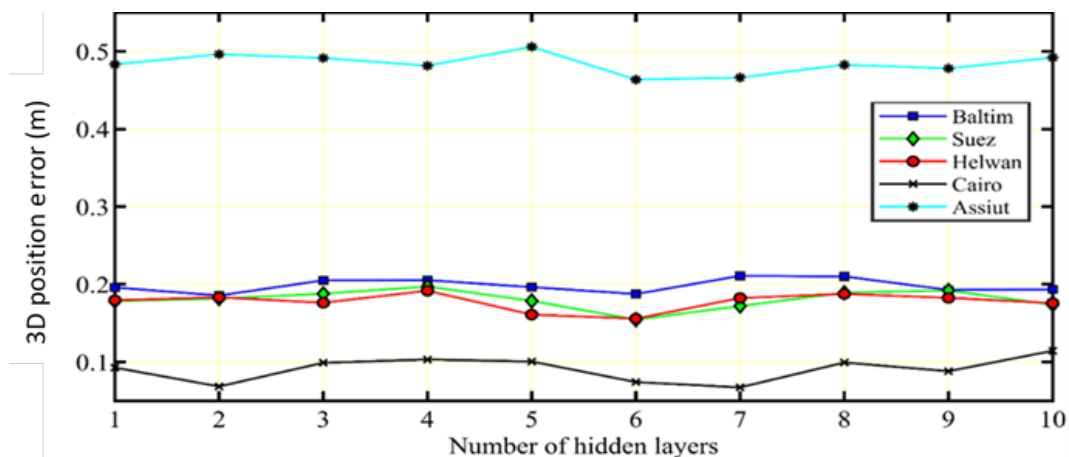


Figure 7. The prediction error represented by the 3D position error versus the number of hidden layers



**Table 5 a-e.** The effect of different transfer functions' constellation on ANN performance in the case of binary and decimal numbers (*part 1/3*)

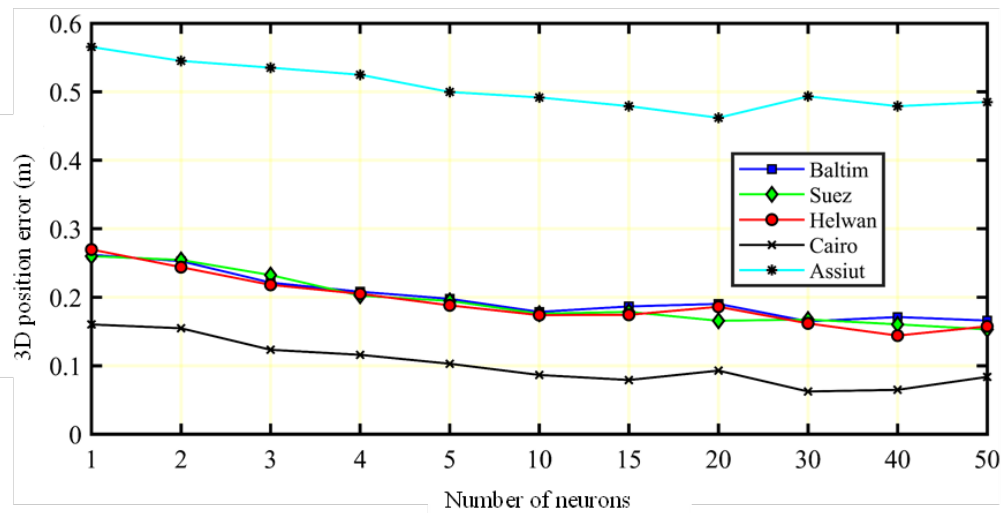
a) Baltim station	<b>Transfer functions (hidden–output) layer</b>	<b>Binary numbers</b>								
		<b>X</b>			<b>Y</b>			<b>Z</b>		
		$\sigma_x$ (m)	MSE (m)	NOF	$\sigma_y$ (m)	MSE (m)	NOF	$\sigma_z$ (m)	MSE (m)	NOF
	Tansig–Purelin	0.112	0.008	1	0.404	0.020	6	0.254	0.025	4
	Tansig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Tansig–Tansig	0.103	0.005	2	0.319	0.007	7	<b>0.050</b>	0.003	5
	Logsig–Purelin	<b>0.051</b>	<b>0.002</b>	<b>0</b>	<b>0.189</b>	<b>0.005</b>	<b>2</b>	0.057	<b>0.002</b>	<b>3</b>
	Logsig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Logsig–Tansig	0.171	0.038	3	0.495	0.085	13	0.282	0.040	19
	Purelin–Purelin	1.048	0.094	7	0.955	0.087	2	0.489	0.054	6
	Purelin–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Purelin–Tansig	0.537	0.041	7	0.805	0.049	6	0.669	0.085	7
	<b>Transfer functions (hidden–output) layer</b>	<b>Decimal numbers</b>								
<b>X, Y and Z</b>										
$\sigma_x$ (m)		$\sigma_y$ (m)		$\sigma_z$ (m)		MSE (m)		NOF		
Purelin–Purelin		0.113		0.274		0.158		$2 \times 10^{-6}$		0
The other constellations of transfer functions		NaN		NaN		NaN		NaN		100
b) Suez station	<b>Transfer functions (hidden–output) layer</b>	<b>Binary numbers</b>								
		<b>X</b>			<b>Y</b>			<b>Z</b>		
		$\sigma_x$ (m)	MSE (m)	NOF	$\sigma_y$ (m)	MSE (m)	NOF	$\sigma_z$ (m)	MSE (m)	NOF
	Tansig–Purelin	0.165	0.004	3	0.227	0.013	3	0.186	0.007	2
	Tansig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Tansig–Tansig	0.183	0.005	3	0.185	0.009	5	0.196	0.017	7
	Logsig–Purelin	<b>0.095</b>	<b>0.002</b>	<b>1</b>	<b>0.133</b>	<b>0.005</b>	<b>0</b>	<b>0.117</b>	<b>0.001</b>	<b>1</b>
	Logsig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Logsig–Tansig	0.245	0.021	7	0.401	0.005	6	0.293	0.004	6
	Purelin–Purelin	0.498	0.081	6	0.554	0.071	3	0.617	0.032	3
	Purelin–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Purelin–Tansig	0.632	0.035	6	0.488	0.045	4	0.650	0.035	10
	<b>Transfer functions (hidden–output) layer</b>	<b>Decimal numbers</b>								
<b>X, Y and Z</b>										
$\sigma_x$ (m)		$\sigma_y$ (m)		$\sigma_z$ (m)		MSE (m)		NOF		
Purelin–Purelin		0.171		0.206		0.256		$3 \times 10^{-8}$		1
The other constellations of transfer functions		NaN		NaN		NaN		NaN		100

