Mathematical and Computer Modelling 57 (2013) 2016–2023

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Analysis of the influence of forestry environments on the accuracy of GPS measurements by means of recurrent neural networks

C. Ordóñez Galán [a,b,*], J.R. Rodríguez Pérez [c], S. García Cortés [a], A. Bernardo Sánchez [d]

[a] Department of Mining Exploitation, University of Oviedo, 33004 Oviedo, Spain
[b] Department of Natural Resources and Environmental Engineering, University of Vigo, 36310 Vigo, Spain
[c] ESTIA, University of León, 24400 Ponferrada, León, Spain
[d] Department of Manufacturing Engineering, University of Oviedo, 33600 Oviedo, Spain

ARTICLE INFO

Article history:
Received 2 October 2011
Received in revised form 3 February 2012
Accepted 5 March 2012

Keywords:
GPS accuracy
Forest canopy
Recurrent neural network
Regression model

ABSTRACT

The present paper analyses the accuracy of the measurements performed by a global positioning system (GPS) receiver located in forested environments. A large set of observations were taken with a GPS receiver at intervals of one second during a total time of an hour at twelve different points placed in forest areas. Each of these areas was characterized by a set of forest stand variables (tree density, volume of wood, Hart-Becking index, etc.) The influence on the accuracy of the measurements of other variables related to the GPS signal, such as the position dilution of precision (PDOP), the signal-to-noise ratio and the number of satellites, was also studied.

Recurrent neural networks (RNNs) were applied to build a mathematical model that associates the observation errors and the GPS signal and forest stand variables. A recurrent neural network is a type of neural network whose topology allows it to exhibit dynamic temporal behaviour. This property, and its utility for discovering patterns in non-linear and chaotic systems, make the RNN a suitable tool for the study of our problem.

Two kinds of models with different numbers of input variables were built. The results obtained are in line with those achieved by the authors in previous research using different techniques; they showed that the variables with the highest influence on the accuracy of the GPS measurements are those related to the forest canopy, that is, the forest variables. The performance of the models of the RNN improved on previous results obtained with other techniques.

1. Introduction

As is well-known, the *global positioning system* (GPS) is a space-based global navigation satellite system that provides reliable location and time information in all weathers and at all times, anywhere on or near the Earth when and where there is an unobstructed line of sight to four or more GPS satellites [1]. The GPS system was established in 1973 by the US Department of Defense (DOD) to overcome the limitations of previous navigation systems [2,3]. GPS satellites broadcast signals from space, which each GPS receiver uses to calculate its three-dimensional location (latitude, longitude, and altitude) plus the current time [4].

Although GPS was initially conceived to be used in open spaces, in practice many users operate GPS receivers in conditions such as forest environments where the signal reception is not so favourable. According to previous research, the presence

* Corresponding author at: Department of Mining Exploitation, University of Oviedo, 33004 Oviedo, Spain. Tel.: +34 98548027; fax: +34 985104245.
E-mail addresses: ordonezcelestino@uniovi.es, cgalan@uvigo.es, cgalan.uniovi@gmail.com (C. Ordóñez Galán), jr.rodriguez@unileon.es (J.R. Rodríguez Pérez), sgcortes@uniovi.es (S. García Cortés), abernardo@uniovi.es (A. Bernardo Sánchez).

0895-7177/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2012.03.006

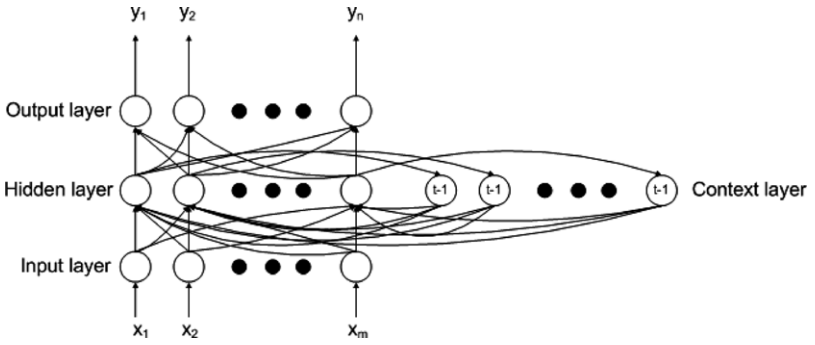

Fig. 1. Schematic representation of the Elman recurrent neural network.

of an overhead canopy may degrade the positional precision by one order of magnitude [5]. In the present work the impact of forest canopy on quality and accuracy of GPS measurements by means of recurrent neural networks is analysed with success. It must be stressed that the aim of the present research is to build a recurrent neural network able to predict the value of the variables H_{acc} and V_{acc} (horizontal and vertical accuracies) using as input variables all those listed in Section 4. This paper is in line with previous research focused on the use of GPS under a forest canopy [6–10].

