Neural Comput & Applic (2011) 20:417–439
DOI 10.1007/s00521-010-0407-3

ORIGINAL ARTICLE

Comparison of new activation functions in neural network for forecasting financial time series

Gecynalda S. da S. Gomes · Teresa B. Ludermir · Leyla M. M. R. Lima

Received: 26 November 2009 / Accepted: 4 June 2010 / Published online: 20 June 2010

Abstract In artificial neural networks (ANNs), the activation function most used in practice are the logistic sigmoid function and the hyperbolic tangent function. The activation functions used in ANNs have been said to play an important role in the convergence of the learning algorithms. In this paper, we evaluate the use of different activation functions and suggest the use of three new simple functions, complementary log-log, probit and log-log, as activation functions in order to improve the performance of neural networks. Financial time series were used to evaluate the performance of ANNs models using these new activation functions and to compare their performance with some activation functions existing in the literature. This evaluation is performed through two learning algorithms: conjugate gradient backpropagation with Fletcher–Reeves updates and Levenberg–Marquardt.

Keywords Neural networks · Activation functions · Complementary log-log · Probit · Log-log · CGF algorithm · LM algorithm

G. S. da S. Gomes · T. B. Ludermir (✉) · L. M. M. R. Lima
Centre of Informatics, Federal University of Pernambuco,
Av. Prof. Luis Freire, s/n, Cidade Universitria Recife/PE,
Recife 50732-970, Brazil
e-mail: tbl@cin.ufpe.br

G. S. da S. Gomes
e-mail: gssg@cin.ufpe.br

L. M. M. R. Lima
e-mail: lmmrl@cin.ufpe.br

1 Introduction

Several modeling techniques have been developed in order to provide a more adequate understanding of complex and nonlinear systems, for example, the evolution of stock prices in financial markets. The behavior of such systems is influenced by factors that can affect the development of the system over time. For example, the stock price movement may adjust very quickly to new information when they become public, making it difficult to forecast this stock market movement. Hence, the use of techniques such as artificial neural networks (ANNs) can be a useful alternative in the modeling of these systems.

Approaching the problems of forecasting time series, for several decades, many authors have used different statistical methods such as autoregressive (AR) models, moving averages (MA) models, linear or nonlinear regression exponential smoothing models for modeling and forecasting. Box and Jenkins [1] developed the autoregressive integrated moving average (ARIMA) model to predict time series. The ARIMA model is used, basically, for non-stationary series, when linearity between variables is valid. However, there are many series in which supposing linearity is not valid. Clearly, the ARIMA models do not produce effective results when used for capturing and explaining nonlinear relations, causing forecasts errors to increase. To improve forecasting time series with nonlinear characteristics, several researchers developed alternative methods that model such approximations, for example, autoregressive heteroscedastic models (ARCH) [2]. Although these methods have shown some improvements over linear methods, they tend to be specific for certain applications.

ANNs nonlinear models are part of an important class which has attracted considerable attention in many applications. The use of ANNs in many applied works is in general motivated by empirical results, showing that, under conditions of regularity, simpler models of ANNs are able to approximate any measurable function to any decision degree; some examples may be seen in Cybenko [3], White [4] or Gallant and White [5]. The central topic in ANNs related literature, considered in many publications, revolves around the specification of such model, i.e., to find good network architecture, the best combination of parameters and input variables.

ANNs have been widely used in studies of complex time series forecasting, such as weather, energy consumption, financial series, among others. There is a long history of research on modeling financial series, and the traditional analysis of time series is among the most used. Time series in finance and economics have the following features:

1. Data intensity
2. Unstructured nature
3. High degree of uncertainty
4. Hidden relationships

Some studies show that financial markets have nonlinear characteristics [6, 7]. Regarding the stock market, statistical methods were an alternative for beginning to understand and predict developments in the market. However, patterns of ANNs have become increasingly attractive in the study of financial time series (as well as other nonlinear systems) due to the fact that these models can keep up with continuous and sudden changes in these systems, which cannot be described by well-developed statistical methods [6]. Therefore, the neural networks models have been used to predict stock market prices because they are able to learn nonlinear mappings between inputs and outputs.

In general, the performance of neural networks depends on various factors such as the number of hidden layers, the number of hidden nodes, the learning algorithm and the activation function of each node. However, the main emphasis in neural network research is on learning algorithms and architectures, neglecting the importance of activation functions. On the other hand, the choice of activation functions may strongly influence complexity and performance of neural networks and have been said to play an important role in the convergence of the learning algorithms [8, 9, 10, 11].

Some types of activation functions have been proposed. Pao [12] used a combination of various functions, such as polynomial, periodic, sigmoidal and Gaussian functions. Hartman et al. [13] proposed Gaussians bars activation functions. Hornik [14, 15] and Leshno et al. [16] used non-polynomial activation functions. Leung and Haykin [17] used rational transfer functions with very good results. Giraud et al. [18] used Lorentzian transfer functions. Singh and Chandra [11] proposed a new class of sigmoidal functions and proved that they satisfy the requirements of the universal approximation theorem. Skoundrianos and Tzafestas [19] proposed a new sigmoidal activation function with good results for modeling of dynamic, discrete time systems. Ma and Khorasani [20] used Hermite Polynomial with very satisfactory results. Wen and Ma [21] proposed a Max-Piecewise-Linear (MPWL) Neural Network for function approximation. Efe [22] introduced two new activation functions labeled `sincos` and `sinc`. Gomes and Ludermir [23] used complementary log-log and probit functions to show that when the data follow a binomial distribution with characteristics of complementary log-log and probit functions using the logistic sigmoid function in the neural networks models is inadequate.

None of the studies mentioned earlier make any kind of comparison based on financial market time series. Some of these studies only show the theoretical aspect of the function and others make comparisons with sigmoid logistic and hyperbolic tangent functions only. This paper considers twelve activation functions and networks that contain seven different amounts of hidden nodes. Besides that, it considers two learning algorithms and evaluates the performance of all the possible combination of these components in the prediction of twelve financial time series with very unusual behavior. Under these conditions, this study gives us an idea of the situations in which each activation function is more adequate. The main contributions of this paper are the following:

- the implementation of new activation functions: complementary log-log, probit and log-log,
- comparison of the performance of the models with the new functions and also with other functions available in the literature using financial market data sets,
- evaluation of the performance of the models with these functions through two learning algorithms,
- demonstration that the models with the new functions can have good performance with few hidden nodes, and
- demonstration that the models with the available functions, such as sigmoid and hyperbolic tangent can perform well too, but they require many hidden nodes to do so.

Financial time series were chosen to evaluate the performance of twelve activation functions, including the functions proposed, for not having well-defined behavior—it sometimes presents a positive trend and other times a negative trend—characteristic of financial market. In the experiments, we used two learning algorithms: conjugate gradient backpropagation with Fletcher–Reeves updates (CGF) and Levenberg–Marquardt (LM). The CGF is a network training function that updates weights and bias values according to the Fletcher–Reeves conjugate gradient algorithm. It has the smallest storage requirements among the conjugate gradient algorithms. See [24] or [25] for a discussion of the Fletcher–Reeves conjugate gradient algorithm. The LM algorithm was designed to approach second-order training speed without using a Hessian matrix [26]. The original description of the Levenberg–Marquardt algorithm is given in [27].

The paper is organized as follows: in Sect. 2, we present the new functions to be used as activation functions in neural networks, and we describe all activation functions considered in this work. In Sect. 3, we present the experimental results. We conclude in Sect. 4.

2 Activation functions

ANNs have been successfully applied in a variety of problems, such as classification, clustering, optimization, time series forecasting, etc. ANNs are capable of adequately modeling a variety of problems due to their ability to approximate a variety of nonlinear mappings, tackle massive parallel processing of information as well as their ability to learn from and adapt to their environment.

We use multilayer perceptron (MLP) networks with the typical network connectivity with p input nodes, q hidden nodes and a single output. The hidden-layer nodes (first-layer synaptic weights) are represented as shown in Fig. 1a, while the output-layer nodes (second-layer synaptic weights) are represented as shown in Fig. 1b. The calculations made for the hidden layer of are given by

$$y_i(t) = \phi_i(u_i(t)) = \phi_i\left(\sum_{j=0}^{p} w_{ij}(t)x_j(t)\right) = \phi_i(\mathbf{w}_i^\top(t)\mathbf{x}(t)),\quad i = 1,\ldots,q, \tag{1}$$

where $\mathbf{w}_i$ is the weight vector associated with the node i, $\mathbf{x}(t)$ is the attribute vector, ϕ_i is the activation function and t represents the iteration where $\mathbf{x}(t)$ is presented.

In the output layer, the calculation of nodes is expressed by

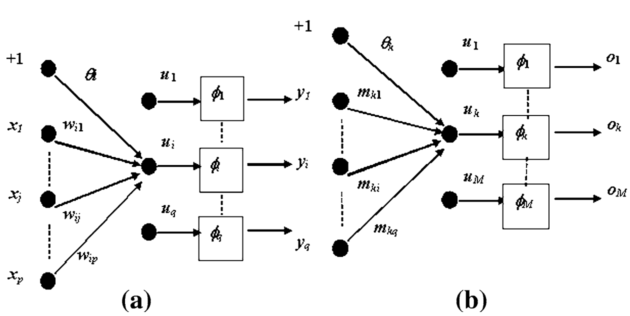

Fig. 1 **a** Nodes in the hidden layer and **b** Nodes in the output layer

$$o_k(t) = \phi_k(u_k(t)) = \phi_k\left(\sum_{i=0}^{q} m_{ki}(t)y_i(t)\right) = \phi(\mathbf{m}_k^\top(t)\mathbf{y}(t)),\quad k = 1,\ldots,M$$

where $\mathbf{m}_i$ is the weight vector associated with the node k and M indicates the number of output nodes. The activation function ϕ_k in the output layer o_k assumes the linear form. A single hidden layer is sufficient for a MLP to uniformly approximate any continuous function with support in a unit hypercube [28] and [3]; based on this fact, we use in this study only one hidden layer.

A neural network can be characterized by three main aspects: (1) the pattern of connections between units (architecture), (2) the method of determining the weights of the connections (training or learning algorithm) and (3) its activation function. The main emphasis in research on neural networks is the learning algorithms and architectures, neglecting the importance of the activation functions [9, 11], since the structure of neural networks with sigmoid or hyperbolic tangent type neuronal nonlinearity is probably the most frequently used connectionist architecture in many applications. However, in practice, there is a need to use simple models, for it is not always feasible to use complex models for optimization of architecture or of learning algorithms, despite its importance. Flexibility of contours of activation functions used for estimation of decision borders is strongly correlated with the number of functions (and thus with the number of adaptive parameters available for training) necessary to model complex shapes of decision borders [9]. The implementation of new activation functions in learning algorithms is a simple task, just replace one activation function (and its corresponding derivative) commonly used in the literature by any of the new activation functions proposed here. Because of this simplicity and the important role that the activation function represents, we consider it relevant to evaluate and compare the performance of MLP neural network with different activation functions.

Table 1 displays a list of activation functions that we used in this study. In the first column, we give the label associated with each activation function. In the second column, we have the activation function and in the last column we have the first derivative of each activation function. Briefly, on the first line we have the logistic sigmoid function called `logsig`. The second line describes the hyperbolic tangent labeled as `tanh`, and on the third line we have the first function proposed in this study, which is the function complementary log-log labeled `cloglog`. On the fourth line, we have a modified function of `cloglog`, which we call `cloglogm`. In the fifth line, we have the second proposed function, which is the probit function labeled as `probit`, and in the sixth line we have

Table 1 Activation functions and their derivatives

Label	Activation function	Corresponding derivative function
`logsig`	$\phi_i(u_i(t)) = \frac{1}{1+\exp(-u_i(t))}$	$\phi'_i(u_i(t)) = \phi_i(u_i(t)) \cdot (1 - \phi_i(u_i(t)))$
`tanh`	$\phi_i(u_i(t)) = \frac{2}{1+\exp(-2 \cdot u_i(t))} - 1$	$\phi'_i(u_i(t)) = 1 - (\phi_i(u_i(t))^2$
`cloglog`	$\phi_i(u_i(t)) = 1 - \exp(-\exp(u_i(t)))$	$\phi'_i(u_i(t)) = \exp(u_i(t)) \cdot (1 - \phi_i(u_i(t)))$
`cloglogm`	$\phi_i(u_i(t)) = 1 - 2 \cdot \exp(-0.7 \cdot \exp(u_i(t)))$	$\phi'_i(u_i(t)) = 1.4 \cdot \exp(u_i(t)) \cdot \exp(-0.7 \cdot \exp(u_i(t)))$
`probit`	$\phi_i(u_i(t)) = \Phi(u_i(t))$	$\phi'_i(u_i(t)) = \frac{1}{\sqrt{2\pi}} \exp(-u_i(t)^2/2)$
`loglog`	$\phi_i(u_i(t)) = \exp(-\exp(-u_i(t)))$	$\phi'_i(u_i(t)) = \exp(-u_i(t)) \cdot \phi_i(u_i(t))$
`sech`	$\phi_i(u_i(t)) = \frac{2}{\exp(u_i(t))+\exp(-u_i(t))}$	$\phi'_i(u_i(t)) = -\phi_i(u_i(t)) \cdot \texttt{tanh}$
`sinc` [22]	$\phi_i(u_i(t)) = \begin{cases} \frac{\sin(\pi u_i(t))}{\pi u_i(t)}, & u_i(t) \neq 0 \\ 1, & u_i(t) = 0 \end{cases}$	$\phi'_i(u_i(t)) = \begin{cases} \frac{\cos(\pi u_i(t)) - \mathrm{sinc}(u_i(t))}{u_i(t)}, & u_i(t) \neq 0 \\ 0, & u_i(t) = 0 \end{cases}$
`wave` [29]	$\phi_i(u_i(t)) = (1 - u_i^2(t)) \cdot \exp(-u_i^2(t))$	$\phi'_i(u_i(t)) = 2 \cdot u_i(t) \cdot \exp(-u_i^2(t)) \cdot (-2 + u_i^2(t))$
`sincos` [22]	$\phi_i(u_i(t)) = \sin(u_i(t)) + \cos(u_i(t))$	$\phi_i(u_i(t)) = \cos(u_i(t)) - \sin(u_i(t))$
`rootsig` [9]	$\phi_i(u_i(t)) = \frac{u_i(t)}{1+\sqrt{(1+u_i^2(t))}}$	$\phi'_i(u_i(t)) = \frac{1}{\left(1+\sqrt{1+u_i^2(t)}\right) \cdot \sqrt{1+u_i^2(t)})}$
`logsigm` [11]	$\phi_i(u_i(t)) = \left(\frac{1}{1+\exp(-u_i(t))}\right)^2$	$\phi'_i(u_i(t)) = \frac{2\exp(-u_i(t))}{(1+\exp(-u_i(t)))^3}$

the third proposed function that is the log-log function is labeled as `loglog`, these new functions proposals will be better defined in Section 2.1. In the seventh row in Table 1, we have the hyperbolic secant function labeled `sech`. On the eighth and tenth lines, we have functions labeled the `sinc` and `sincos` proposed by Efe [22], whereas on the ninth line we have the function inspired by wavelets proposed by Hara and Nakayamma [29] labeled `wave`. On the 11th line, we have a sigmoid function with roots shown by Duch and Jankowski [9], which we call `rootsig`, and on the last line we have the modified logistic sigmoid function proposed by Singh and Chandra [11] labeled `logsigm`.