**Table 5 a-e.** The effect of different transfer functions' constellation on ANN performance in the case of binary and decimal numbers (*part 2/3*)

c) Helwan station	<b>Transfer functions (hidden–output) layer</b>	<b>Binary numbers</b>								
		<b>X</b>			<b>Y</b>			<b>Z</b>		
		$\sigma_x$ (m)	MSE (m)	NOF	$\sigma_y$ (m)	MSE (m)	NOF	$\sigma_z$ (m)	MSE (m)	NOF
	Tansig–Purelin	<b>0.042</b>	0.007	3	0.156	0.008	3	0.123	0.009	3
	Tansig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Tansig–Tansig	0.065	0.005	5	0.192	0.012	5	0.099	<b>0.004</b>	5
	Logsig–Purelin	0.048	<b>0.004</b>	<b>2</b>	<b>0.129</b>	<b>0.002</b>	<b>3</b>	<b>0.086</b>	0.010	<b>2</b>
	Logsig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Logsig–Tansig	0.275	0.013	2	0.352	0.007	6	0.356	0.032	6
	Purelin–Purelin	0.520	0.027	4	0.581	0.035	4	0.492	0.009	7
	Purelin–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Purelin–Tansig	0.452	0.008	5	0.611	0.041	5	0.470	0.036	5
	<b>Transfer functions (hidden–output) layer</b>	<b>Decimal numbers</b>								
		<b>X, Y and Z</b>								
$\sigma_x$ (m)		$\sigma_y$ (m)		$\sigma_z$ (m)		MSE (m)		NOF		
0.157		0.305		0.276		$5 \times 10^{-7}$		2		
NaN		NaN		NaN		NaN		100		
d) Cairo station	<b>Transfer functions (hidden–output) layer</b>	<b>Binary numbers</b>								
		<b>X</b>			<b>Y</b>			<b>Z</b>		
		$\sigma_x$ (m)	MSE (m)	NOF	$\sigma_y$ (m)	MSE (m)	NOF	$\sigma_z$ (m)	MSE (m)	NOF
	Tansig–Purelin	0.143	0.007	4	0.031	0.010	3	0.133	0.008	2
	Tansig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Tansig–Tansig	0.127	<b>0.001</b>	3	0.108	0.008	4	0.186	0.015	3
	Logsig–Purelin	<b>0.073</b>	0.004	<b>2</b>	<b>0.014</b>	<b>0.006</b>	<b>2</b>	<b>0.052</b>	<b>0.003</b>	<b>2</b>
	Logsig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Logsig–Tansig	0.272	0.009	5	0.358	0.031	5	0.296	0.002	5
	Purelin–Purelin	0.382	0.012	4	0.399	0.023	3	0.488	0.047	7
	Purelin–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100
	Purelin–Tansig	0.484	0.023	6	0.511	0.045	5	0.388	0.071	4
	<b>Transfer functions (hidden–output) layer</b>	<b>Decimal numbers</b>								
		<b>X, Y and Z</b>								
$\sigma_x$ (m)		$\sigma_y$ (m)		$\sigma_z$ (m)		MSE (m)		NOF		
0.273		0.242		0.338		0.006		0		
NaN		NaN		NaN		NaN		100		