2. The aim of the present research

As has been stated above, the aim of the present research is the prediction of the vertical and horizontal accuracies of the coordinates and the determination of which of the variables have the most important influence over them. The input variables that are used are the dasometric characteristics of the fields (i.e. arithmetic mean diameter, average height, crown height, stand density, etc.) together with the GPS signal variables (position dilution of precision, X accuracy, Y accuracy, vertical accuracy, horizontal accuracy, etc.) which are taken into account. The above-mentioned output variables and horizontal and vertical accuracies were calculated for each sample through the following expressions:

$$H_{acc} = \sqrt{(E_i - E_{true})^2 + (N_i - N_{true})^2} \tag{1}$$

$$V_{acc} = |Z_i - Z_{true}| \tag{2}$$

where H_{acc} and V_{acc} indicate horizontal and vertical accuracies, respectively. E_i, N_i and Z_i are the measured positions at the ith second, and E_{true}, N_{true} and Z_{true} are the true positions along the easting, northing and ellipsoidal height directions, respectively.

3. The mathematical model

3.1. Recurrent neural networks (RNNs)

In recent years, recurrent neural network techniques have been applied to a wide variety of problems. Such neural networks can be divided into two main categories: fully recurrent and partially recurrent neural networks. For the purposes of the present research, a kind of partially recurrent neural network called an Elman network [11] will be employed. Simple partially recurrent neural networks were introduced in the late 1980s [12] for learning strings of characters. Many other applications have been developed since then, and they can now be found in many different research areas such as linguistics [13], communication systems [14], electrical power load prediction [15] and stock market forecasting [16].

An Elman RNN is a network with an initial configuration based on a regular feedforward neural network. As is well-known, in a feedforward neural network the information moves in only one direction, forward, from the input nodes, through the hidden nodes and to the output nodes without cycles or loops. This is the main difference between the feedforward neural network and the Elman network, because the latter has a layer called the context layer. The neurons in the context layer, called context neurons, hold a copy of the outputs that are given by the neurons of the hidden layer to the output layer. This means that in the following computing step, information that was given as an output by the hidden layer is used as a new input information for this layer. Fig. 1 represents the architecture of the Elman recurrent neural network (please note that the neurons of the context layer are signalled with $t-1$ inside in order to remark that, in the time t, they contain the information generated by the hidden layer in the time $(t-1)$).

In the same way as in other neural network models, the strength of the relationships between neurons in an Elman RNN are indicated by weights. For this kind of neural network the weight values of the neurons are chosen randomly and their values are changed during the training in order to optimize them. There is only one exception to this rule: for the weights

from the hidden layer to the context layer, whose values are set to 1 [11,17] and which do not change during the training process because the values of the context neurons need to receive the output information from the hidden layer as it is calculated.

Let the $[q \times 1]$ vector $x(t)$ be the state of a discrete non-linear system and the vector $u(t)$ the input, while the vector $y(t)$ of dimension $[p \times 1]$ is the output. Then the following equations represent a dynamic system with m inputs and p outputs of order q:

$$X(t+1) = \varphi(W_a \cdot X(t) + W_b \cdot u(t)) \tag{3}$$

$$y(t) = C \cdot x(t). \tag{4}$$

Please note that W_a is a squared matrix of dimension q while W_b is a $[q \times m]$ dimension matrix and the dimension of C is $[p \times q]$. W_a contains the synaptic weights of the q processing neurons which are connected to the feedback nodes in the input layer. In the same way, W_b represents the synaptic weights for each one of the q neurons that are connected to the input neuron and C defines the combination of neurons that will characterize the neural network output. The non-linear function $\varphi(.)$ characterizes the activation function.

The two main activation functions used in current applications are the hyperbolic tangent function (tansig) and the sigmoid function. They can be described from the mathematical point of view as follows, respectively:

$$\varphi(t) = \tanh(v) \tag{5}$$

$$\varphi(t) = (1 + e^{-t})^{-1} \tag{6}$$

where (3) is a hyperbolic tangent ranging from -1 to 1, and (4) is equivalent in shape, but ranges from 0 to 1. More specialized activation functions include radial basis functions, which are used in another class of supervised neural network models.