2.1 New activation functions

The aim of our work is to implement sigmoid functions commonly used in binomial regression models in the processing units of neural networks and evaluate the performance of neural network models. The binomial regression models is a special case of generalized linear models (GLM) [30, 31]. The general structure of a GLM is formed by a random component, a systematic component and a monotonic differentiable function, known as link function $g(\cdot)$, which relates the random and systematic components. The link functions commonly used in a binomial model are given below

$$\eta = g(\tau) = \log(-\log(1-\tau)), \tag{2}$$

$$\eta = g(\tau) = \Phi^{-1}(\tau) \quad \text{and} \tag{3}$$

$$\eta = g(\tau) = -\log(-\log(\tau)). \tag{4}$$

representing the complementary log-log, probit and log-log link functions, respectively. $\Phi(\cdot)$ denotes the cumulative probability function for the normal distribution. Bliss [32] introduced modeling proportion data using a binomial model with the probit link function. On the other hand, the complementary log-log link function is recommended by Collett when the distribution of proportions is very asymmetric [33]. Logistic and probit link functions are appropriate if the distribution is symmetric [34]. In binomial regression, the response variable of interest is always a probability. Thus, the use of cumulative distribution functions for generating new links and, consequently, new models can be a viable alternative. The proposed functions are the inverse link functions given by

$$\tau = f(\eta) = 1 - \exp(-\exp(\eta)), \tag{5}$$

$$\tau = f(\eta) = \Phi(\eta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\eta} e^{-\eta^2/2} \mathrm{d}\eta \quad \text{and} \tag{6}$$

$$\tau = f(\eta) = \exp(-\exp(-\eta)). \tag{7}$$

representing, equivalently, the complementary log-log, probit and log-log activation functions, respectively. Figure 2 compares the shape of the inverse logistic, complementary log-log, probit and log-log link functions. The four functions are almost linearly related over the interval $0.1 \leq f(\eta) \leq 0.9$. The complementary log-log function is very similar to the logistic, for small values of $f(\eta)$ and the log-log function is very similar to the logistic, for large values of $f(\eta)$. As $f(\eta)$ approaches 1, the complementary log-log function approaches infinity much more slowly than either the logistic or the probit function [35], the opposite occur with the log-log function.

The universal approximation theorem (UAT) gives a mathematical justification for the approximation of an arbitrary continuous function opposed to its exact representation [36]. Our next step is to show how these new functions satisfy the TAU.

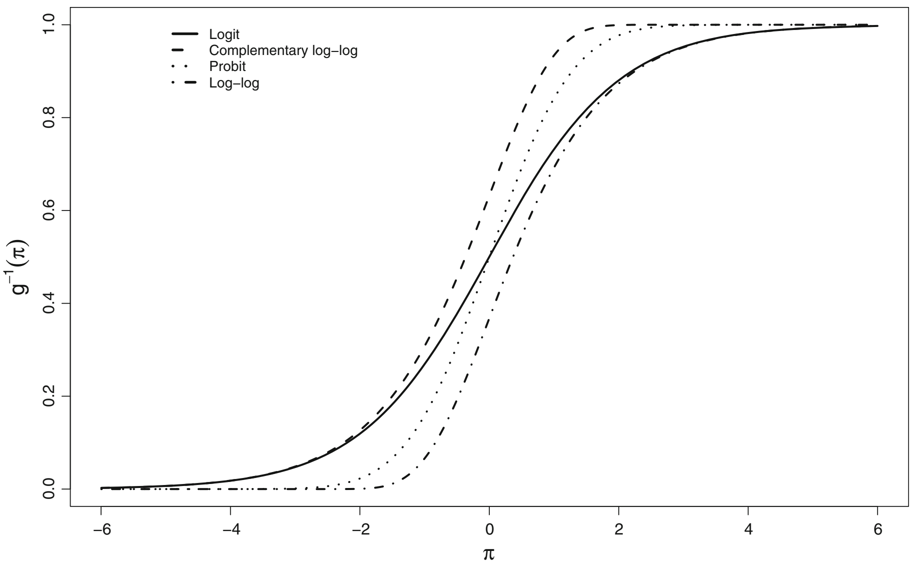

Fig. 2 A graphical comparison of four link functions

The following propositions require the new functions to be monotonically increasing, limited and non-constant.

Proposition 1 *The complementary log-log, probit and log-log functions are monotonically increasing (MI).*

Proof Hypothesis: $\eta_2 > \eta_1$.

COMPLEMENTARY LOG-LOG FUNCTION: By hypothesis, $\exp(\eta_2) > \exp(\eta_1)$, because $\forall\eta$, $\exp(\eta) > 0$, then

$$\begin{aligned} -\exp(\eta_2) &< -\exp(\eta_1) \\ \exp(-\exp(\eta_2)) &< \exp(-\exp(\eta_1)) \\ -\exp(-\exp(\eta_2)) &> -\exp(-\exp(\eta_1)) \\ 1-\exp(-\exp(\eta_2)) &> 1-\exp(-\exp(\eta_1)). \end{aligned}$$

Therefore, the complementary log-log function is MI.

PROBIT FUNCTION: The function $\Phi(\eta)$ is the cumulative distribution function of the standard normal, as previously said, therefore, the function $\Phi(\eta)$ constitutes a probability, then, $\Phi(\eta) > 0$. As $\eta_2 > \eta_1$, it follows that $\Phi(\eta_2) > \Phi(\eta_1)$. Therefore, the probit function is MI.

LOG-LOG FUNCTION: By hypothesis, $-\eta_2 < -\eta_1$, then

$$\begin{aligned} \exp(-\eta_2) &< \exp(-\eta_1) \\ -\exp(-\eta_2) &> -\exp(-\eta_1) \\ \exp(-\exp(-\eta_2)) &> \exp(-\exp(-\eta_1)). \end{aligned}$$

Therefore, the log-log function is MI. □

Proposition 2 *The complementary log-log, probit and log-log functions are limited by 1 when $\eta \to +\infty$ and by 0 when $\eta \to -\infty$.*

Proof COMPLEMENTARY LOG-LOG FUNCTION: For the upper limit, we have, by the properties of limit,

$$\lim_{\eta\to+\infty} f(\eta) = \lim_{\eta\to+\infty} 1-\exp(-\exp(\eta)) = 1-0 = 1;$$

for the bottom limit, we have

$$\lim_{\eta\to-\infty} f(\eta) = \lim_{\eta\to-\infty} 1-\exp(-\exp(\eta)) = 1-1 = 0.$$

PROBIT FUNCTION: For the probit function, the proof is direct, since $\Phi(\eta)$ is a probability and therefore limited to the interval [0, 1].

LOG-LOG FUNCTION: For the upper limit we have,

$$\lim_{\eta\to+\infty} f(\eta) = \lim_{\eta\to+\infty} \exp(-\exp(-\eta)) = 1;$$

for the bottom limit, we have

$$\lim_{\eta\to-\infty} f(\eta) = \lim_{\eta\to-\infty} \exp(-\exp(-\eta)) = 0.$$

□

Proposition 3 *Complementary log-log, probit and log-log are continuous differentiable functions, i.e., these functions are non-constant, because the derivatives, given in Eqs. 8–10, respectively, are non-zero.*

Proof Taking the derivative of functions Eqs. 5–7, we have the forms of the complementary log-log, probit and log-log derivatives functions, respectively,

$$f'(\eta) = \exp(\eta)\cdot\exp\{-\exp(\eta)\} \tag{8}$$

$$f'(\eta) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\eta^2}{2}\right) \tag{9}$$

$$f'(\eta) = \exp(-\eta)\cdot\exp(-\exp(-\eta)). \tag{10}$$

□

From the propositions 1–3, we can see that the complementary log-log, probit and log-log functions are non-constant, limited and monotonically increasing. The modified complementary log-log function, `cloglogm`, is also a non-constant function, monotonically increasing and limited by 1 when $\eta \rightarrow +\infty$ and by -1 when $\eta \rightarrow -\infty$. So, these functions satisfy the properties required by the TAU, and therefore they can be used as activation functions of a neural network.

It is known that when activation functions are changed, their derivatives also change, so they must be replaced in the equations for the calculation of the local gradients in the learning rules of the weights. Due to the fact that the learning rule has a direct impact on the convergence of learning algorithms, appropriate changes were made to use new activation functions.

3 Experimental results

In this section, the experimental results obtained with the implementation of the new activation functions, complementary log-log, probit and log-log, and also the other functions exposed in Table 1, will be presented. The experiments were conducted with twelve different time series data bases referring to financial market, whose description is presented in Table 2. These time series were obtained from http://www.finance.yahoo.com/. All the experiments were executed using the Matlab platform. We used six small bases of monthly series, which were "Vale do Rio Doce", "Apple", "Yahoo", "Microsoft", "Motorola", "GM" (these are represented in Fig. 3), and six larger bases of daily series, which were "Wal-Mart", "HSBC", "TAM", "Itau", "Brasil Telecom" and "Petrobras", represented in Fig. 4. Through Figs. 3 and 4, it is observable that each series has a unique behavior.

Table 3 shows descriptive statistics regarding the bases used in this study. Through the coefficient of asymmetry it is possible to see which series behave asymmetrically just by following one rule: symmetric data → skewness = 0; positive asymmetric data → skewness > 0; negative asymmetric data → skewness < 0. We are going to analyze the results of this study based on observations of the Collet [33], according to which, the use of complementary log-log function is most appropriate when the data are very asymmetric.

For all the series, we executed AR models to select the number of lags. The selected amount was used as input in the neural networks model (see Table 4). For example, for the "Vale do Rio Doce" series, the amount of lags selected was equal to 1, as in its neural network model, so for the neural network model we use 1 input node. Therefore, the amount of lags showed in Table 4 represents p input nodes.

We built models based in MLP networks built with one hidden layer, q hidden nodes, with q varying as follows, $q = \{2, 4, 6, 8, 12, 16, 20\}$, and one node in the output layer, with linear function. For forecasting (using the test set), we used 12 steps ahead for the monthly series and 30 steps ahead for daily series. On the validation set, we used the same amount of values as the test set and the remaining values were used on the training set. To evaluate the results, we used two learning algorithms, the conjugate gradient backpropagation with Fletcher–Reeves updates (CGF) and the LM. The neural network models were evaluated using twelve different activation functions in the hidden layer shown in Table 1. A hundred different initializations were executed for the weights and bias from a uniform distribution $U(-1,1)$. Computing the first 100 experiments for each amount of hidden nodes, we can see that 700 experiments were executed for each model, in each algorithm used in the training process and in each data base, leading to a total of 201,600 experiments. Every individual training trial is terminated after the completion of 5,000 epochs or if the epoch error increases by five consecutive epochs.

The criteria used to choose the best model is the criteria of mean absolute percentage error of forecast (MAPE). The MAPE is a measure of accuracy in a fitted time series value in statistics, which usually expresses accuracy as a percentage, and is found by the formula

$$\mathrm{MAPE}(\%) = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100 \quad (11)$$

where A_t is the actual value and F_t is the forecast value.