**Table 5 a-e.** The effect of different transfer functions' constellation on ANN performance in the case of binary and decimal numbers (*part 3/3*)

e) Assiut station	Transfer functions (hidden–output) layer	Binary numbers								
		X			Y			Z		
		$\sigma_x$ (m)	MSE (m)	NOF	$\sigma_y$ (m)	MSE (m)	NOF	$\sigma_z$ (m)	MSE (m)	NOF
	Tansig–Purelin	0.219	0.008	3	0.327	0.010	1	0.530	0.016	6
Tansig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100	
Tansig–Tansig	0.177	<b>0.004</b>	4	0.294	0.013	2	0.549	0.017	5	
Logsig–Purelin	<b>0.149</b>	0.006	<b>2</b>	<b>0.225</b>	<b>0.007</b>	<b>0</b>	<b>0.481</b>	<b>0.015</b>	<b>5</b>	
Logsig–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100	
Logsig–Tansig	0.314	0.009	5	0.470	0.023	4	0.640	0.022	7	
Purelin–Purelin	0.433	0.014	4	0.402	0.036	2	0.691	0.035	14	
Purelin–Logsig	NaN	NaN	100	NaN	NaN	100	NaN	NaN	100	
Purelin–Tansig	0.503	0.071	4	0.630	0.610	0	0.747	0.051	15	
Transfer functions (hidden–output) layer	Decimal numbers									
	X, Y and Z									
	$\sigma_x$ (m)	$\sigma_y$ (m)	$\sigma_z$ (m)	MSE (m)	NOF					
	Purelin–Purelin	0.259	0.301	0.606	0.003	7				
The other constellations of transfer functions	NaN	NaN	NaN	NaN	100					



**Figure 8.** The prediction error represented by the 3D position error versus number of neurons

Results of the NOI test (test 5) are shown in Fig. 9 and Table 7. Each group's output was represented by the Root Mean Square Error (RMSE) of X, Y, and Z and the 3D position error with the group values. Not only the group values were compared, but also the groups. From each group's results, as shown in Fig. 9, it can be noticed that there is a slight improvement in precision with increasing NOI. However, this improvement became clearer when that

comparison was performed within the groups (i.e., in the vertical direction through Fig. 9). This can be confirmed by the results of Table 7, which represents the average values of SDs and the 3D position error for each group at each test station. These results reveal that the parameter NOI is more effective on the precision if it is supplied to ANN as a group rather than as a value.

It is noteworthy that Table 7 illustrates the execution time for each group, which is associated with specific precision. It means that selecting a particular group's precision for performing any work depends on the required time for that work. As a general insight for all tests, it can be observed that Assiut station provided the worst precision. This could be due to the baseline length, which is the longest one among all test stations' baselines. Thus, according to Ögütçü and Kalaycı (2018), Dogan et al. (2018), and Dawson and Woods (2010), the errors' magnitude will be the highest. After all parameters' tests, the final ANN simulation (Fig. 10) was designed based on the optimum values obtained from the results.

Tables 8 and 9 display the application stage results for single- and dual-frequency observations, respectively. As shown in these tables, the coordinates deviations ( $dX$ ,  $dY$ ,  $dZ$ ) and the 3D position error were calculated for each test station. The improvement that occurred due to the application of classification and ANN algorithms in the two cases of observations, as well as the outputs of IGS final orbits in the case of dual-frequency observations were determined based on the 3D position error and presented in Table 10. Analyzing the results in Table 9 shows that applying the classification algorithm improved the single- and dual-frequency observations by 39% and 44%, respectively. Obviously, this improvement increased to 79% and 72% for single- and dual-frequency observations, respectively, when the ANN algorithm was applied.

**Table 6.** The SD of X, Y, and Z coordinates, 3D position error, and elapsed time for three different types of ANN

Station	Pattern net					Fit net					Cascade forward net				
	$\sigma_x$ (m)	$\sigma_Y$ (m)	$\sigma_Z$ (m)	$\sigma_P$ (m)	Elapsed time (s)	$\sigma_x$ (m)	$\sigma_Y$ (m)	$\sigma_Z$ (m)	$\sigma_P$ (m)	Elapsed time (s)	$\sigma_x$ (m)	$\sigma_Y$ (m)	$\sigma_Z$ (m)	$\sigma_P$ (m)	Elapsed time (s)
<b>Baltim</b>	0.090	0.263	0.071	0.287	44.4	0.056	0.219	0.030	0.228	38.8	0.043	0.188	0.021	0.194	27.9
<b>Suez</b>	0.078	0.118	0.157	0.211	47.3	0.072	0.101	0.106	0.163	36.2	0.077	0.099	0.103	0.162	24.4
<b>Helwan</b>	0.127	0.193	0.089	0.248	45.8	0.089	0.152	0.076	0.192	37.4	0.084	0.146	0.079	0.186	27.2
<b>Cairo</b>	0.161	0.068	0.118	0.211	46.9	0.115	0.044	0.066	0.140	35.1	0.084	0.029	0.048	0.101	28.9
<b>Assiut</b>	0.227	0.224	0.448	0.550	49.7	0.179	0.198	0.431	0.507	39.5	0.147	0.179	0.425	0.484	26.8

**Table 7.** The averages of SDs in the directions of the coordinate axes, 3D position error, and elapsed time for the four groups of initialization numbers