In order to perform the Elman RNN training, there are now several methods available. For instance, it is possible to apply gradient descent backpropagation and optimization methods, in a similar way to regular feedforward neural networks [18], as this method has been previously implemented by other authors. In spite of this, this method has not been chosen due to the well-known problem of the lack of guarantees of the algorithm finding a global minimum of the error function, since gradient descent may get stuck in any of the local minimum values [19,20]. In our case we have used the Levenberg–Marquardt algorithm as the training method. It is a modification of the Gauss–Newton method, designed in order to minimize the sum of squares of non-linear functions combining this technique with the steepest-descent algorithm [21]. Finally, we would like to remark that this method, the application of which is currently very common for RNNs [22–24], does not suffer the slow convergence problems that were reported for the methods from which it derived [25].

4. The variables and the database

The data used in the present research was collected using two double-frequency GPS receivers (Hiper-Plus, Topcon Positioning Systems, Inc., Livermore, CA, USA) observing the GPS pseudo-range and carrier phase. These data were used by the researchers in a previous study [1]. The GPS experimental data were collected for four days in 5–6 h periods between 20 and 23 August 2007. The antenna heights ranged from 1.45 to 1.60 m, and the logging rate was 1 s. The collection of observations lasted for at least 1.5 h and the process was repeated three times per day. GPS data were revised to ensure continuity and cut to obtain 12 data sets of 1 h (three data sets per day). The coordinates of the reference station were 42°41′08.79872″ N, 6°38′03.210587″ (latitude–longitude WGS84) and the ellipsoidal height was 933.829 m. These coordinates were calculated by differential correction, using the data from the base station named *ponf*, which is the nearest reference station of the Regional GNSS Network (http://gnss.itacyl.es/). This control point was projected by setting up the following UTM coordinates (m): 693814.623, 4728635.531 (easting, northing—datum ETRS89; zone 29N). This position was used to calculate the "true positions" of experimental points.

The input variables employed are those listed below:

- In order to characterize each tree mass, the trees in each parcel in a radius of 10 m around the observation point were measured and the following dasometric parameters were calculated: arithmetic mean diameter (d_m), mean height (h_m), slenderness coefficient (SLC), crown height (h_c), dominant height (H_0), treetop height (Tt_h), tree density (N), basal area (G), quadratic mean diameter (d_g), Hart-Becking index (HBI), wood volume (V) and biomass (W) (see [1] for a detailed explanation of each of these parameters).
- In addition to parameters reflecting plant cover, a number of variables were recorded in each second measurement, so the accuracy of observation was conditioned independently of the plant cover (GPS signal variables). These variables are listed as follows:
  - $PDOP_p$: position dilution of precision ($PDOP$) for each point under the forest canopy.
  - $PDOP_r$: $PDOP$ for the reference point.
  - σ_{Xr_acc}: error X for the reference point.
  - σ_{Yr_acc}: error Y for the reference point.
  - σ_{XYr_acc}: error XY for the reference point.
  - σ_{Zr_acc}: error Z for the reference point.

- E_p: mean elevation angle for the satellites transmitting the signal for points under the forest canopy.
- E_r: mean elevation angle for the satellites transmitting the signal received by the reference point.
- $DLLSNCA_p$: indicator of the signal–noise ratio in *CA* code (in dB*Hz) for a point under the forest canopy.
- $DLLSNCA_r$: indicator of the signal–noise ratio in *CA* code (in dB*Hz) for the reference point.
- nCA_p: number of satellites receiving *CA* code for a point under the forest canopy.
- nCA_r: number of satellites receiving *CA* code for the reference point.
- $DLLSNL1_p$: indicator of the signal–noise ratio in *P* code for *L*1 (in dB*Hz) for a point under the forest canopy.
- $DLLSNL1_r$: indicator of the signal–noise ratio in *P* code for *L*1 (in dB*Hz) for the reference point.
- $nL1_p$: number of satellites receiving code in the *L*1 carrier for a point under the forest canopy.
- $nL1_r$: number of satellites receiving code in the *L*1 carrier for the reference point.
- $DLLSNL2_p$: indicator of the signal–noise ratio in *P* code for *L*2 (in dB*Hz) for a point under the forest canopy.
- $DLLSNL2_r$: indicator of the signal–noise ratio in *P* code for *L*2 (in dB*Hz) for the reference point.
- $nL2_p$: number of satellites receiving code in the *L*2 carrier for a point under the forest canopy.
- $nL2_r$: number of satellites receiving code in the *L*2 carrier for the reference point.