Table 4 presents the results of MAPE for forecasting, obtained by AR model to serve as a reference when we present the results forecasting of the neural network models. In Tables 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, we present the results of the average performance predicted by the experiments with ANNs models, but due to the large volume of data, we will not present the results of the average performance for the training set. In these tables, we

Table 2 Characteristics of each time series

Time series	From	Until	Type	Size
Vale do Rio Doce	09-Oct-03	24-Apr-09	Monthly	67
Apple	01-Jan-03	24-Apr-09	Monthly	76
Yahoo	12-Apr-96	02-Mar-09	Monthly	156
Microsoft	13-Mar-86	01-Apr-09	Monthly	278
Motorola	03-Jan-77	23-Apr-09	Monthly	388
GM	02-Jan-62	01-Apr-09	Monthly	562
Wal-Mart	21-Feb-06	24-Apr-09	Daily	800
HSBC	03-Mar-03	24-Apr-09	Daily	1,549
TAM	01-Jan-03	24-Apr-09	Daily	1,634
Itau	28-Feb-02	06-Mar-09	Daily	1,826
Brasil Telecom	19-Nov-01	24-Apr-09	Daily	1,868
Petrobras	03-Jan-00	30-Mar-09	Diary	2,400

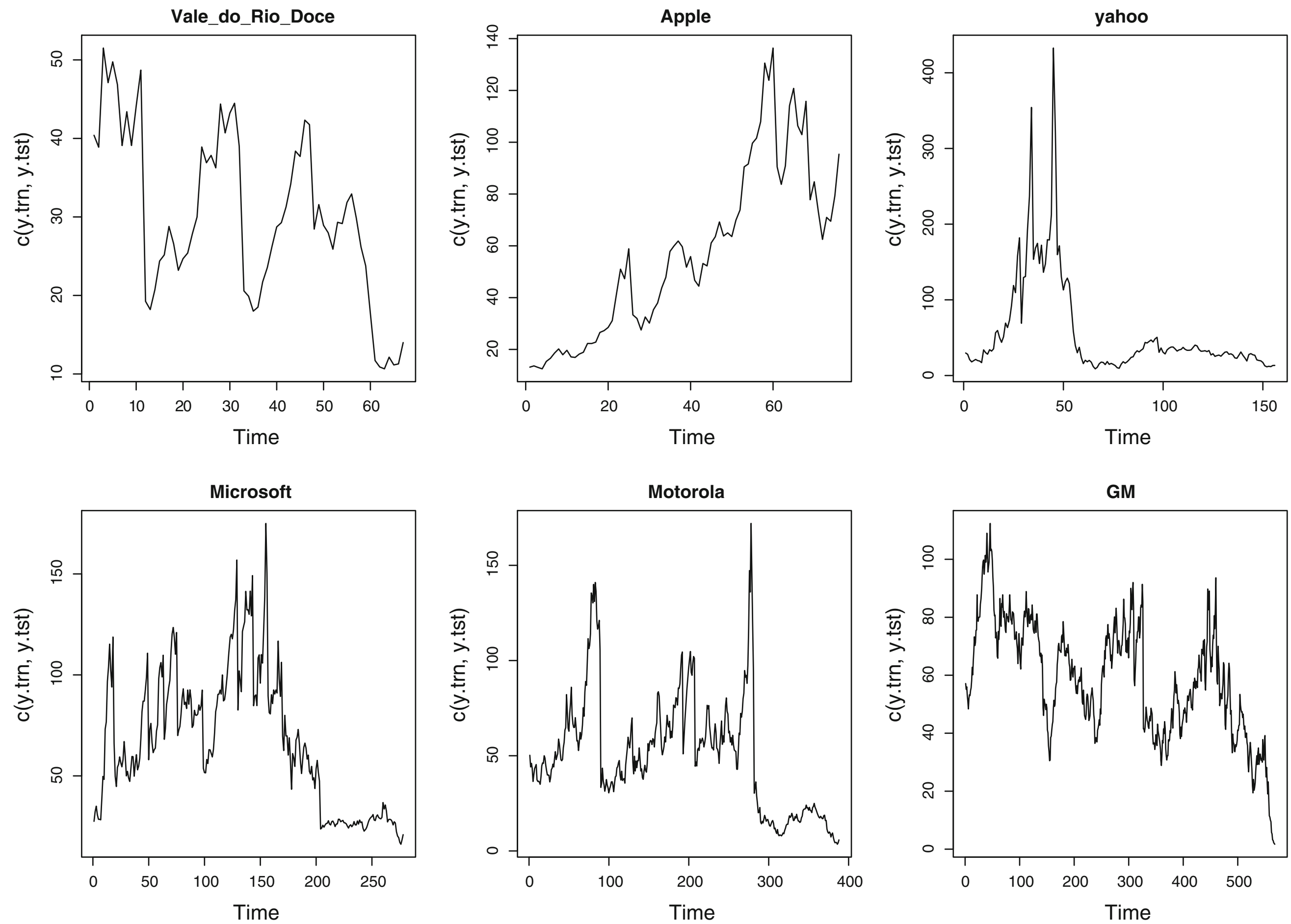

Fig. 3 Monthly time series

present the average MAPE from the 100 initialization for each model, which averages,

$$\overline{\text{MAPE}}(\%) = \frac{1}{100}\sum_{i=1}^{100} \text{MAPE}_i, \tag{12}$$

having its confidence interval in a level of 95% (95% CI) and the average number of epochs for each amount of hidden nodes in the different learning algorithms. The average number of epochs for the models with different activation functions was equivalent for all of them. The 95% CI is presented with the goal of evaluating the statistical hypothesis that the average performance of the selected models on these real stock market time series are the same (or not). The best results are emphasized in bold. In some situations, the average result is equal, but was emphasized the model with less variability, which averages the one with the smaller CI.

In Table 5, we present the results for the "Vale do Rio Doce" stock market series. Through these results, we can see that all the results of ANNs were better than the results for the AR model (Table 4). For this series, the degree of asymmetry is almost undetectable, because the value of the coefficient is almost zero (skewness = −0.02, see Table 3), so we can consider the series to be symmetric. The model with the `probit` activation function got the best average result with the use of the CGF algorithm and with 2 hidden nodes. The 95% CI for the $\overline{\text{MAPE}}$ corroborates this fact, because its superior limit was smaller than the inferior limit of any other interval, except for the `tanh` model with presented equivalent average result with the use of 12 hidden nodes. With the use of the LM algorithm, the average result of the models with the `tanh`, `cloglogm` and `sech` were equivalent for the networks with two hidden nodes. However, it is worth pointing out that the result for the model with the `probit` function was equivalent to the previous result, which averages that the reduction on the average performance using the LM algorithm compared to the use of the CGF was 3.6%. This shows that the use of the LM algorithm did not influence the result so much when using the `probit` activation function. The model with the `logsig` function reaches equivalent results with six hidden nodes, using the LM algorithm.

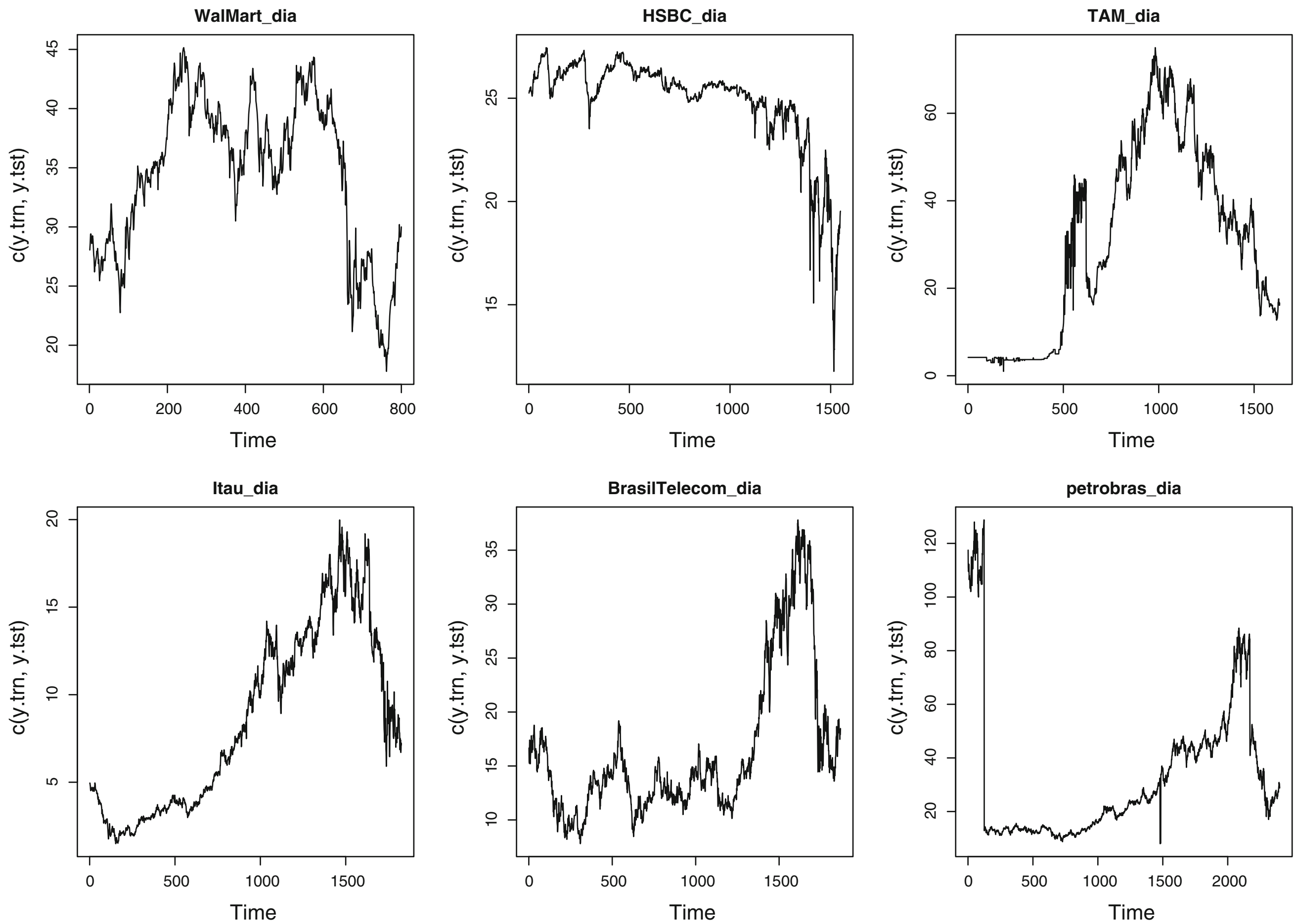

Fig. 4 Daily time series

Table 3 Descriptive statistics of time series

Time series	Average	SD	Kurtosis	Skewness	Minimum	Maximum
Vale do Rio Doce	30.18	10.78	−0.81	−0.02	10.65	51.51
Apple	56.92	33.26	−0.64	0.52	12.50	136.37
Yahoo	56.75	66.51	9.61	2.79	8.81	432.69
Microsoft	66.50	34.03	−0.51	0.46	16.15	175.00
Motorola	50.61	30.58	0.89	0.85	3.52	172.00
GM	58.21	19.43	−0.26	−0.23	1.69	105.00
Wal-Mart	34.40	6.58	−0.79	−0.50	17.79	45.15
HSBC	24.84	2.32	5.89	−2.31	11.77	27.45
TAM	30.07	21.92	−1.26	0.21	1.00	75.00
Itau	8.53	5.14	−1.24	0.34	1.49	19.98
Brasil Telecom	16.42	6.85	1.25	1.49	7.80	37.80
Petrobras	32.67	26.02	2.74	1.75	8.02	128.75

SD standard deviation

Table 6 contains the results of the "Apple" stock market series. For this series, not all the ANNs models presented average performance better than the AR model. This happened for the models with the `sech`, `sinc` and `wave` activation functions, especially with the use of the CGF algorithm. The models with the `cloglogm` function presented the best results for networks with smaller architectures (two and four hidden nodes) independent of the algorithm used, and this series presents a small degree of positive asymmetry (skewness = 0.52), which is an important fact to consider in the analysis of the `cloglog` and `cloglogm` functions. The function `logsigm`

Table 4 Result of selecting the lags through the AR model and test MAPE

Time series	Lags	Test MAPE (%)
Vale do Rio Doce	1	110.6
Apple	1	27.2
Yahoo	3	174.8
Microsoft	2	71.6
Motorola	2	270.9
GM	3	285.7
Wal-Mart	3	16.2
HSBC	19	11.9
TAM	18	19.0
Itau	8	7.7
Brasil Telecom	7	4.6
Petrobras	2	6.9

presented better results with networks containing 8, 12 and 16 hidden nodes using the CGF algorithm and with networks containing 8 and 12 hidden nodes using the LM algorithm. In networks with larger architecture (20 hidden nodes), the models with `logsig` activation function presented the best performance, not only with the use of CGF algorithm but also with the LM algorithm.

Table 7 presents the results of the "Yahoo" stock market series, this series has a very high degree of positive asymmetry (skewness = 2.79). In this case, all the results of ANNs models were better than the result obtained with the AR model. In general, the best average result reached was through the `cloglogm` activation function model, using the CGF algorithm with four hidden nodes. This fact can be confirmed by the confidence interval in which the upper limit is smaller than most of the average results from other models. Besides, the average number of epochs was smaller with the CGF algorithm than with the LM algorithm.

The results of the ANNs models for the "Microsoft" stock market series are presented in Table 8. This series presents a small degree of positive asymmetry (skewness = 0.46). With the use of the CGF algorithm, the best average performance was obtained by the models with the proposed activation functions, `cloglogm`, `probit` and `loglog` in almost all the variations on the number of hidden nodes. We can see that with the use of the LM algorithm, the models with the functions usually found in literature, `logsig` and `tanh`, presented good results with few hidden nodes. However, these results are equivalent to the results obtained with the new proposed functions, using both the learning algorithms CGF and LM. Besides, the average number of epochs needed to achieve convergence with the LM algorithm was larger than the number needed for the CGF algorithm. All the results of ANNs models were better than the results from the AR model, except for the model with the `logsig` activation function with two hidden nodes using CGF algorithm.

The results of the ANNs models for the "Motorola" stock market series are presented in Table 9. This series presents a little higher degree of positive asymmetry (skewness = 0.85). All the results of ANNs models were better than the result of the AR model, except the model with the `sech` activation function with two hidden nodes that used the CGF algorithm. The average results for the models with the new functions, `cloglog`, `cloglogm` and `loglog`, with until six hidden nodes presented good results compared to average performance of other models using the CGF algorithm. Using the LM algorithm, the best performance was observed in the models with `sincos` and `sech` activation functions. However, no significant statistical difference was observed regarding the models with the `cloglog`, `cloglogm` and `loglog` activation functions. It was also observed that the average number of epochs necessary to achieve convergence with the LM algorithm was larger than the number needed for the CGF algorithm.