Station	Group 1				Group 2				Group 3				Group 4			
	Nt values		Elapsed time (h)		Nt values		Elapsed time (h)		Nt values		Elapsed time (h)		Nt values		Elapsed time (h)	
	1–10		0.02		10–100		0.15		100–1000		1.5		1000–10,000		15.5	
	$\sigma_{m(X)}$ (m)	$\sigma_{m(Y)}$ (m)	$\sigma_{m(Z)}$ (m)	$\sigma_{m(P)}$ (m)	$\sigma_{m(X)}$ (m)	$\sigma_{m(Y)}$ (m)	$\sigma_{m(Z)}$ (m)	$\sigma_{m(P)}$ (m)	$\sigma_{m(X)}$ (m)	$\sigma_{m(Y)}$ (m)	$\sigma_{m(Z)}$ (m)	$\sigma_{m(P)}$ (m)	$\sigma_{m(X)}$ (m)	$\sigma_{m(Y)}$ (m)	$\sigma_{m(Z)}$ (m)	$\sigma_{m(P)}$ (m)
<b>Baltim</b>	0.080	0.195	0.040	0.215	0.068	0.174	0.036	0.191	0.042	0.137	0.019	0.145	0.017	0.080	0.009	0.083
<b>Suez</b>	0.094	0.128	0.105	0.190	0.087	0.119	0.101	0.179	0.068	0.085	0.082	0.137	0.045	0.035	0.044	0.074
<b>Helwan</b>	0.067	0.177	0.086	0.208	0.058	0.142	0.067	0.168	0.052	0.107	0.049	0.129	0.034	0.065	0.030	0.080
<b>Cairo</b>	0.083	0.044	0.035	0.100	0.066	0.036	0.036	0.084	0.047	0.030	0.036	0.066	0.026	0.023	0.024	0.043
<b>Assiut</b>	0.184	0.216	0.572	0.639	0.164	0.196	0.453	0.520	0.134	0.172	0.350	0.413	0.103	0.125	0.249	0.297

**Table 8.** The differences in coordinates between the known points and the output data, and the position errors for the three main stages in the case of GNSS- single frequency

Station	Post-processing (broadcast ephemerides)				Classification algorithm				ANN algorithm			
	dX (m)	dY (m)	dZ (m)	Position error (m)	dX (m)	dY (m)	dZ (m)	Position error (m)	dX (m)	dY (m)	dZ (m)	Position error (m)
<b>Baltim</b>	0.118	0.309	0.046	0.334	0.090	0.220	0.047	0.242	0.031	0.06	0.021	0.071
<b>Suez</b>	0.145	0.262	0.134	0.328	0.104	0.141	0.112	0.208	0.052	0.054	0.063	0.098
<b>Helwan</b>	0.123	0.277	0.204	0.365	0.075	0.190	0.092	0.224	0.078	0.081	0.051	0.123
<b>Cairo</b>	0.093	0.045	0.054	0.117	0.081	0.046	0.031	0.098	0.042	0.027	0.030	0.058
<b>Assiut</b>	0.286	0.330	0.992	1.084	0.190	0.241	0.675	0.741	0.101	0.132	0.381	0.416

**Table 9.** The coordinates' differences between the known points and the output data, and the position errors for the three main stages and IGS final orbits in the case of GNSS- dual frequency

Station	Post-processing (broadcast ephemerides)				Post-processing (final orbits)				Classification algorithm				ANN algorithm			
	dX (m)	dY (m)	dZ (m)	Position error (m)	dX (m)	dY (m)	dZ (m)	Position error (m)	dX (m)	dY (m)	dZ (m)	Position error (m)	dX (m)	dY (m)	dZ (m)	Position error (m)
<b>Baltim</b>	0.050	0.020	0.003	0.054	0.001	0.002	0.004	0.005	0.034	0.011	0.009	0.037	0.011	0.007	0.008	0.015
<b>Suez</b>	0.006	0.003	0.006	0.009	0.003	0.007	0.003	0.008	0.003	0.002	0.004	0.005	0.002	0.003	0.004	0.005
<b>Helwan</b>	0.047	0.040	0.019	0.065	0.015	0.015	0.011	0.024	0.024	0.021	0.024	0.040	0.017	0.016	0.021	0.031
<b>Cairo</b>	0.022	0.014	0.013	0.029	0.015	0.010	0.019	0.026	0.017	0.012	0.015	0.026	0.011	0.007	0.012	0.018
<b>Assiut</b>	0.157	0.051	0.322	0.362	0.044	0.083	0.110	0.145	0.103	0.062	0.220	0.251	0.051	0.021	0.153	0.163

**Table 10.** Precision improvement due to applying classification and ANN algorithms in the two cases of observations

<b>Percentage of improvement (%)</b>					
<b>Station</b>	<b>Single-frequency observations</b>		<b>Dual-frequency observations</b>		
	<b>Classification algorithm</b>	<b>ANN algorithm</b>	<b>Classification algorithm</b>	<b>ANN algorithm</b>	<b>IGS final orbits</b>
<b>Baltim</b>	28	79	31	72	91
<b>Suez</b>	37	70	44	<b>44</b>	11
<b>Helwan</b>	39	66	38	52	63
<b>Cairo</b>	16	50	10	<b>38</b>	10
<b>Assiut</b>	32	62	31	55	60