5. The model

In the present research, two different kinds of models were trained and validated:

- The first kind of model that we checked has 20 neurons in the input layer. This means that according to Fig. 1, $m = 20$. This value corresponds to our 20 input variables. The output layer is composed of two neurons, $n = 2$, corresponding to the variables H_{acc} and V_{acc}. In order to evaluate the most appropriate number of neurons in the intermediate layer, a pruning methodology was used. Pruning involves evaluating the weighted connections between the layers. If the network contains any hidden neurons which contain only zero-weighted connections, they can be removed [26]. In our case, the optimum was fixed at 23 neurons in the hidden layer and the same number in the context layer. The optimum values of the learning rate, the parameter that controls the step size when weights are iteratively adjusted, the momentum (this allows a change to the weights to persist for a number of adjustment cycles) and the kind of activation function to be employed were found by means of the design of experiments (DOE) using the methodology proposed by Sánchez Lasheras et al. [27]. The results were as follows: 0.4872 for the momentum, a learning rate of 10^{-4}; the activation function finally chosen was the sigmoid (please note that both tansig and sigmoid were tested).
- The second kind of model used only seven variables. We would like to remark that due to the results obtained by the cited previous work, we were interested in comparing the performance that a recurrent neural network model using only this reduced set of variables would have, in comparison with the model that used the 20 items listed in Section 4 as input variables. In our previous article the influence of dasometric and GPS parameters in the accuracy of GPS measurements under the forest GPS canopy proved that for both H_{acc} and V_{acc} variables, the most important variables were, in this order: *SLC*, *HBI*, V, h_m, H_0, N and G. However, other variables were found to be important for both H_{acc} and V_{acc} (these other variables were: d_g, W, $PDOP_p$, $DLLSNL1_p$, h_c, d_m, $nL2_r$, $PDOP_r$ and $DLLSNCA_p$). Therefore, the second model used in this study employed seven input neurons, 13 neurons in the hidden layer and another 13 in the context layer, the number of neurons in the output layer being equal to 2 with a momentum of 0.5176, a learning rate of 10^{-4} and using the sigmoid as the activation function. All of these parameters were defined using the same methodology as for the first model.

6. Results

The total database used in the present research was made up of 43 000 cases. It was decided to use 75% of the information for the training and another 25% for the validation of the results obtained. This operation was repeated five times and the same subsets were used for the training and validation of the two kinds of models. Therefore, five training models with their corresponding validations were created for the model with 20 neurons, and the same numbers of validation and training models were created for the seven-variable model (the reason for creating a seven-variable model will be explained below). The validation sets were chosen randomly, but taking into account that they should be representative of the whole parameter space. In order to guarantee this condition, their Tukey numbers were studied and compared with those corresponding to the training data sets. No statistically significant differences were found using one-way ANOVA test at a significance level of $p = 0.05$. Therefore none of the random selections were repeated. Although the selection of data for the training and validation sets were performed randomly, we would like to remark that in order to take advantage of the particular properties of the RNN, the way in which the random selection was performed was as follows. As data were recorded in a temporal framework of four days, a date and hour for each one of the five sets was selected randomly. That moment was taken as the initial point for validation data and information up to a 20% of the total from such a moment was used for validation. If we reached the final recording moment before achieving 20% of the data, the rest of the information up to 20% was taken from the initial recording moment. Table 1 represents the network architecture of the five models that were trained, indicating the number of neurons, the CPU time that was necessary for its training and the number of epochs that were employed for both the models, with twenty and seven input variables.

Table 1
The twenty- and seven-variable models: network architecture, CPU times, training epochs.