In Table 10, we present the results for the "GM" stock market series. It was observed that for this case, all the results of ANNs models were better than the results of the AR model. Just like the series "Vale do Rio Doce", this series presents a very small and negative degree of asymmetry (skewness = −0.23). Still, the best result of the network with two hidden nodes was with the `cloglogm` model with the CGF algorithm, a fact that can be proved by 95% CI. With four hidden nodes, the best result was obtained by the `probit` model using the LM algorithm. With six and eight hidden nodes, the best result was obtained by the `cloglogm` model with the LM algorithm. From 12 hidden nodes or more, good results were obtained by the `logsigm`, `rootsig`, `logsig` and `probit` activation functions as well.

The results of the ANNs models for the "Wal-Mart" stock market series are presented in Table 11. This series presents a small degree of negative asymmetry (skewness = −0.50). Almost all the average performances of ANNs models observed were better than those of the AR model, except in the following cases: the models with `sinc` and `wave` activation functions containing two hidden nodes and the model with `sincos` activation function containing 16 and 20 hidden nodes, using the CGF algorithm. We could observe that with the CGF algorithm, the networks of small size, two and four hidden nodes, presented the best results with the `cloglogm`, `probit` and `loglog`. From 6 to 16 hidden nodes, the best results were observed with the `rootsig` and `logsigm` models. Using the LM algorithm, we observed that the average performance was very similar for most of the models studied.

Table 5 Results of the average performance of the forecasting values in models of neural networks for the "Vale do Rio Doce" time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	25.9 - [24.44; 27.36]	42.7 - [36.42; 48.98]	29.9 - [29.17; 30.63]	41.8 - [36.83; 46.77]	47.1 - [41.33; 52.87]	38.3 - [33.96; 42.64]	33.1 - [31.52; 34.68]
`tanh`	39.1 - [33.54; 44.66]	32.6 - [30.36; 34.84]	33.9 - [31.90; 35.90]	**27.2 - [25.69; 28.71]**	**22.6 - [21.71; 23.49]**	**25.4 - [23.79; 27.01]**	29.7 - [28.07; 31.33]
`cloglog`	31.0 - [28.57; 33.43]	**31.1 - [28.96; 33.24]**	28.8 - [27.89; 29.71]	37.1 - [34.58; 39.62]	45.5 - [39.98; 51.02]	36.3 - [33.56; 39.04]	35.2 - [33.16; 37.24]
`cloglogm`	30.6 - [28.57; 32.63]	33.7 - [31.76; 35.64]	30.1 - [28.14; 32.06]	36.5 - [34.50; 38.50]	33.4 - [31.90; 34.90]	28.9 - [28.19; 29.61]	33.0 - [30.96; 35.04]
`probit`	**22.5 - [21.72; 23.28]**	38.0 - [32.77; 43.23]	38.7 - [33.53; 43.87]	31.8 - [30.44; 33.16]	33.9 - [31.61; 36.19]	30.6 - [28.53; 32.67]	**29.6 - [27.65; 31.55]**
`loglog`	42.0 - [32.02; 51.98]	31.7 - [29.66; 33.74]	30.6 - [29.41; 31.79]	30.7 - [29.37; 32.03]	32.5 - [29.82; 35.18]	31.3 - [28.98; 33.62]	35.7 - [34.21; 37.19]
`sech`	30.6 - [27.66; 33.54]	32.8 - [30.69; 34.91]	38.0 - [36.19; 39.81]	34.5 - [32.52; 36.48]	38.9 - [36.41; 41.39]	40.3 - [37.36; 43.24]	30.0 - [26.90; 33.10]
`sinc`	47.5 - [42.47; 52.53]	41.0 - [37.80; 44.20]	32.0 - [29.44; 34.56]	44.5 - [41.68; 47.32]	34.5 - [31.83; 37.17]	48.2 - [44.11; 52.29]	47.0 - [39.71; 54.29]
`wave`	26.8 - [25.57; 28.03]	35.7 - [31.46; 39.94]	34.9 - [32.04; 37.76]	28.3 - [26.14; 30.46]	27.3 - [24.99; 29.61]	33.2 - [30.73; 35.67]	37.1 - [34.19; 40.01]
`sincos`	34.2 - [31.47; 36.93]	33.9 - [31.47; 36.33]	**26.5 - [24.79; 28.21]**	37.1 - [33.75; 40.45]	35.0 - [33.08; 36.92]	27.8 - [25.14; 30.46]	38.4 - [35.09; 41.71]
`rootsig`	38.6 - [32.83; 44.37]	38.6 - [36.23; 40.97]	29.9 - [28.27; 31.53]	34.2 - [32.27; 36.13]	30.7 - [28.42; 32.98]	29.5 - [27.38; 31.62]	36.1 - [33.59; 38.61]
`logsigm`	27.8 - [25.55; 30.05]	42.6 - [37.59; 47.61]	28.5 - [26.89; 30.11]	40.3 - [36.44; 44.16]	32.9 - [30.51; 35.29]	37.9 - [35.50; 40.30]	38.7 - [36.06; 41.34]
Average no. of epochs	15	10	11	10	10	9	9
LM algorithm							
`logsig`	25.7 - [24.66; 26.74]	29.9 - [27.54; 32.26]	**20.6 - [19.95; 21.25]**	24.1 - [22.99; 25.21]	27.5 - [25.14; 29.86]	30.5 - [28.27; 32.73]	31.5 - [28.76; 34.24]
`tanh`	20.0 - [19.49; 20.51]	28.6 - [26.60; 30.60]	23.8 - [22.54; 25.06]	25.5 - [24.02; 26.98]	**22.0 - [21.05; 22.95]**	23.0 - [21.64; 24.36]	26.0 - [24.13; 27.87]
`cloglog`	22.7 - [21.40; 24.00]	26.0 - [23.70; 28.30]	24.6 - [23.49; 25.71]	22.4 - [21.54; 23.26]	31.4 - [29.22; 33.58]	32.4 - [30.30; 34.50]	28.7 - [26.74; 30.66]
`cloglogm`	**19.9 - [19.37; 20.43]**	24.4 - [22.95; 25.85]	25.6 - [23.95; 27.25]	24.3 - [22.72; 25.88]	31.4 - [28.93; 33.87]	26.2 - [24.50; 27.90]	**22.0 - [21.03; 22.97]**
`probit`	21.7 - [20.72; 22.68]	27.6 - [25.57; 29.63]	23.2 - [22.32; 24.08]	27.7 - [25.78; 29.62]	30.1 - [28.06; 32.14]	31.6 - [29.90; 33.30]	25.4 - [24.27; 26.53]
`loglog`	24.0 - [22.81; 25.19]	25.6 - [23.74; 27.46]	25.7 - [23.78; 27.62]	23.3 - [22.13; 24.47]	23.7 - [22.26; 25.14]	27.0 - [25.04; 28.96]	25.1 - [23.76; 26.44]
`sech`	20.4 - [19.26; 21.54]	23.9 - [22.59; 25.21]	23.1 - [21.81; 24.39]	**21.3 - [20.20; 22.40]**	24.1 - [21.92; 26.28]	24.0 - [23.11; 24.89]	23.7 - [21.93; 25.47]
`sinc`	39.5 - [36.62; 42.38]	28.4 - [27.38; 29.42]	29.9 - [26.84; 32.96]	27.1 - [26.22; 27.98]	23.6 - [22.73; 24.47]	24.7 - [23.68; 25.72]	24.9 - [23.67; 26.13]
`wave`	25.6 - [24.26; 26.94]	**23.6 - [22.71; 24.49]**	21.7 - [20.94; 22.46]	26.1 - [25.01; 27.19]	28.1 - [25.48; 30.72]	22.7 - [21.52; 23.88]	25.5 - [24.22; 26.78]
`sincos`	25.4 - [24.25; 26.55]	23.9 - [22.27; 25.53]	23.2 - [22.44; 23.96]	25.1 - [23.79; 26.41]	25.0 - [23.78; 26.22]	**22.0 - [21.44; 22.56]**	24.4 - [22.92; 25.88]
`rootsig`	22.6 - [21.57; 23.63]	28.5 - [26.47; 30.53]	29.1 - [27.11; 31.09]	29.3 - [28.10; 30.50]	27.0 - [25.54; 28.46]	30.7 - [28.34; 33.06]	28.4 - [26.68; 30.12]
`logsigm`	25.7 - [24.93; 26.47]	24.3 - [23.30; 25.30]	25.0 - [23.64; 26.36]	26.7 - [25.63; 27.77]	37.5 - [35.24; 39.76]	30.1 - [27.81; 32.39]	29.9 - [27.84; 31.96]
Average no. of epochs	10	10	10	10	9	8	9

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 6 Results of the average performance of the forecasting values in models of neural networks for the "Apple" time series

Activation function	2 Nodes MAPE - [95% CI]	4 Nodes MAPE - [95% CI]	6 Nodes MAPE - [95% CI]	8 Nodes MAPE - [95% CI]	12 Nodes MAPE - [95% CI]	16 Nodes MAPE - [95% CI]	20 Nodes MAPE - [95% CI]
CGF algorithm							
`logsig`	19.8 - [18.00; 21.60]	25.1 - [20.50; 29.70]	15.3 - [14.82; 15.78]	15.3 - [14.52; 16.08]	18.1 - [17.47; 18.73]	17.5 - [15.52; 19.48]	**14.5 - [14.10; 14.90]**
`tanh`	15.6 - [15.07; 16.13]	16.1 - [14.89; 17.31]	16.9 - [16.15; 17.65]	19.8 - [18.86; 20.74]	20.7 - [18.95; 22.45]	16.7 - [16.18; 17.22]	19.1 - [18.02; 20.18]
`cloglog`	20.1 - [18.42; 21.78]	17.9 - [16.16; 19.64]	15.2 - [14.83; 15.57]	18.9 - [17.46; 20.34]	17.5 - [16.81; 18.19]	15.3 - [14.67; 15.93]	18.3 - [17.23; 19.37]
`cloglogm`	**14.9 - [14.45; 15.35]**	**16.0 - [15.17; 16.83]**	15.9 - [14.91; 16.89]	18.8 - [17.39; 20.21]	18.4 - [17.10; 19.70]	22.6 - [19.17; 26.03]	16.6 - [15.84; 17.36]
`probit`	19.3 - [17.96; 20.64]	19.6 - [17.15; 22.05]	15.1 - [14.66; 15.54]	16.1 - [15.50; 16.70]	18.2 - [17.16; 19.24]	15.1 - [14.52; 15.68]	15.7 - [14.69; 16.71]
`loglog`	16.1 - [15.62; 16.58]	21.0 - [18.95; 23.05]	**14.4 - [13.92; 14.88]**	16.2 - [15.45; 16.95]	16.8 - [15.55; 18.05]	14.3 - [13.73; 14.87]	16.4 - [15.21; 17.59]
`sech`	17.0 - [16.27; 17.73]	28.7 - [24.97; 32.43]	18.3 - [17.10; 19.50]	19.0 - [17.17; 20.83]	21.3 - [18.95; 23.65]	16.7 - [15.88; 17.52]	24.9 - [22.41; 27.39]
`sinc`	23.2 - [21.18; 25.22]	24.0 - [21.57; 26.43]	34.9 - [32.86; 36.94]	37.4 - [33.91; 40.89]	34.2 - [30.69; 37.71]	29.8 - [27.46; 32.14]	47.6 - [41.48; 53.72]
`wave`	26.9 - [24.56; 29.24]	21.3 - [19.60; 23.00]	26.7 - [24.34; 29.06]	30.4 - [27.93; 32.87]	35.5 - [31.93; 39.07]	28.8 - [26.82; 30.78]	37.0 - [34.34; 39.66]
`sincos`	17.5 - [16.52; 18.48]	20.7 - [19.23; 22.17]	24.3 - [21.46; 27.14]	26.5 - [24.81; 28.19]	24.2 - [22.46; 25.94]	24.5 - [22.93; 26.07]	31.7 - [28.56; 34.84]
`rootsig`	20.0 - [17.42; 22.58]	16.5 - [15.77; 17.23]	14.9 - [14.61; 15.19]	16.2 - [15.46; 16.94]	16.6 - [15.43; 17.77]	15.4 - [14.92; 15.88]	16.1 - [15.39; 16.81]
`logsigm`	19.1 - [17.22; 20.98]	18.5 - [17.06; 19.94]	15.5 - [14.88; 16.12]	**15.2 - [14.30; 16.10]**	**15.0 - [14.02; 15.98]**	**13.9 - [13.47; 14.33]**	20.4 - [17.29; 23.51]
Average no. of epochs	13	12	11	11	11	10	9
LM algorithm							
`logsig`	14.3 - [14.07; 14.53]	15.4 - [15.04; 15.76]	**13.7 - [13.53; 13.87]**	14.4 - [14.22; 14.58]	14.8 - [14.37; 15.23]	14.8 - [14.56; 15.04]	**13.6 - [13.41; 13.79]**
`tanh`	14.7 - [14.37; 15.03]	14.0 - [13.70; 14.30]	14.8 - [14.58; 15.02]	15.6 - [15.43; 15.77]	15.4 - [14.87; 15.93]	15.2 - [14.93; 15.47]	15.0 - [14.61; 15.39]
`cloglog`	15.1 - [14.80; 15.40]	14.4 - [14.11; 14.69]	13.8 - [13.44; 14.16]	15.6 - [15.22; 15.98]	14.4 - [14.18; 14.62]	15.1 - [14.71; 15.49]	14.9 - [14.59; 15.21]
`cloglogm`	**13.9 - [13.74; 14.06]**	**13.5 - [13.14; 13.86]**	15.7 - [14.75; 16.65]	14.3 - [14.01; 14.59]	14.9 - [14.64; 15.16]	15.6 - [15.36; 15.84]	15.1 - [14.74; 15.46]
`probit`	15.4 - [14.87; 15.93]	15.8 - [14.66; 16.94]	13.7 - [13.37; 14.03]	15.0 - [14.52; 15.48]	16.0 - [15.18; 16.82]	14.4 - [14.12; 14.68]	14.9 - [14.66; 15.14]
`loglog`	15.9 - [15.29; 16.51]	16.3 - [14.63; 17.97]	14.2 - [13.78; 14.62]	15.3 - [14.92; 15.68]	15.0 - [14.38; 15.62]	**14.2 - [13.81; 14.59]**	14.5 - [14.30; 14.70]
`sech`	18.1 - [17.57; 18.63]	15.1 - [14.47; 15.73]	15.2 - [14.87; 15.53]	16.7 - [16.50; 16.90]	15.6 - [15.20; 16.00]	16.1 - [15.75; 16.45]	16.7 - [16.38; 17.02]
`sinc`	29.0 - [25.63; 32.37]	18.2 - [17.37; 19.03]	17.1 - [16.57; 17.63]	15.1 - [14.43; 15.77]	16.1 - [15.40; 16.80]	24.2 - [22.40; 26.00]	23.0 - [21.57; 24.43]
`wave`	20.1 - [18.56; 21.64]	16.0 - [15.46; 16.54]	19.0 - [18.38; 19.62]	18.4 - [17.79; 19.01]	17.7 - [17.25; 18.15]	19.0 - [18.14; 19.86]	21.4 - [19.75; 23.05]
`sincos`	15.6 - [14.68; 16.52]	15.5 - [15.16; 15.84]	15.9 - [15.60; 16.20]	19.2 - [18.53; 19.87]	16.8 - [16.41; 17.19]	16.9 - [16.29; 17.51]	16.0 - [15.34; 16.66]
`rootsig`	14.5 - [14.25; 14.75]	13.8 - [13.52; 14.08]	14.5 - [14.20; 14.80]	14.5 - [14.35; 14.65]	14.9 - [14.70; 15.10]	14.4 - [14.20; 14.60]	14.2 - [13.99; 14.41]
`logsigm`	15.0 - [14.79; 15.21]	16.8 - [15.57; 18.03]	14.2 - [13.89; 14.51]	**13.7 - [13.41; 13.99]**	**13.5 - [13.25; 13.75]**	14.4 - [14.19; 14.61]	14.1 - [13.91; 14.29]
Average no. of epochs	12	12	19	16	12	14	11