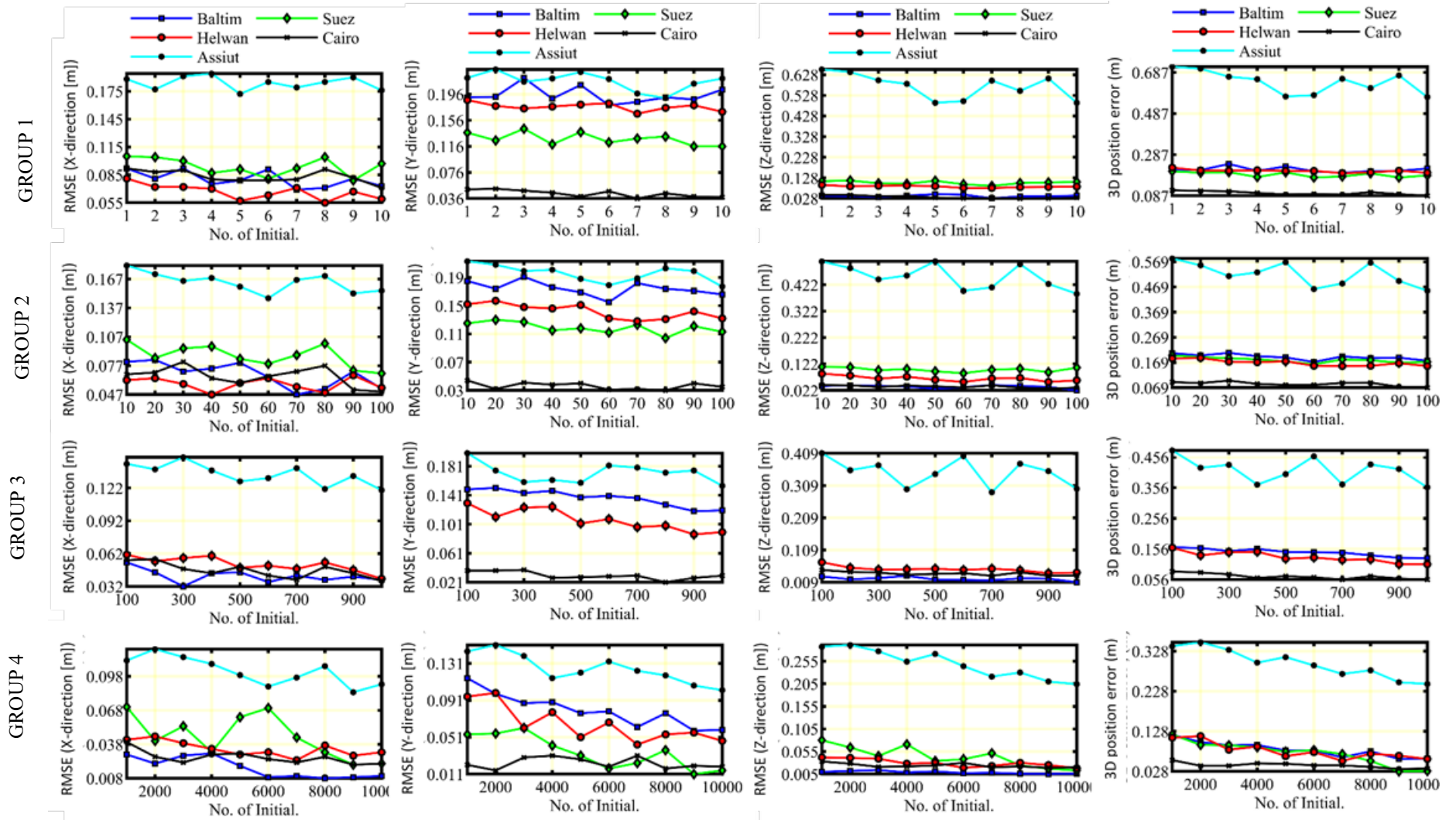


Figure 9. RMSE of X, Y, and Z directions and 3D position error according to the different groups of the number of initializations



Furthermore, comparison of the obtained improvement due to the ANN algorithm with that produced by IGS final orbits shows that the ANN algorithm is close to IGS final orbits' outputs. Notwithstanding, the improvement due to the ANN algorithm at Suez and Cairo stations is better than that of IGS's final orbits. This superiority can be interpreted as the accuracy of broadcast observations. In other words, when the observations are collected through the best conditions, the broadcast outputs will be more accurate, thereby improving the ANN outputs. There is an anomaly at Suez station for dual-frequency observations, where the ANN algorithm provided a percentage of improvement similar to the classification algorithm. According to Tables 8–10, it can be stated that the proposed integration of satellite configuration and ANN has the capability for providing a significant improvement in the single-frequency GNSS observations in real time, as well as achieving precision in the dual-frequency GNSS observations close to that of IGS final orbits.