Implemented neural network	Architecture[a]	CPU time (min.:sec.)	Training epochs
20 variables—Model 1	$20 + (23 + 23) + 2$	87:24	15 000
20 variables—Model 2	$20 + (23 + 23) + 2$	88:01	15 500
20 variables—Model 3	$20 + (23 + 23) + 2$	87:45	15 300
20 variables—Model 4	$20 + (23 + 23) + 2$	87:21	15 000
20 variables—Model 5	$20 + (23 + 23) + 2$	87:23	15 000
7 variables—Model 1	$7 + (13 + 13) + 2$	67:24	12 400
7 variables—Model 2	$7 + (13 + 13) + 2$	68:01	13 500
7 variables—Model 3	$7 + (13 + 13) + 2$	67:45	13 500
7 variables—Model 4	$7 + (13 + 13) + 2$	67:21	13 500
7 variables—Model 5	$7 + (13 + 13) + 2$	67:23	13 500

[a] Design of the recurrent neural network: number of neurons in the input layer plus number of neurons in the hidden layer and recurrent layer and number of neurons in the output layer.

Table 2
Performance evaluation parameters for the RNNs during the validation process.

Implemented neural network	MSE H_{acc}	MAE H_{acc}	RSQ H_{acc}	MSE V_{acc}	MAE V_{acc}	RSQ V_{acc}
20 variables—Model 1	0.0376	0.1817	0.8498	0.0374	0.1865	0.7143
20 variables—Model 2	0.0352	0.1827	0.8110	0.0370	0.1873	0.7247
20 variables—Model 3	0.0354	0.1823	0.8103	0.0374	0.1882	0.7243
20 variables—Model 4	0.0355	0.1885	0.8387	0.0377	0.1895	0.7296
20 variables—Model 5	0.0357	0.1905	0.8309	0.0378	0.1906	0.7207
7 variables—Model 1	0.0358	0.2006	0.8098	0.0380	0.1927	0.7095
7 variables—Model 2	0.0358	0.2126	0.8101	0.0382	0.1930	0.7019
7 variables—Model 3	0.0361	0.2145	0.8036	0.0385	0.1944	0.7093
7 variables—Model 4	0.0361	0.2188	0.8022	0.0392	0.1962	0.7021
7 variables—Model 5	0.0363	0.2237	0.8056	0.0397	0.1972	0.7065

In the present study, in order to measure the error of the RNNs during the validation process the following three performance measurements were employed: mean square error (MSE), mean absolute error (MAE), and determination coefficient (RSQ). Both MSE and MAE are suitable indicators for showing how close the predicted values of the models are to the real values. As is well-known, the correlation coefficient values are always in the range between -1 and 1. When the value of the said coefficient is equal to 1, it means that there is a perfect positive linear correlation between the result for the network output and the real value; that is, they vary by the same amount. In the same way, when r is equal to -1 it must be interpreted as indicating that there is a linear negative correlation between the network output and real value which means that they vary in opposite ways. Finally, correlation coefficients close to zero mean a complete lack of correlation between the outputs of the network (predicted values) and the real values. All this information is condensed in Table 2. As can be observed, there is complete coherence between the results from the RNNs and the real values, demonstrating the validity of the results from the RNNs trained in this study. But even more interesting than this coherence is the fact that the performance of the models formed from seven input variables is almost as good as the performance of those models formed from twenty variables. Finally, and in order to confirm from a visual point of view the results listed in Table 2, Fig. 2 represents, for one of the twenty-variable models and a seven-variable model, their results versus the real values for the variable H_{acc}. A detailed area of this figure is presented in Fig. 3.

As has been mentioned earlier, the present research compared the results obtained by means of two different models: one formed with twenty input variables and another with only seven. Although the selection of the seven input variables was already studied by the authors in previous research, in which genetic algorithms and multivariate adaptive regression splines (MARS) were employed [1], in this work the selection task was repeated using a new algorithm. In this case the referred to selection of the best variables for the model was performed taking into account the importance of its contribution to the RSQ values of the models formed when they were included in the models. In order to perform this analysis, the same five data samples were used for the training and validation of different models, using for this purpose all the different variables subsets. The results for the best subsets are shown in Fig. 4, where the RSQ means and 95% confidence intervals for the horizontal accuracy (H_{acc}) and vertical accuracy (V_{acc}) are represented. The RSQ increases each time a new variable is added to the model until the model with variables *SLC*, *HBI*, V, h_m, H_0, N and G is constructed. Therefore, the addition of signal variables to the model with these seven dasometric variables does not produce a significant increase of the RSQ.

In order to analyse the relative importance of the variables in the best model with seven variables, seven models with six of these variables were constructed and their respective RSQ values calculated. Results are shown in Table 3, where the importance of the variables was calculated as the relative increase in RSQ when the variable is added to a model with the rest of the most important dasometric variables. According to the table, *HBI*, G and *SLC* are the most important variables affecting both horizontal and vertical accuracy.