CGF conjugate gradient backpropagation with Fletcher-Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 7 Results of the average performance of the forecasting values in models of neural networks for the "Yahoo" time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	48.0 - [30.34; 65.66]	27.5 - [25.57; 29.43]	52.5 - [36.49; 68.51]	24.4 - [20.83; 27.97]	37.0 - [34.07; 39.93]	47.0 - [36.22; 57.78]	57.1 - [44.32; 69.88]
`tanh`	34.0 - [26.20; 41.80]	19.2 - [18.21; 20.19]	29.2 - [24.72; 33.68]	21.1 - [18.78; 23.42]	26.2 - [22.24; 30.16]	22.4 - [20.92; 23.88]	22.8 - [21.42; 24.18]
`cloglog`	49.5 - [31.75; 67.25]	29.5 - [25.49; 33.51]	**20.2 - [17.22; 23.18]**	21.1 - [19.87; 22.33]	23.3 - [20.24; 26.36]	27.7 - [24.85; 30.55]	40.0 - [30.63; 49.37]
`cloglogm`	41.4 - [30.96; 51.84]	**16.1 - [15.00; 17.20]**	28.6 - [25.35; 31.85]	17.8 - [16.86; 18.74]	**22.3 - [19.88; 24.72]**	27.1 - [24.96; 29.24]	32.3 - [25.86; 38.74]
`probit`	**23.0 - [20.67; 25.33]**	23.0 - [20.43; 25.57]	21.1 - [19.50; 22.70]	31.6 - [27.80; 35.40]	32.8 - [28.38; 37.22]	25.2 - [22.21; 28.19]	**22.1 - [19.70; 24.50]**
`loglog`	79.7 - [57.13; 102.27]	26.1 - [21.51; 30.69]	30.4 - [26.35; 34.45]	25.2 - [21.81; 28.59]	46.1 - [39.06; 53.14]	28.5 - [25.94; 31.06]	45.2 - [38.22; 52.18]
`sech`	47.2 - [35.00; 59.40]	68.3 - [56.06; 80.54]	45.4 - [32.47; 58.33]	20.5 - [19.34; 21.66]	23.4 - [21.89; 24.91]	27.3 - [23.91; 30.69]	33.2 - [29.97; 36.43]
`sinc`	63.9 - [47.61; 80.19]	31.9 - [27.94; 35.86]	36.6 - [29.11; 44.09]	53.2 - [46.84; 59.56]	40.4 - [33.13; 47.67]	34.4 - [29.40; 39.40]	34.3 - [30.09; 38.51]
`wave`	39.8 - [34.30; 45.30]	29.5 - [26.48; 32.52]	40.8 - [32.47; 49.13]	33.6 - [30.42; 36.78]	23.8 - [21.42; 26.18]	22.3 - [20.52; 24.08]	58.6 - [47.53; 69.67]
`sincos`	60.7 - [43.01; 78.39]	33.0 - [28.09; 37.91]	25.5 - [23.05; 27.95]	19.7 - [18.31; 21.09]	25.8 - [22.90; 28.70]	30.6 - [26.49; 34.71]	46.6 - [32.87; 60.33]
`rootsig`	50.4 - [32.70; 68.10]	17.1 - [15.97; 18.23]	23.6 - [21.16; 26.04]	**17.4 - [16.54; 18.26]**	23.4 - [19.59; 27.21]	**21.5 - [19.77; 23.23]**	31.5 - [28.39; 34.61]
`logsigm`	53.9 - [36.41; 71.39]	37.8 - [27.09; 48.51]	20.3 - [17.83; 22.77]	28.8 - [24.41; 33.19]	24.1 - [21.26; 26.94]	31.0 - [26.25; 35.75]	53.0 - [45.03; 60.97]
Average no. of epochs	12	14	13	13	11	11	11
LM algorithm							
`logsig`	21.0 - [19.63; 22.37]	29.3 - [26.70; 31.90]	24.8 - [23.77; 25.83]	23.7 - [22.49; 24.91]	24.2 - [22.83; 25.57]	22.3 - [21.32; 23.28]	24.1 - [22.86; 25.34]
`tanh`	31.2 - [30.02; 32.38]	25.4 - [23.86; 26.94]	27.4 - [26.76; 28.04]	26.2 - [24.85; 27.55]	27.5 - [26.84; 28.16]	23.7 - [22.37; 25.03]	26.1 - [24.86; 27.34]
`cloglog`	28.1 - [26.83; 29.37]	30.9 - [30.28; 31.52]	25.6 - [24.18; 27.02]	24.6 - [23.14; 26.06]	26.3 - [25.10; 27.50]	21.1 - [19.58; 22.62]	**22.9 - [21.53; 24.27]**
`cloglogm`	26.8 - [25.27; 28.33]	26.4 - [25.10; 27.70]	25.9 - [24.58; 27.22]	25.9 - [24.65; 27.15]	25.5 - [24.24; 26.76]	30.5 - [27.71; 33.29]	26.9 - [26.03; 27.77]
`probit`	24.5 - [23.23; 25.77]	29.2 - [27.66; 30.74]	27.4 - [26.04; 28.76]	25.9 - [25.03; 26.77]	25.0 - [23.74; 26.26]	20.8 - [20.00; 21.60]	25.9 - [25.24; 26.56]
`loglog`	30.2 - [28.91; 31.49]	28.9 - [25.24; 32.56]	26.6 - [25.28; 27.92]	**23.5 - [22.25; 24.75]**	23.8 - [22.55; 25.05]	22.9 - [21.70; 24.10]	26.7 - [25.41; 27.99]
`sech`	45.2 - [35.52; 54.88]	29.1 - [27.58; 30.62]	27.5 - [26.15; 28.85]	27.9 - [27.19; 28.61]	27.4 - [26.52; 28.28]	27.2 - [25.72; 28.68]	26.7 - [26.20; 27.20]
`sinc`	23.9 - [20.93; 26.87]	26.1 - [22.04; 30.16]	**20.8 - [19.18; 22.42]**	25.5 - [23.64; 27.36]	24.7 - [23.40; 26.00]	26.3 - [25.44; 27.16]	25.6 - [24.67; 26.53]
`wave`	35.3 - [34.04; 36.56]	28.3 - [26.97; 29.63]	25.8 - [24.82; 26.78]	28.3 - [26.62; 29.98]	25.8 - [24.18; 27.42]	27.1 - [26.31; 27.89]	24.6 - [23.59; 25.61]
`sincos`	27.9 - [26.52; 29.28]	24.6 - [23.46; 25.74]	26.6 - [25.13; 28.07]	27.2 - [26.26; 28.14]	28.7 - [28.02; 29.38]	28.2 - [27.66; 28.74]	27.4 - [26.71; 28.09]
`rootsig`	23.3 - [21.97; 24.63]	**20.4 - [18.74; 22.06]**	23.2 - [21.74; 24.66]	26.0 - [24.58; 27.42]	27.7 - [24.58; 30.82]	23.5 - [21.85; 25.15]	24.7 - [23.02; 26.38]
`logsigm`	**20.1 - [18.70; 21.50]**	30.5 - [27.01; 33.99]	24.9 - [23.67; 26.13]	25.2 - [23.97; 26.43]	**21.6 - [20.05; 23.15]**	**20.0 - [18.64; 21.36]**	23.4 - [22.02; 24.78]
Average no. of epochs	26	18	29	31	17	23	18

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 8 Results of the average performance of the forecasting values in models of neural networks for the "Microsoft" time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	74.0 - [57.37; 90.63]	19.3 - [14.30; 24.30]	13.2 - [12.25; 14.15]	13.8 - [12.49; 15.11]	26.8 - [22.26; 31.34]	13.9 - [11.93; 15.87]	14.7 - [13.48; 15.92]
`tanh`	14.5 - [13.00; 16.00]	12.8 - [11.55; 14.05]	13.5 - [12.12; 14.88]	13.8 - [12.63; 14.97]	13.0 - [11.98; 14.02]	**10.4 - [10.02; 10.78]**	14.4 - [12.32; 16.48]
`cloglog`	31.3 - [22.63; 39.97]	27.8 - [18.25; 37.35]	12.9 - [11.98; 13.82]	11.6 - [10.57; 12.63]	22.9 - [19.37; 26.43]	14.0 - [12.54; 15.46]	40.7 - [30.56; 50.84]
`cloglogm`	**10.5 - [9.71; 11.29]**	10.1 - [9.51; 10.69]	11.0 - [10.41; 11.59]	13.2 - [11.96; 14.44]	**12.4 - [11.48; 13.32]**	12.1 - [11.16; 13.04]	**10.1 - [9.62; 10.58]**
`probit`	33.6 - [25.96; 41.24]	**9.9 - [9.62; 10.18]**	11.7 - [11.10; 12.30]	**9.7 - [9.44; 9.96]**	18.1 - [15.33; 20.87]	24.8 - [18.08; 31.52]	23.1 - [17.24; 28.96]
`loglog`	26.7 - [20.59; 32.81]	11.4 - [10.84; 11.96]	**9.7 - [9.48; 9.92]**	9.7 - [9.09; 10.31]	14.6 - [12.01; 17.19]	15.1 - [13.68; 16.52]	17.6 - [15.69; 19.51]
`sech`	53.1 - [36.83; 69.37]	24.0 - [15.90; 32.10]	19.9 - [13.82; 25.98]	13.7 - [11.98; 15.42]	14.9 - [11.85; 17.95]	16.4 - [13.96; 18.84]	14.1 - [12.85; 15.35]
`sinc`	37.4 - [29.50; 45.30]	17.1 - [14.89; 19.31]	12.6 - [11.53; 13.67]	20.2 - [17.24; 23.16]	33.2 - [24.71; 41.69]	20.1 - [18.51; 21.69]	22.1 - [19.15; 25.05]
`wave`	19.4 - [17.08; 21.72]	13.8 - [12.53; 15.07]	21.0 - [17.06; 24.94]	11.8 - [11.17; 12.43]	12.5 - [11.49; 13.51]	17.4 - [15.67; 19.13]	20.0 - [16.90; 23.10]
`sincos`	28.6 - [17.28; 39.92]	24.3 - [18.81; 29.79]	11.2 - [10.12; 12.28]	17.6 - [15.89; 19.31]	13.6 - [11.76; 15.44]	32.1 - [22.83; 41.37]	29.4 - [23.70; 35.10]
`rootsig`	22.8 - [17.99; 27.61]	10.8 - [9.78; 11.82]	10.7 - [10.21; 11.19]	11.1 - [10.57; 11.63]	18.4 - [15.12; 21.68]	10.4 - [9.95; 10.85]	12.4 - [10.98; 13.82]
`logsigm`	28.2 - [22.70; 33.70]	10.5 - [10.08; 10.92]	36.3 - [22.65; 49.95]	21.7 - [16.01; 27.39]	13.2 - [12.46; 13.94]	14.9 - [12.74; 17.06]	13.9 - [12.97; 14.83]
Average no. of epochs	13	13	13	13	12	11	13
LM algorithm							
`logsig`	**9.6 - [9.38; 9.82]**	9.6 - [9.37; 9.83]	8.8 - [8.63; 8.97]	9.5 - [9.33; 9.67]	10.4 - [9.78; 11.02]	10.0 - [9.81; 10.19]	10.4 - [10.03; 10.77]
`tanh`	9.8 - [9.60; 10.00]	**9.5 - [9.24; 9.76]**	9.4 - [9.18; 9.62]	9.4 - [9.06; 9.74]	10.3 - [9.89; 10.71]	9.9 - [9.79; 10.01]	**9.3 - [9.08; 9.52]**
`cloglog`	13.5 - [11.24; 15.76]	10.1 - [9.86; 10.34]	9.6 - [9.21; 9.99]	10.9 - [10.24; 11.56]	10.9 - [10.43; 11.37]	10.1 - [9.94; 10.26]	9.8 - [9.66; 9.94]
`cloglogm`	9.7 - [9.54; 9.86]	11.4 - [10.80; 12.00]	10.7 - [10.34; 11.06]	10.5 - [9.94; 11.06]	**9.3 - [9.03; 9.57]**	**9.5 - [9.27; 9.73]**	9.6 - [9.41; 9.79]
`probit`	12.0 - [10.82; 13.18]	9.8 - [9.58; 10.02]	9.2 - [8.98; 9.42]	**9.2 - [8.81; 9.59]**	9.6 - [9.39; 9.81]	10.2 - [10.08; 10.32]	10.1 - [9.84; 10.36]
`loglog`	11.2 - [10.58; 11.82]	9.6 - [9.35; 9.85]	9.7 - [9.36; 10.04]	9.7 - [9.29; 10.11]	9.4 - [9.25; 9.55]	11.1 - [10.55; 11.65]	10.0 - [9.72; 10.28]
`sech`	10.4 - [10.15; 10.65]	10.6 - [10.26; 10.94]	9.5 - [9.28; 9.72]	10.1 - [9.97; 10.23]	9.8 - [9.60; 10.00]	11.9 - [10.21; 13.59]	10.2 - [9.89; 10.51]
`sinc`	14.7 - [12.51; 16.89]	10.4 - [9.88; 10.92]	**8.6 - [8.33; 8.87]**	10.8 - [10.13; 11.47]	10.9 - [9.84; 11.96]	10.9 - [10.24; 11.56]	11.5 - [10.68; 12.32]
`wave`	10.2 - [9.96; 10.44]	10.3 - [10.07; 10.53]	8.9 - [8.61; 9.19]	9.3 - [9.04; 9.56]	10.0 - [9.58; 10.42]	10.1 - [9.53; 10.67]	12.3 - [11.37; 13.23]
`sincos`	10.2 - [10.02; 10.38]	10.4 - [10.02; 10.78]	9.9 - [9.68; 10.12]	10.4 - [10.19; 10.61]	9.8 - [9.60; 10.00]	10.4 - [10.23; 10.57]	10.1 - [9.96; 10.24]
`rootsig`	10.2 - [10.06; 10.34]	11.7 - [10.99; 12.41]	9.6 - [9.37; 9.83]	10.0 - [9.64; 10.36]	9.9 - [9.64; 10.16]	10.4 - [10.32; 10.48]	9.6 - [9.42; 9.78]
`logsigm`	17.5 - [13.02; 21.98]	9.7 - [9.44; 9.96]	10.7 - [9.84; 11.56]	9.7 - [9.27; 10.13]	10.4 - [9.62; 11.18]	10.3 - [10.19; 10.41]	9.4 - [9.26; 9.54]
Average no. of epochs	22	19	15	22	16	14	14