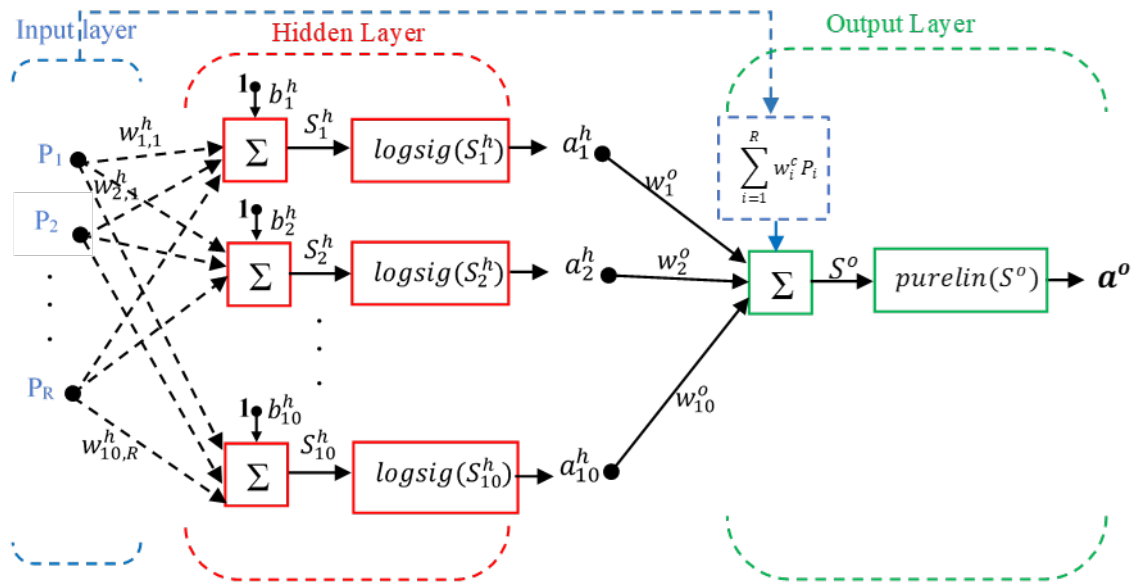


Figure 10. The ANN designed from the results; its type is cascade forward net

## 5. CONCLUSIONS

Improving the accuracy of single-frequency GNSS observations in the absence of final IGS orbits is the essential aim of this study. An algorithm was formulated based on the ANN approach, considering realizing the optimum performance at each ANN layer to achieve this goal. Furthermore, a classification stage of GNSS observations was conducted by implementing a designed algorithm to produce the input layer's optimal values of coordinates. This research found that the numbers in binary format give more accurate results than those in decimal format. For the hidden layer, there are validation tests performed to determine the best variables with high accuracy. However, the results revealed that increasing the NHL has not provided any significant improvement. So, the  $\text{NHL} = 1$  was selected as the optimal value, while the  $\text{NON} = 10$  provided an acceptable precision. Besides, the results of selecting the transfer functions' constellation in the hidden and output layers demonstrated that Logsig–Purelin constellation is the most accurate. Again, ANN type testing indicated that the cascade forward net is more flexible than the fit net and the pattern net, which produced an acceptable precision with less execution time. The last test for the hidden layer variables is the NOI test, which represents the most influential variable on accuracy. This test was performed through four groups of data, and the results showed that each group produced a certain precision at a specific execution time.

Noticeably, the data trend indicated a positive relationship between the NOI and precision in terms of SDs. Interestingly, it has been observed that the improvement and the best precision are achieved if the ANN is set at a group period rather than a specific value of the NOI. Furthermore, the output layer's outputs were filtered by removing the outliers before applying the LSM to estimate the best values. It is important to note that the simulation became too ready for prediction operation after applying the obtained optimal variables. As it is known, the DGNS technique eliminates the systematic biases such as ionospheric delay, but for the long baselines such as the Assiut station, this error caused the lowest quality of the results. The simulation results improved by 79% and 72% in single- and dual-frequency observations, respectively, in the 3D position, while the results from the final IGS orbit showed 91% improvement. It is essential to mention that applying the two stages, classification and ANN, on dual-frequency observations showed better improvement than IGS final orbits' outputs at two test stations. According to the results of this study, it can be concluded that the precision of the IGS DGNS in the two cases of observations excessively increased in real time due to the proposed integration. As well, the results proved that implementing ANN with its optimal variables represented an important role in GNSS applications. This study fulfilled these results in the study conditions; accordingly, it is recommended to apply it in other conditions to decide if this investigation can be applied for all conditions all over the world.

## REFERENCES

- Ansari, K., O. Corumluoglu, P. Verma & V. S. Pavelyev (2017) An overview of the international GNSS service (IGS). *Grenze International Journal of Computer Theory and Engineering*.
- Azami, H. & S. Sanei (2013) GPS GDOP classification via improved neural network trainings and principal component analysis. *International Journal of Electronics*, 101, 1300-1313.
- Azami, H., S. Sanei & H. Alizadeh. 2012. *GPS GDOP Classification via Advanced Neural Network Training*.
- Azami, H., M. D. Soltani & I. Tavakkolnia. 2016. An intelligent approach for classification of GPS satellites based on neural network, genetic algorithm and particle swarm optimization. In *2016 International Conference for Students on Applied Engineering (ICSAE)*, 70-74.
- Azarbad, M., H. Azami, S. Sanei & A. Ebrahimzadeh (2014) New Neural Network-based Approaches for GPS GDOP Classification based on Neuro-Fuzzy Inference System, Radial Basis Function, and Improved Bee Algorithm. *Applied Soft Computing*, 25, 285-292.
- Barrile, V., G. M. Meduri, G. Bilotta & U. M. Trungadi (2016) GPS- GIS and Neural Networks for Monitoring Control, Cataloging the Prediction and Prevention in Tectonically Active Areas. *Procedia - Social and Behavioral Sciences*, 223, 909-914.
- Beale, H. D., H. B. Demuth & M. Hagan (1996) Neural network design. *Pws, Boston*.
- Bear, J., A. H.-D. Cheng, S. Sorek, D. Ouazar & I. Herrera. 1999. *Seawater intrusion in coastal aquifers: concepts, methods and practices*. Springer Science & Business Media.
- Biswas, S. K., L. Qiao & A. G. Dempster (2017) Effect of PDOP on performance of Kalman Filters for GNSS-based space vehicle position estimation. *GPS Solutions*, 21, 1379-1387.
- Chandra, A. 2005. *Surveying: Problem Solving with theory and objective type questions*. New Age International.
- Chien-Sheng, C. & S. Szu-Lin (2010) Resilient Back-propagation Neural Network for Approximation 2-D GDOP. *Lecture Notes in Engineering and Computer Science*, 2181.