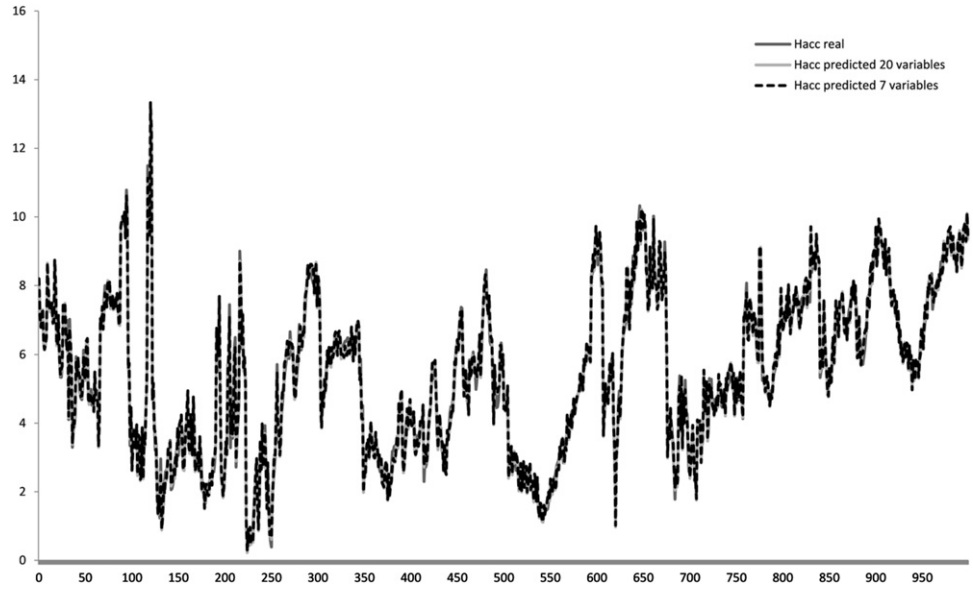

Fig. 2. Horizontal accuracy of a sample data set: real values, predicted values by means of the twenty-variable model and values predicted by means of the seven-variable model.

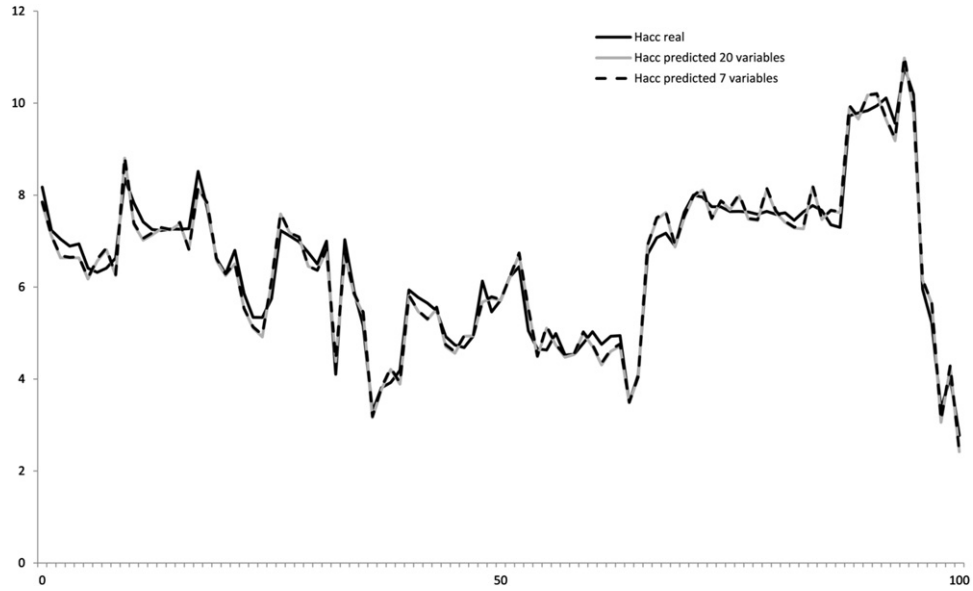

Fig. 3. Horizontal accuracy of a sample data set (detail of Fig. 2): real values, values predicted by means of the twenty-variable model and values predicted by means of the seven-variable model.