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 9 Results of the average performance of the forecasting values in models of neural networks for the "Motorola" time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	29.9 - [26.98; 32.82]	34.8 - [29.98; 39.62]	23.2 - [21.39; 25.01]	25.3 - [23.51; 27.09]	96.0 - [58.79; 133.21]	69.2 - [46.99; 91.41]	46.5 - [43.15; 49.85]
`tanh`	32.7 - [29.21; 36.19]	31.3 - [26.52; 36.08]	31.0 - [27.32; 34.68]	**21.8 - [21.14; 22.46]**	41.0 - [36.82; 45.18]	60.8 - [53.52; 68.08]	59.3 - [50.45; 68.15]
`cloglog`	27.1 - [23.08; 31.12]	**25.8 - [24.32; 27.28]**	34.0 - [30.47; 37.53]	27.8 - [25.29; 30.31]	50.5 - [42.79; 58.21]	106.3 - [62.76; 149.84]	67.0 - [51.24; 82.76]
`cloglogm`	**20.6 - [20.09; 21.11]**	27.6 - [23.77; 31.43]	29.8 - [27.04; 32.56]	24.3 - [22.58; 26.02]	41.2 - [33.44; 48.96]	27.7 - [24.64; 30.76]	49.4 - [42.02; 56.78]
`probit`	28.0 - [24.32; 31.68]	101.6 - [55.06; 148.14]	33.8 - [29.35; 38.25]	29.4 - [23.34; 35.46]	34.8 - [31.55; 38.05]	**27.4 - [24.55; 30.25]**	38.0 - [33.62; 42.38]
`loglog`	33.7 - [28.60; 38.80]	86.1 - [55.39; 116.81]	**23.0 - [22.10; 23.90]**	24.2 - [22.61; 25.79]	38.2 - [33.01; 43.39]	40.0 - [33.68; 46.32]	42.4 - [37.81; 46.99]
`sech`	297.0 - [224.46; 369.54]	86.4 - [57.61; 115.19]	147.5 - [103.97; 191.03]	25.4 - [23.76; 27.04]	50.9 - [33.16; 68.64]	76.7 - [51.7; 101.7]	101.4 - [66.28; 136.52]
`sinc`	152.2 - [104.07; 200.33]	50.7 - [44.03; 57.37]	42.1 - [36.89; 47.31]	71.6 - [54.80; 88.40]	74.9 - [64.16; 85.64]	76.8 - [56.56; 97.04]	85.1 - [75.11; 95.09]
`wave`	50.9 - [41.53; 60.27]	72.3 - [46.19; 98.41]	44.4 - [36.60; 52.20]	72.9 - [49.79; 96.01]	34.6 - [27.93; 41.27]	39.1 - [34.29; 43.91]	32.7 - [28.33; 37.07]
`sincos`	115.9 - [67.25; 164.55]	45.3 - [38.65; 51.95]	62.8 - [52.24; 73.36]	41.3 - [33.03; 49.57]	122.5 - [62.65; 182.35]	101.9 - [85.15; 118.65]	149.5 - [84.22; 214.78]
`rootsig`	40.7 - [33.38; 48.02]	31.4 - [27.60; 35.20]	25.6 - [22.31; 28.89]	25.6 - [22.94; 28.26]	**29.1 - [26.59; 31.61]**	27.7 - [25.67; 29.73]	**26.6 - [24.22; 28.98]**
`logsigm`	108.7 - [68.26; 149.14]	56.0 - [42.95; 69.05]	28.0 - [25.76; 30.24]	24.1 - [22.16; 26.04]	35.0 - [30.29; 39.71]	29.7 - [26.92; 32.48]	49.8 - [42.22; 57.38]
Average no. of epochs	11	13	11	12	9	11	9
LM algorithm							
`logsig`	19.6 - [19.36; 19.84]	20.3 - [19.83; 20.77]	19.9 - [19.52; 20.28]	20.0 - [19.68; 20.32]	19.4 - [19.32; 19.48]	19.4 - [19.35; 19.45]	19.4 - [19.33; 19.47]
`tanh`	19.3 - [19.13; 19.47]	21.1 - [20.31; 21.89]	21.1 - [20.38; 21.82]	20.3 - [19.79; 20.81]	22.5 - [21.56; 23.44]	22.1 - [21.10; 23.10]	22.2 - [21.46; 22.94]
`cloglog`	19.5 - [19.21; 19.79]	19.2 - [19.01; 19.39]	21.5 - [20.94; 22.06]	22.1 - [21.07; 23.13]	20.9 - [19.70; 22.10]	19.4 - [18.98; 19.82]	20.7 - [19.84; 21.56]
`cloglogm`	19.8 - [19.59; 20.01]	20.1 - [19.55; 20.65]	19.3 - [19.23; 19.37]	**19.2 - [19.10; 19.30]**	20.6 - [19.99; 21.21]	19.9 - [19.48; 20.32]	19.8 - [19.43; 20.17]
`probit`	20.1 - [19.61; 20.59]	19.2 - [19.06; 19.34]	20.7 - [20.08; 21.32]	20.6 - [20.07; 21.13]	21.7 - [20.90; 22.50]	22.2 - [21.25; 23.15]	23.3 - [22.00; 24.60]
`loglog`	21.5 - [20.57; 22.43]	20.4 - [19.83; 20.97]	19.5 - [19.31; 19.69]	20.1 - [19.54; 20.66]	19.9 - [19.59; 20.21]	19.5 - [19.12; 19.88]	19.7 - [19.05; 20.35]
`sech`	20.4 - [19.96; 20.84]	19.6 - [19.26; 19.94]	**19.2 - [19.07; 19.33]**	19.2 - [19.01; 19.39]	21.7 - [20.45; 22.95]	26.5 - [24.33; 28.67]	22.8 - [21.60; 24.00]
`sinc`	52.2 - [40.45; 63.95]	21.6 - [20.30; 22.90]	20.1 - [19.39; 20.81]	19.5 - [19.12; 19.88]	21.6 - [20.44; 22.76]	23.2 - [22.32; 24.08]	26.1 - [24.50; 27.70]
`wave`	27.6 - [24.72; 30.48]	19.9 - [19.31; 20.49]	20.5 - [19.97; 21.03]	22.7 - [21.46; 23.94]	22.0 - [21.16; 22.84]	21.3 - [20.27; 22.33]	21.7 - [20.62; 22.78]
`sincos`	**19.1 - [19.01; 19.19]**	**19.0 - [18.85; 19.15]**	19.3 - [19.24; 19.36]	21.2 - [20.28; 22.12]	**19.1 - [19.02; 19.18]**	**19.3 - [19.27; 19.33]**	19.4 - [19.36; 19.44]
`rootsig`	21.5 - [20.77; 22.23]	19.3 - [19.19; 19.41]	20.4 - [20.03; 20.77]	20.9 - [20.28; 21.52]	20.0 - [19.57; 20.43]	21.2 - [20.61; 21.79]	21.7 - [20.65; 22.75]
`logsigm`	19.4 - [19.14; 19.66]	20.3 - [20.01; 20.59]	22.2 - [20.63; 23.77]	20.0 - [19.61; 20.39]	20.2 - [19.58; 20.82]	22.5 - [21.46; 23.54]	**19.2 - [19.07; 19.33]**
Average no. of epochs	44	39	42	21	24	24	25

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 10 Results of the average performance of the forecasting values in models of neural networks for the "GM" time series