- Correa Muñoz, N. A. & L. A. Cerón-Calderón (2018) Precision and accuracy of the static GNSS system for surveying networks used in Civil Engineering. *Ingeniería e Investigación*, 38, 52-59.
- Davidovic, M. & N. Mijic (2017) ANALYSIS OF THE INFLUENCE OF SATELLITES CONSTELLATION IN GNSS POSITIONING ACCURACY. *International Journal of Engineering*, 15, 141-148.
- Dawson, J. & A. Woods (2010) ITRF to GDA94 coordinate transformations. *Journal of Applied Geodesy*, 4.
- Dogan, A. H., N. Tunalioglu, B. Erdogan & T. Ocalan (2018) Evaluation of the GPS Precise Point Positioning technique during the 21 July 2017 Kos-Bodrum (East Aegean Sea) Mw 6.6 earthquake. *Arabian Journal of Geosciences*, 11.
- El-naggar, A. M. (2011) Enhancing the accuracy of GPS point positioning by converting the single frequency data to dual frequency data. *Alexandria Engineering Journal*, 50, 237-243.
- El-Tokhey, M., Y. M. Mogahed, M. Mamdouh & T. W. Hassan (2018) Establishment of New Continuous Operating Reference Station (CORS) at Faculty of Engineering, Ain Shams University. *International Journal of Engineering and Advanced Technology (IJEAT)*, ISSN, 2249-8958.
- El Manaily, E., M. Abd Rabbou, A. El-Shazly & M. Baraka (2017) Evaluation of Quad-Constellation GNSS Precise Point Positioning in Egypt. *Artificial Satellites*, 52, 9-18.
- Elshambaky, H. T. (2018) Application of neural network technique to determine a corrector surface for global geopotential model using GPS/levelling measurements in Egypt. *Journal of Applied Geodesy*, 12, 29-43.
- Elshambaky, H. T., M. R. Kaloop & J. W. Hu (2018) A novel three-direction datum transformation of geodetic coordinates for Egypt using artificial neural network approach. *Arabian Journal of Geosciences*, 11.
- Farah, A. (2017) Variation of Static-PPP Positioning Accuracy Using GPS-Single Frequency Observations (Aswan, Egypt). *Artificial Satellites*, 52, 19-26.
- Gera, G., R. Cossu, C. S. Regazzoni & L. Bruzzone (2011) A neural network based algorithm to performances enhancement of GNSS receivers in a multipath environment. *Isip40 It*, 45, 1130-1137.
- Ghoddousi-Fard, R. & P. Dare (2005) Online GPS processing services: an initial study. *GPS Solutions*, 10, 12-20.
- Han, S., Q. Gui, G. Li & Y. Du (2014) Minimum of PDOP and its applications in inter-satellite links (ISL) establishment of Walker- $\delta$  constellation. *Advances in Space Research*, 54, 726-733.
- Haykin, S. S. 2009. *Neural networks and learning machines*/Simon Haykin. New York: Prentice Hall.
- Heßelbarth, A. & L. Wanninger (2010) Performance of GNSS-PPP in Post-Processing Mode. *Rostock-Warnemünde, Germany*, 02-05.
- Inal, C., S. Bulbul & B. Bilgen (2018) Statistical analysis of accuracy and precision of GNSS receivers used in network RTK. *Arabian Journal of Geosciences*, 11.
- Jamieson, M. & D. T. Gillins (2018) Comparative analysis of online static GNSS postprocessing services. *Journal of surveying engineering*, 144, 05018002.