Table 3
Relative importance of each of the seven most important variables.

Variable	H_{acc} Importance (%)	V_{acc} Importance (%)
SLC	13.22	19.30
HBI	24.15	21.27
V	10.12	7.14
h_m	9.67	8.04
H_0	7.06	10.78
N	10.37	12.39
G	25.41	21.08

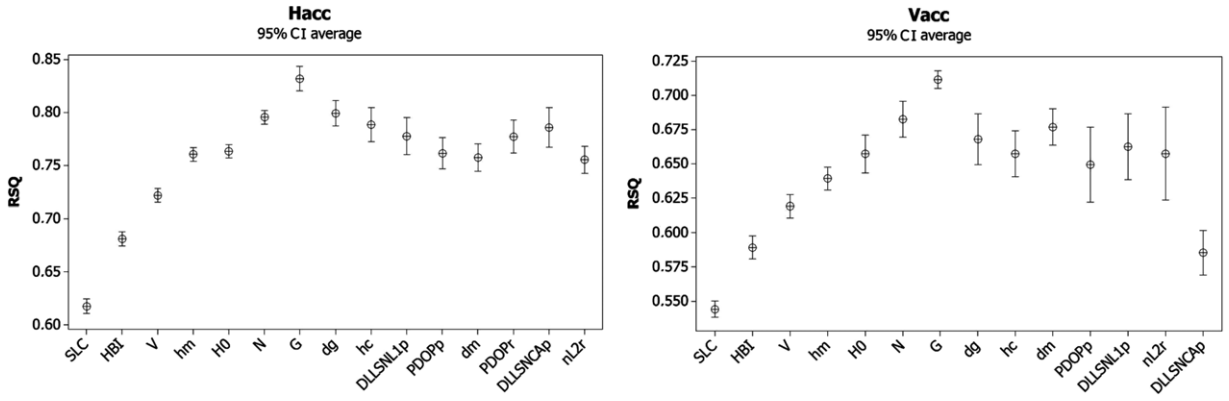

Fig. 4. RSQ means and 95% confidence intervals for the horizontal (left) and vertical (right) accuracy corresponding to the best subsets of variables.

7. Discussion and conclusions

According to the results presented in this research, the recurrent neural network is able to reproduce, with appreciable accuracy, the mechanism of the prediction of the GPS measurements and the associated errors due to the forest canopy.

In a similar way to how it was found by the authors in previous research [1], this article shows the influence of dasometric and GPS parameters on the accuracy of GPS measurements under the forest. This influence has been studied in depth for both H_{acc} and V_{acc} variables. The results of this research are in line with previous studies [1,9,10] and proved that for both variables the most important explaining variables are: *SLC*, *HBI*, V, d_m, H_0, N and G. A study of the relative importance of these variables showed that *HBI*, G and *SLC* are those with the greatest influence on the horizontal and vertical accuracy.

We would like to remark that the important influence of forest cover variables has already been highlighted by previous research performed with the same database [1,28], but with the use of recurrent neural networks we have improved the maximum RSQ values achieved for both H_{acc} and V_{acc} variables, which were 0.7869 and 0.6584 respectively in the cited paper.

Acknowledgements

We would like to thank Anthony Ashworth for his revision of the English grammar and spelling of the manuscript.