Activation function	2 Nodes MAPE - [95% CI]	4 Nodes MAPE - [95% CI]	6 Nodes MAPE - [95% CI]	8 Nodes MAPE - [95% CI]	12 Nodes MAPE - [95% CI]	16 Nodes MAPE - [95% CI]	20 Nodes MAPE - [95% CI]
CGF algorithm							
logsig	74.2 - [57.07; 91.33]	64.8 - [52.28; 77.32]	34.8 - [32.66; 36.94]	40.8 - [37.36; 44.24]	47.9 - [40.83; 54.97]	**35.8 - [33.70; 37.90]**	51.7 - [38.51; 64.89]
tanh	41.1 - [38.15; 44.05]	40.5 - [37.89; 43.11]	34.1 - [31.82; 36.38]	39.6 - [36.59; 42.61]	46.3 - [40.00; 52.60]	42.6 - [39.41; 45.79]	49.8 - [45.44; 54.16]
cloglog	66.1 - [47.47; 84.73]	50.6 - [44.04; 57.16]	**31.4 - [29.85; 32.95]**	36.6 - [34.09; 39.11]	37.1 - [33.59; 40.61]	38.1 - [36.17; 40.03]	54.6 - [40.36; 68.84]
cloglogm	**32.0 - [30.17; 33.83]**	40.0 - [35.69; 44.31]	41.6 - [37.77; 45.43]	36.6 - [32.95; 40.25]	35.0 - [32.67; 37.33]	45.7 - [41.39; 50.01]	39.7 - [36.35; 43.05]
probit	67.6 - [49.47; 85.73]	41.5 - [38.58; 44.42]	37.9 - [34.75; 41.05]	67.7 - [48.39; 87.01]	34.5 - [32.18; 36.82]	38.6 - [36.20; 41.00]	**25.7 - [23.38; 28.02]**
loglog	56.1 - [47.08; 65.12]	38.9 - [37.00; 40.80]	34.8 - [32.83; 36.77]	37.4 - [33.77; 41.03]	32.6 - [30.10; 35.10]	38.1 - [35.05; 41.15]	29.3 - [27.48; 31.12]
sech	79.5 - [62.47; 96.53]	**36.1 - [33.74; 38.46]**	53.4 - [48.85; 57.95]	54.1 - [49.16; 59.04]	70.9 - [63.79; 78.01]	60.8 - [55.47; 66.13]	78.8 - [67.77; 89.83]
sinc	96.8 - [86.41; 107.19]	98.6 - [80.82; 116.38]	113.8 - [96.91; 130.69]	134.0 - [112.16; 155.84]	172.4 - [156.29; 188.51]	196.2 - [163.47; 228.93]	175.8 - [150.45; 201.15]
wave	102.6 - [82.07; 123.13]	82.4 - [70.90; 93.90]	102.2 - [90.74; 113.66]	103.6 - [92.49; 114.71]	115.3 - [100.79; 129.81]	114.9 - [91.17; 138.63]	110.2 - [96.71; 123.69]
sincos	73.8 - [55.35; 92.25]	50.2 - [42.72; 57.68]	76.4 - [67.99; 84.81]	54.2 - [50.03; 58.37]	76.4 - [64.16; 88.64]	59.1 - [53.28; 64.92]	77.6 - [61.76; 93.44]
rootsig	61.8 - [47.53; 76.07]	39.7 - [38.11; 41.29]	34.5 - [32.35; 36.65]	**35.7 - [33.53; 37.87]**	**29.2 - [27.03; 31.37]**	36.2 - [34.75; 37.65]	34.0 - [30.87; 37.13]
logsigm	67.3 - [53.11; 81.49]	50.3 - [46.28; 54.32]	38.3 - [35.71; 40.89]	48.8 - [42.85; 54.75]	47.1 - [40.36; 53.84]	42.8 - [39.26; 46.34]	65.6 - [47.52; 83.68]
Average no. of epochs	12	13	12	11	11	12	10
LM algorithm							
logsig	**33.8 - [30.91; 36.69]**	42.2 - [33.88; 50.52]	31.0 - [29.45; 32.55]	46.7 - [42.60; 50.80]	32.5 - [29.60; 35.40]	**32.9 - [30.63; 35.17]**	**26.4 - [25.21; 27.59]**
tanh	57.1 - [48.84; 65.36]	30.7 - [29.50; 31.90]	32.6 - [31.26; 33.94]	35.3 - [32.96; 37.64]	34.4 - [32.55; 36.25]	38.2 - [36.41; 39.99]	36.2 - [34.10; 38.30]
cloglog	47.8 - [42.28; 53.32]	33.2 - [31.03; 35.37]	30.3 - [29.34; 31.26]	32.3 - [30.60; 34.00]	33.9 - [31.11; 36.69]	33.2 - [30.59; 35.81]	28.8 - [27.53; 30.07]
cloglogm	36.8 - [30.47; 43.13]	29.2 - [28.29; 30.11]	**29.7 - [28.48; 30.92]**	**26.8 - [25.61; 27.99]**	30.3 - [28.76; 31.84]	36.6 - [33.81; 39.39]	31.9 - [29.93; 33.87]
probit	35.7 - [31.72; 39.68]	**29.0 - [28.26; 29.74]**	30.8 - [29.73; 31.87]	34.6 - [32.16; 37.04]	32.5 - [31.17; 33.83]	35.5 - [33.24; 37.76]	28.9 - [27.28; 30.52]
loglog	44.4 - [39.03; 49.77]	30.7 - [29.67; 31.73]	33.4 - [31.97; 34.83]	39.4 - [36.20; 42.60]	31.8 - [30.21; 33.39]	35.0 - [33.11; 36.89]	26.7 - [25.59; 27.81]
sech	55.7 - [52.06; 59.34]	43.2 - [40.42; 45.98]	42.3 - [39.93; 44.67]	45.3 - [41.33; 49.27]	55.6 - [49.85; 61.35]	44.3 - [42.77; 45.83]	36.9 - [34.85; 38.95]
sinc	90.1 - [83.22; 96.98]	50.7 - [48.16; 53.24]	55.4 - [45.63; 65.17]	46.5 - [41.92; 51.08]	43.6 - [38.70; 48.50]	56.2 - [49.86; 62.54]	39.3 - [36.69; 41.91]
wave	59.1 - [54.43; 63.77]	40.6 - [38.19; 43.01]	40.6 - [37.22; 43.98]	41.6 - [39.32; 43.88]	46.8 - [44.92; 48.68]	51.4 - [49.18; 53.62]	47.2 - [43.76; 50.64]
sincos	47.5 - [42.20; 52.80]	31.7 - [30.19; 33.21]	36.0 - [33.93; 38.07]	34.3 - [32.16; 36.44]	31.3 - [30.03; 32.57]	37.6 - [34.35; 40.85]	33.6 - [31.50; 35.70]
rootsig	47.6 - [41.51; 53.69]	33.1 - [31.01; 35.19]	37.5 - [33.36; 41.64]	30.6 - [28.68; 32.52]	32.8 - [31.51; 34.09]	34.3 - [31.87; 36.73]	29.2 - [27.81; 30.59]
logsigm	34.1 - [31.90; 36.30]	36.6 - [33.53; 39.67]	32.2 - [31.09; 33.31]	31.8 - [30.56; 33.04]	**29.6 - [28.73; 30.47]**	36.7 - [33.86; 39.54]	37.0 - [34.77; 39.23]
Average no. of epochs	11	10	9	10	17	11	19

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 11 Results of the average performance of the forecasting values in models of neural networks for the “Wal-Mart” time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	14.6 - [10.91; 18.29]	6.0 - [4.44; 7.56]	4.5 - [3.74; 5.26]	11.5 - [8.49; 14.51]	5.4 - [3.97; 6.83]	6.4 - [4.65; 8.15]	5.6 - [4.54; 6.66]
`tanh`	11.2 - [7.13; 15.27]	4.0 - [3.56; 4.44]	4.8 - [4.14; 5.46]	4.3 - [3.71; 4.89]	4.1 - [3.80; 4.40]	5.2 - [4.68; 5.72]	5.3 - [4.45; 6.15]
`cloglog`	12.0 - [8.55; 15.45]	6.6 - [4.81; 8.39]	6.9 - [4.94; 8.86]	10.9 - [8.21; 13.59]	3.1 - [2.94; 3.26]	8.9 - [6.56; 11.24]	4.5 - [3.75; 5.25]
`cloglogm`	3.2 - [3.08; 3.32]	**2.8 - [2.68; 2.92]**	3.5 - [3.28; 3.72]	6.7 - [4.34; 9.06]	3.3 - [3.11; 3.49]	3.9 - [3.45; 4.35]	3.6 - [3.08; 4.12]
`probit`	**3.1 - [2.94; 3.26]**	3.1 - [2.96; 3.24]	10.0 - [6.44; 13.56]	6.6 - [4.90; 8.30]	3.5 - [3.26; 3.74]	2.8 - [2.72; 2.88]	4.1 - [3.65; 4.55]
`loglog`	13.8 - [9.00; 18.60]	**2.8 - [2.69; 2.91]**	10.2 - [7.62; 12.78]	7.5 - [4.59; 10.41]	3.3 - [3.08; 3.52]	3.1 - [2.98; 3.22]	**3.5 - [3.13; 3.87]**
`sech`	10.4 - [7.47; 13.33]	8.3 - [5.48; 11.12]	7.9 - [5.52; 10.28]	3.8 - [3.33; 4.27]	2.9 - [2.75; 3.05]	5.7 - [4.66; 6.74]	4.3 - [4.06; 4.54]
`sinc`	27.9 - [22.28; 33.52]	7.8 - [6.37; 9.23]	4.3 - [4.02; 4.58]	10.6 - [8.46; 12.74]	8.8 - [7.59; 10.01]	9.4 - [8.33; 10.47]	14.7 - [13.21; 16.19]
`wave`	23.3 - [17.25; 29.35]	7.6 - [6.02; 9.18]	4.7 - [4.35; 5.05]	6.3 - [4.82; 7.78]	5.6 - [4.41; 6.79]	4.9 - [4.37; 5.43]	6.4 - [5.51; 7.29]
`sincos`	14.8 - [10.40; 19.20]	4.2 - [3.84; 4.56]	4.0 - [3.43; 4.57]	4.6 - [4.32; 4.88]	3.8 - [3.40; 4.20]	16.4 - [11.24; 21.56]	17.3 - [10.7; 23.9]
`rootsig`	8.4 - [6.24; 10.56]	3.2 - [2.92; 3.48]	**2.9 - [2.75; 3.05]**	**2.9 - [2.70; 3.10]**	3.3 - [3.08; 3.52]	4.8 - [4.15; 5.45]	3.7 - [3.53; 3.87]
`logsigm`	7.8 - [5.73; 9.87]	3.4 - [3.23; 3.57]	5.7 - [4.35; 7.05]	5.9 - [3.98; 7.82]	**2.9 - [2.72; 3.08]**	**2.6 - [2.50; 2.70]**	3.6 - [3.06; 4.14]
Average no. of epochs	12	12	11	11	12	11	10
LM algorithm							
`logsig`	2.3 - [2.18; 2.42]	2.4 - [2.27; 2.53]	2.3 - [2.19; 2.41]	**2.0 - [1.99; 2.01]**	2.2 - [2.16; 2.24]	**2.0 - [1.98; 2.02]**	**2.0 - [1.99; 2.01]**
`tanh`	**2.1 - [2.08; 2.12]**	2.4 - [2.28; 2.52]	2.9 - [2.56; 3.24]	2.4 - [2.31; 2.49]	2.5 - [2.34; 2.66]	2.1 - [2.04; 2.16]	2.8 - [2.47; 3.13]
`cloglog`	2.1 - [2.06; 2.14]	2.7 - [2.44; 2.96]	2.2 - [2.15; 2.25]	2.8 - [2.35; 3.25]	2.1 - [2.06; 2.14]	2.8 - [2.64; 2.96]	2.1 - [2.07; 2.13]
`cloglogm`	2.2 - [2.14; 2.26]	2.2 - [2.15; 2.25]	2.7 - [2.48; 2.92]	2.1 - [2.06; 2.14]	2.4 - [2.31; 2.49]	2.2 - [2.15; 2.25]	2.4 - [2.19; 2.61]
`probit`	2.6 - [2.47; 2.73]	2.1 - [2.07; 2.13]	2.2 - [2.15; 2.25]	2.2 - [2.09; 2.31]	2.4 - [2.20; 2.60]	2.2 - [2.14; 2.26]	**2.0 - [1.99; 2.01]**
`loglog`	8.5 - [4.66; 12.34]	**2.1 - [2.08; 2.12]**	2.3 - [2.19; 2.41]	2.3 - [2.23; 2.37]	**2.1 - [2.07; 2.13]**	2.3 - [2.17; 2.43]	2.0 - [1.97; 2.03]
`sech`	2.6 - [2.39; 2.81]	4.7 - [3.73; 5.67]	2.9 - [2.57; 3.23]	3.3 - [2.72; 3.88]	2.4 - [2.32; 2.48]	2.8 - [2.42; 3.18]	3.7 - [3.19; 4.21]
`sinc`	10.5 - [7.52; 13.48]	9.3 - [7.47; 11.13]	3.9 - [3.32; 4.48]	4.1 - [3.71; 4.49]	2.6 - [2.47; 2.73]	4.1 - [3.52; 4.68]	3.2 - [2.84; 3.56]
`wave`	6.8 - [4.35; 9.25]	4.0 - [3.24; 4.76]	2.4 - [2.18; 2.62]	3.6 - [3.14; 4.06]	3.7 - [3.06; 4.34]	6.8 - [5.46; 8.14]	3.0 - [2.71; 3.29]
`sincos`	5.6 - [4.02; 7.18]	2.3 - [2.17; 2.43]	2.4 - [2.31; 2.49]	2.9 - [2.57; 3.23]	2.5 - [2.29; 2.71]	2.8 - [2.5; 3.1]	3.2 - [2.75; 3.65]
`rootsig`	4.1 - [2.83; 5.37]	**2.1 - [2.08; 2.12]**	**2.1 - [2.07; 2.13]**	2.2 - [2.12; 2.28]	2.2 - [2.15; 2.25]	2.2 - [2.11; 2.29]	2.3 - [2.15; 2.45]
`logsigm`	2.3 - [2.19; 2.41]	2.6 - [2.36; 2.84]	2.5 - [2.33; 2.67]	2.4 - [2.35; 2.45]	2.3 - [2.24; 2.36]	2.2 - [2.12; 2.28]	2.1 - [2.02; 2.18]
Average no. of epochs	16	18	11	17	10	12	9