- Janowski, A. & J. Rapinski (2016) The Analyzes of PDOP Factors for a Zigbee Ground - Based Augmentation Systems. *Polish Maritime Research*, 24, 108-114.
- Jwo, D.-J. & C.-C. Lai (2006) Neural network-based GPS GDOP approximation and classification. *GPS Solutions*, 11, 51-60.
- Kaygısız, B. H., A. M. Erkmen & İ. Erkmen (2007) Enhancing positioning accuracy of GPS/INS system during GPS outages utilizing artificial neural network. *Neural Processing Letters*, 25, 171-186.
- Kim, M. & J. Kim (2015) Predicting IGS RTS Corrections Using ARMA Neural Networks. *Mathematical Problems in Engineering*, 2015, 1-11.
- Lemmon, T. R. & G. P. Gerdan (1999) The influence of the number of satellites on the accuracy of RTK GPS positions. *Australian Surveyor*, 44, 64-70.
- Li, X. (2008) Comparing the Kalman filter with a Monte Carlo-based artificial neural network in the INS/GPS vector gravimetric system. *Journal of Geodesy*, 83, 797-804.
- Malleswaran, M., V. Vaidehi & N. Sivasankari (2014) A novel approach to the integration of GPS and INS using recurrent neural networks with evolutionary optimization techniques. *Aerospace Science and Technology*, 32, 169-179.
- Martín, A., A. B. Anquela, R. Capilla & J. L. Berné (2011) PPP Technique Analysis Based on Time Convergence, Repeatability, IGS Products, Different Software Processing, and GPS+GLONASS Constellation. *Journal of Surveying Engineering*, 137, 99-108.
- McGaughey, R. J., K. Ahmed, H.-E. Andersen & S. E. Reutebuch (2017) Effect of occupation time on the horizontal accuracy of a mapping-grade GNSS receiver under dense forest canopy. *Photogrammetric Engineering & Remote Sensing*, 83, 861-868.
- Memarian Sorkhabi, O. (2015) Geoid determination based on log sigmoid function of artificial neural networks:(a case study: Iran). *Journal of Artificial Intelligence in Electrical Engineering*, 3, 18-24.
- Mohamed, H. F. (2015) Assessment of Factors Influencing Static GNSS Precise Point Positioning: A case study in Egypt. *International Journal of Applied Science and Engineering Research*, 4, 692-701.
- Mohammed, F. M., S. A. Aziez & H. N. Abdul-Rihda (2015) Comparison between wavelet and radial basis function neural networks for GPS prediction. *Engineering and Technology Journal*, 33, 560-572.
- Mosavi, M. R. & K. Mohammadi. 2002. Improve the position accuracy on low cost GPS receiver with adaptive neural networks. In *Student Conference on Research and Development*, 322-325.
- Nie, Z., Y. Gao, Z. Wang & S. Ji (2016) A new method for satellite selection with controllable weighted PDOP threshold. *Survey Review*, 49, 285-293.
- Öğütçü, S. & İ. Kalaycı (2018) Accuracy and precision of network-based RTK techniques as a function of baseline distance and occupation time. *Arabian Journal of Geosciences*, 11.
- Olyazadeh, R., H. Setan & N. Fouladinejad. 2011. Network Adjustment Program using MATLAB. In *Geospatial World Forum*.
- Ordóñez Galán, C., J. R. Rodríguez Pérez, S. García Cortés & A. Bernardo Sánchez (2013) Analysis of the influence of forestry environments on the accuracy of GPS measurements by means of recurrent neural networks. *Mathematical and Computer Modelling*, 57, 2016-2023.

- Rabbou, M. A. & A. El-Rabbany (2015) Precise Point Positioning using Multi-Constellation GNSS Observations for Kinematic Applications. *Journal of Applied Geodesy*, 9.
- Specht, C., M. Mania, M. Skóra & M. Specht (2015) Accuracy Of The GPS Positioning System In The Context Of Increasing The Number Of Satellites In The Constellation. 22, 9.
- Tafazoli, S., M. Mosavi & N. Rahemi (2011) Comparing GPS GDOP Approximation Accuracy using Recurrent Wavelet Neural Network and ARMA Modeling. *JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGIES*, 1.
- Tafazoli, S. & M. R. Mosavi (2011) Performance Improvement of GPS GDOP Approximation Using Recurrent Wavelet Neural Network. *Journal of Geographic Information System*, 03, 318-322.
- Teng, Y. & J. Wang (2015) Some Remarks on PDOP and TDOP for Multi-GNSS Constellations. *Journal of Navigation*, 69, 145-155.
- Teunissen, P. J. G., R. Odolinski & D. Odijk (2013) Instantaneous BeiDou+GPS RTK positioning with high cut-off elevation angles. *Journal of Geodesy*, 88, 335-350.
- Wang, Y., Z. Lu, Y. Qu, L. Li & N. Wang (2016) Improving prediction performance of GPS satellite clock bias based on wavelet neural network. *GPS Solutions*, 21, 523-534.
- Yang, Z., Y. Wang & O. Xu (2015) Improvement of the Positioning Accuracy in GNSS Positioning Based on Neural Network. *Lecture Notes in Electrical Engineering*, 334, 505-511.
- Yousef, M. A. & M. K. Ragheb (2014) Effect of Recording Interval on GPS Accuracy. *Journal of Engineering Sciences, Assiut University*, 42, 1215-1231.
- Zhang, G. P. (2000) Neural networks for classification: a survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 30, 451-462.
- Zhou, Y., J. Wan, Z. Li & Z. Song. 2017. GPS/INS integrated navigation with BP neural network and Kalman filter. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2515-2520.
- Ziggah, Y. Y., H. Youjian, A. Tierra, A. A. Konaté & Z. Hui (2016a) Performance evaluation of artificial neural networks for planimetric coordinate transformation—a case study, Ghana. *Arabian Journal of Geosciences*, 9.
- Ziggah, Y. Y., H. Youjian, X. Yu & L. P. Basommi (2016b) Capability of Artificial Neural Network for Forward Conversion of Geodetic Coordinates  $(\phi, \lambda, h)$  to Cartesian Coordinates (X, Y, Z). *Mathematical Geosciences*, 48, 687-721.

**Received:** 2021-07-07

**Reviewed:** 2021-11-16 (undisclosed name); 2021-12-10 (undisclosed name)

**Accepted:** 2022-02-02