References

[1] C. Ordóñez Galán, J.R. Rodríguez Pérez, J. Martínez Torres, P.J. García Nieto, Analysis of the influence of forest environments on the accuracy of GPS measurements by using genetic algorithms, Math. Comput. Modelling 54 (2011) 1829–1834.
[2] J. Bao-Yen Tsui, Fundamentals of Global Positioning System Receivers: A Software Approach, Wiley-Interscience, New York, 2004.
[3] E.D. Kaplan, C. Hegarty, Understanding GPS: Principles and Applications, second ed., Artech House Publishers, New York, 2005.
[4] A. El-Rabbany, Introduction to GPS: The Global Positioning System, second ed., Artech House Publishers, New York, 2006.
[5] P. Sigrist, P. Coppin, M. Hermy, Impact of forest canopy on quality and accuracy of GPS measurements, Int. J. Remote Sens. 20 (1999) 3595–3610.
[6] M.G. Wing, J. Frank, Vertical measurement accuracy and reliability of mapping-grade GPS receivers, Comput. Electron. Agric. 78 (2011) 188–194.
[7] B. Husch, T.W. Beers, J.A. Kershaw, Forest Mensuration, fourth ed., John Wiley and Sons, Hoboken, 2003.
[8] A. Walpersdorf, M.N. Bouin, O. Bock, E. Doerflinger, Assessment of GPS data for meteorological applications over Africa: study of error sources and analysis of positioning accuracy, J. Atmos. Sol.–Terr. Phys. 69 (2007) 1312–1330.
[9] R. Cordero, O. Mardones, M. Marticorena, Evaluation of forestry machinery performance in harvesting operations using GPS technology, in: P.A. Ackerman, D.W. Längin, M.C. Antonides (Eds.), Precision Forestry in Plantations, Semi-natural and Natural Forests, Proceedings of the International Precision Forestry Symposium, Stellenbosch University, South Africa, 2006, pp. 163–173.
[10] J.C. Gegout, C. Piedallu, Effects of forest environment and survey protocol on GPS accuracy, Photogramm. Eng. Remote Sens. 71 (2005) 1071–1078.
[11] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (1990) 179–211.
[12] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, 1986, p. 45.
[13] W. Hinoshita, H. Arie, J. Tani, H.G. Okuno, Tetsuya Emergence of hierarchical structure mirroring linguistic composition in a recurrent neural network, Neural Netw. 24 (2011) 311–320.
[14] H. Zhao, X. Zeng, J. Zhang, Y. Liu, X. Wang, T. Li, A novel joint-processing adaptive nonlinear equalizer using a modular recurrent neural network for chaotic communication systems, Neural Netw. 24 (2011) 12–18.
[15] S.M. Kelo, S.V. Dudul, Short-term Maharashtra state electrical power load prediction with special emphasis on seasonal changes using a novel focused time lagged recurrent neural network based on time delay neural network model, Expert Syst. Appl. 38 (2011) 1554–1564.
[16] T. Hsieh, H. Hsiao, W. Yeh, Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm, Appl. Soft Comput. 11 (2011) 2510–2525.
[17] T.T. Rogers, J.L. McClelland, Semantic Cognition: A Parallel Distributed Processing Approach, first ed., MIT Press, Cambridge, 2004.
[18] F.J. Pineda, Generalization of back-propagation to recurrent neural networks, Phys. Rev. Lett. 59 (1987) 2229–2232.
[19] M. Gori, A. Tesi, On the problem of local minima in backpropagation, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 76–86.
[20] X.G. Wang, Z. Tang, H. Tamura, M. Ishii, W.D. Sun, An improved backpropagation algorithm to avoid the local minima problem, Neurocomputing 56 (2004) 455–460.
[21] M.T. Hagan, M.B. Menhaj, Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Netw. 6 (1994) 989–993.

[22] H. Mirzaee, Long-term prediction of chaotic time series with multi-step prediction horizons by a neural network with Levenberg–Marquardt learning algorithm, Chaos Solitons Fractals 41 (2009) 1975–1979.
[23] E.D. Übeyli, Recurrent neural networks with composite features for detection of electrocardiographic changes in partial epileptic patients, Comput. Biol. Med. 38 (2008) 401–410.
[24] A.B. Chelani, C.V.C. Rao, K.M. Phadke, M.Z. Hasan, Prediction of sulphur dioxide concentration using artificial neural networks, Environ. Model. Software 17 (2002) 159–166.
[25] R. Battiti, First- and second-order methods for learning: between steepest descent and Newton's method, Neural Comput. 4 (1992) 141–166.
[26] N. Fnaiech, F. Fnaiech, B.W. Jervis, Feedforward neural networks pruning algorithms, in: J.D. Irwin (Ed.), Intelligent Systems, CRC Press, Boca Raton, 2011, pp. 15-1–15-15. (Chapter 15).
[27] F. Sánchez Lasheras, J.A. Vilán Vilán, P.J. García Nieto, J.J. del Coz Díaz, The use of design of experiments to improve a neural network model in order to predict the thickness of the chromium layer in a hard chromium plating process, Math. Comput. Modelling 52 (2010) 1169–1176.
[28] C. Ordóñez, J.R. Rodríguez-Pérez, J.J. Moreira, J.M. Matías, E. San-Ablanedo, Machine learning techniques applied to the assessment of GPS accuracy under the forest canopy, J. Surveying Eng. 137 (2011) 140–149. http://dx.doi.org/10.1061/(ASCE)SU.1943-5428.0000049.