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 12 Results of the average performance of the forecasting values in models of neural networks for the "HSBC" time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	25.1 - [21.28; 28.92]	24.1 - [19.63; 28.57]	17.6 - [14.14; 21.06]	14.6 - [10.77; 18.43]	12.3 - [9.19; 15.41]	22.8 - [18.38; 27.22]	22.5 - [18.60; 26.40]
`tanh`	6.5 - [4.71; 8.29]	13.8 - [8.81; 18.79]	10.0 - [7.69; 12.31]	6.2 - [4.48; 7.92]	7.6 - [6.19; 9.01]	7.4 - [5.14; 9.66]	18.6 - [12.87; 24.33]
`cloglog`	6.9 - [4.77; 9.03]	26.5 - [22.23; 30.77]	9.5 - [6.69; 12.31]	16.4 - [11.77; 21.03]	14.5 - [11.35; 17.65]	20.3 - [15.47; 25.13]	23.2 - [18.38; 28.02]
`cloglogm`	**3.2 - [3.09; 3.31]**	**4.9 - [3.85; 5.95]**	**5.7 - [4.67; 6.73]**	11.4 - [8.03; 14.77]	5.1 - [4.43; 5.77]	15.7 - [10.26; 21.14]	26.5 - [21.13; 31.87]
`probit`	9.0 - [6.13; 11.87]	26.4 - [22.14; 30.66]	13.4 - [9.89; 16.91]	**3.1 - [3.05; 3.15]**	**3.2 - [3.06; 3.34]**	15.6 - [10.98; 20.22]	15.1 - [10.55; 19.65]
`loglog`	9.7 - [7.48; 11.92]	12.7 - [9.35; 16.05]	14.9 - [11.66; 18.14]	14.7 - [10.82; 18.58]	3.3 - [3.16; 3.44]	14.8 - [10.31; 19.29]	12.9 - [8.81; 16.99]
`sech`	12.9 - [10.17; 15.63]	19.1 - [15.27; 22.93]	8.7 - [7.10; 10.30]	3.6 - [3.26; 3.94]	17.7 - [14.00; 21.40]	10.6 - [6.97; 14.23]	31.4 - [26.00; 36.80]
`sinc`	14.2 - [11.02; 17.38]	14.6 - [10.12; 19.08]	7.8 - [5.82; 9.78]	6.2 - [5.54; 6.86]	9.3 - [7.72; 10.88]	13.1 - [10.19; 16.01]	24.1 - [19.06; 29.14]
`wave`	7.4 - [5.46; 9.34]	22.3 - [16.47; 28.13]	7.3 - [5.54; 9.06]	5.2 - [4.44; 5.96]	16.8 - [10.40; 23.20]	7.3 - [6.24; 8.36]	**8.3 - [6.48; 10.12]**
`sincos`	6.3 - [4.49; 8.11]	21.1 - [16.05; 26.15]	16.0 - [11.61; 20.39]	21.2 - [15.95; 26.45]	14.3 - [12.00; 16.60]	40.5 - [31.32; 49.68]	20.7 - [15.35; 26.05]
`rootsig`	6.9 - [4.84; 8.96]	10.4 - [6.63; 14.17]	9.5 - [7.15; 11.85]	3.4 - [3.17; 3.63]	8.6 - [6.86; 10.34]	**3.5 - [3.25; 3.75]**	24.8 - [20.42; 29.18]
`logsigm`	12.4 - [8.59; 16.21]	13.7 - [10.50; 16.90]	9.2 - [6.91; 11.49]	10.3 - [7.09; 13.51]	3.5 - [3.10; 3.90]	11.2 - [7.68; 14.72]	10.2 - [7.53; 12.87]
Average no. of epochs	11	9	10	10	8	11	7
LM algorithm							
`logsig`	2.8 - [2.71; 2.89]	3.0 - [2.93; 3.07]	3.1 - [2.98; 3.22]	3.0 - [2.96; 3.04]	2.9 - [2.86; 2.94]	2.9 - [2.87; 2.93]	2.8 - [2.77; 2.83]
`tanh`	2.9 - [2.77; 3.03]	2.8 - [2.72; 2.88]	3.0 - [2.93; 3.07]	2.9 - [2.81; 2.99]	2.9 - [2.75; 3.05]	2.7 - [2.58; 2.82]	3.1 - [2.85; 3.35]
`cloglog`	3.1 - [2.97; 3.23]	3.2 - [3.08; 3.32]	3.3 - [3.14; 3.46]	3.2 - [3.14; 3.26]	2.8 - [2.77; 2.83]	2.9 - [2.86; 2.94]	2.9 - [2.84; 2.96]
`cloglogm`	3.3 - [3.11; 3.49]	3.0 - [2.93; 3.07]	3.0 - [2.93; 3.07]	3.0 - [2.92; 3.08]	2.7 - [2.62; 2.78]	2.9 - [2.81; 2.99]	2.7 - [2.62; 2.78]
`probit`	3.1 - [2.99; 3.21]	2.9 - [2.83; 2.97]	3.3 - [3.15; 3.45]	3.2 - [3.13; 3.27]	2.9 - [2.85; 2.95]	2.9 - [2.86; 2.94]	2.9 - [2.83; 2.97]
`loglog`	2.9 - [2.84; 2.96]	2.9 - [2.83; 2.97]	4.4 - [3.52; 5.28]	3.2 - [3.17; 3.23]	2.9 - [2.85; 2.95]	2.9 - [2.83; 2.97]	3.1 - [3.00; 3.20]
`sech`	3.1 - [2.94; 3.26]	4.4 - [3.45; 5.35]	3.6 - [3.22; 3.98]	4.1 - [3.74; 4.46]	3.1 - [2.87; 3.33]	2.9 - [2.78; 3.02]	2.7 - [2.54; 2.86]
`sinc`	4.9 - [3.97; 5.83]	3.2 - [3.08; 3.32]	3.6 - [3.37; 3.83]	3.6 - [3.30; 3.90]	3.7 - [3.32; 4.08]	5.2 - [3.92; 6.48]	3.2 - [2.97; 3.43]
`wave`	5.9 - [4.07; 7.73]	3.2 - [3.09; 3.31]	3.4 - [3.20; 3.60]	3.9 - [3.60; 4.20]	3.7 - [3.42; 3.98]	2.9 - [2.81; 2.99]	3.0 - [2.86; 3.14]
`sincos`	2.9 - [2.82; 2.98]	**2.6 - [2.55; 2.65]**	**2.6 - [2.55; 2.65]**	**2.8 - [2.62; 2.98]**	**2.5 - [2.44; 2.56]**	**2.5 - [2.44; 2.56]**	3.0 - [2.80; 3.20]
`rootsig`	**2.7 - [2.63; 2.77]**	3.3 - [3.07; 3.53]	3.2 - [3.08; 3.32]	3.1 - [3.01; 3.19]	3.0 - [2.96; 3.04]	3.0 - [2.91; 3.09]	2.8 - [2.74; 2.86]
`logsigm`	3.7 - [3.38; 4.02]	3.0 - [2.91; 3.09]	2.8 - [2.74; 2.86]	2.9 - [2.80; 3.00]	2.7 - [2.64; 2.76]	2.6 - [2.56; 2.64]	**2.7 - [2.65; 2.75]**
Average no. of epochs	23	19	10	9	8	8	8

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 13 Results of the average performance of the forecasting values in models of neural networks for the "TAM" time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	14.6 - [9.08; 20.12]	10.3 - [5.85; 14.75]	3.6 - [3.36; 3.84]	**3.0 - [2.94; 3.06]**	6.0 - [4.89; 7.11]	3.2 - [3.13; 3.27]	6.4 - [4.37; 8.43]
`tanh`	4.7 - [4.20; 5.20]	4.9 - [3.67; 6.13]	4.9 - [3.69; 6.11]	3.0 - [2.93; 3.07]	5.2 - [4.05; 6.35]	**3.1 - [2.97; 3.23]**	4.0 - [3.39; 4.61]
`cloglog`	6.0 - [4.27; 7.73]	6.1 - [4.90; 7.30]	**3.3 - [3.20; 3.40]**	4.1 - [3.59; 4.61]	4.3 - [4.03; 4.57]	3.7 - [3.51; 3.89]	5.9 - [4.45; 7.35]
`cloglogm`	**3.0 - [2.93; 3.07]**	8.9 - [6.05; 11.75]	4.2 - [3.42; 4.98]	3.7 - [3.23; 4.17]	3.5 - [3.16; 3.84]	4.4 - [3.86; 4.94]	4.6 - [3.73; 5.47]
`probit`	6.6 - [5.39; 7.81]	**3.4 - [3.23; 3.57]**	3.4 - [3.32; 3.48]	3.5 - [3.21; 3.79]	3.8 - [3.40; 4.20]	5.0 - [4.13; 5.87]	5.3 - [4.39; 6.21]
`loglog`	4.3 - [3.87; 4.73]	8.5 - [5.58; 11.42]	3.6 - [3.37; 3.83]	3.6 - [3.36; 3.84]	3.6 - [3.31; 3.89]	3.5 - [3.28; 3.72]	3.6 - [3.40; 3.80]
`sech`	3.9 - [3.52; 4.28]	3.5 - [3.35; 3.65]	3.8 - [3.43; 4.17]	3.3 - [3.13; 3.47]	3.9 - [3.66; 4.14]	4.0 - [3.68; 4.32]	4.1 - [3.50; 4.70]
`sinc`	28.1 - [21.67; 34.53]	3.6 - [3.31; 3.89]	4.6 - [4.25; 4.95]	5.7 - [5.27; 6.13]	5.1 - [4.50; 5.70]	4.4 - [4.11; 4.69]	4.2 - [3.92; 4.48]
`wave`	3.4 - [3.31; 3.49]	4.2 - [3.85; 4.55]	3.8 - [3.56; 4.04]	5.5 - [4.64; 6.36]	5.4 - [4.81; 5.99]	3.7 - [3.46; 3.94]	7.1 - [6.19; 8.01]
`sincos`	3.6 - [3.24; 3.96]	12.4 - [8.08; 16.72]	3.9 - [3.31; 4.49]	3.2 - [3.07; 3.33]	3.3 - [3.17; 3.43]	5.1 - [3.97; 6.23]	**3.5 - [3.32; 3.68]**
`rootsig`	3.6 - [3.30; 3.90]	4.8 - [4.24; 5.36]	3.7 - [3.38; 4.02]	3.5 - [3.28; 3.72]	5.5 - [4.33; 6.67]	3.6 - [3.24; 3.96]	3.7 - [3.33; 4.07]
`logsigm`	4.1 - [3.89; 4.31]	12.8 - [7.21; 18.39]	3.6 - [3.43; 3.77]	4.6 - [3.76; 5.44]	**3.1 - [3.02; 3.18]**	3.7 - [3.41; 3.99]	3.5 - [3.32; 3.68]
Average no. of epochs	13	13	13	12	10	8	9
LM algorithm							
`logsig`	3.0 - [2.96; 3.04]	2.9 - [2.86; 2.94]	2.9 - [2.88; 2.92]	**2.8 - [2.79; 2.81]**	**2.8 - [2.79; 2.81]**	2.9 - [2.88; 2.92]	2.9 - [2.88; 2.92]
`tanh`	2.9 - [2.88; 2.92]	2.9 - [2.84; 2.96]	3.0 - [2.96; 3.04]	2.9 - [2.86; 2.94]	2.9 - [2.87; 2.93]	2.9 - [2.88; 2.92]	3.2 - [3.12; 3.28]
`cloglog`	3.2 - [3.08; 3.32]	3.0 - [2.96; 3.04]	2.9 - [2.88; 2.92]	**2.8 - [2.79; 2.81]**	2.9 - [2.88; 2.92]	2.9 - [2.87; 2.93]	2.9 - [2.87; 2.93]
`cloglogm`	2.9 - [2.86; 2.94]	**2.8 - [2.79; 2.81]**	2.9 - [2.88; 2.92]	2.9 - [2.88; 2.92]	**2.8 - [2.79; 2.81]**	**2.8 - [2.79; 2.81]**	2.9 - [2.86; 2.94]
`probit`	**2.8 - [2.79; 2.81]**	2.9 - [2.86; 2.94]	2.9 - [2.88; 2.92]	2.9 - [2.89; 2.91]	2.9 - [2.88; 2.92]	2.9 - [2.88; 2.92]	2.9 - [2.88; 2.92]
`loglog`	2.9 - [2.88; 2.92]	2.9 - [2.84; 2.96]	2.9 - [2.88; 2.92]	2.8 - [2.78; 2.82]	2.9 - [2.89; 2.91]	2.9 - [2.88; 2.92]	2.9 - [2.88; 2.92]
`sech`	2.9 - [2.88; 2.92]	2.9 - [2.88; 2.92]	3.0 - [2.91; 3.09]	2.9 - [2.87; 2.93]	2.9 - [2.85; 2.95]	3.0 - [2.96; 3.04]	3.1 - [2.97; 3.23]
`sinc`	4.6 - [4.24; 4.96]	2.9 - [2.87; 2.93]	3.1 - [3.05; 3.15]	3.1 - [3.00; 3.20]	3.2 - [3.08; 3.32]	3.4 - [3.19; 3.61]	3.6 - [3.27; 3.93]
`wave`	3.0 - [2.87; 3.13]	3.0 - [2.94; 3.06]	3.5 - [3.30; 3.70]	3.6 - [3.41; 3.79]	3.9 - [3.57; 4.23]	3.6 - [3.47; 3.73]	3.6 - [3.42; 3.78]
`sincos`	**2.8 - [2.79; 2.81]**	8.9 - [5.12; 12.68]	**2.8 - [2.79; 2.81]**	2.9 - [2.85; 2.95]	2.8 - [2.78; 2.82]	2.9 - [2.88; 2.92]	**2.8 - [2.79; 2.81]**
`rootsig`	2.9 - [2.87; 2.93]	2.9 - [2.82; 2.98]	2.9 - [2.87; 2.93]	2.9 - [2.86; 2.94]	2.9 - [2.89; 2.91]	2.9 - [2.88; 2.92]	2.9 - [2.87; 2.93]
`logsigm`	2.9 - [2.84; 2.96]	3.0 - [2.94; 3.06]	2.9 - [2.89; 2.91]	2.9 - [2.86; 2.94]	2.9 - [2.87; 2.93]	2.9 - [2.88; 2.92]	**2.8 - [2.79; 2.81]**
Average no. of epochs	13	13	10	11	9	9	8

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 14 Results of the average performance of the forecasting values in models of neural networks for the "Itau" time series

Activation function	2 Nodes MAPE - [95% CI]	4 Nodes MAPE - [95% CI]	6 Nodes MAPE - [95% CI]	8 Nodes MAPE - [95% CI]	12 Nodes MAPE - [95% CI]	16 Nodes MAPE - [95% CI]	20 Nodes MAPE - [95% CI]
CGF algorithm							
`logsig`	12.5 - [12.01; 12.99]	10.2 - [9.70; 10.70]	8.5 - [7.72; 9.28]	9.3 - [8.57; 10.03]	8.4 - [7.82; 8.98]	6.6 - [5.96; 7.24]	9.9 - [8.86; 10.94]
`tanh`	8.3 - [7.91; 8.69]	8.5 - [7.58; 9.42]	8.2 - [6.83; 9.57]	8.7 - [7.70; 9.70]	8.8 - [7.85; 9.75]	9.2 - [7.64; 10.76]	8.6 - [7.83; 9.37]
`cloglog`	9.9 - [9.24; 10.56]	9.3 - [8.67; 9.93]	9.7 - [8.74; 10.66]	9.5 - [8.83; 10.17]	7.5 - [6.79; 8.21]	8.9 - [7.50; 10.30]	7.9 - [7.20; 8.60]
`cloglogm`	**6.0** - [**5.58**; **6.42**]	8.7 - [8.00; 9.40]	**5.7** - [**5.37**; **6.03**]	7.6 - [6.67; 8.53]	8.5 - [7.81; 9.19]	8.6 - [7.30; 9.90]	7.4 - [6.64; 8.16]
`probit`	9.1 - [8.29; 9.91]	9.1 - [8.74; 9.46]	7.9 - [6.94; 8.86]	9.7 - [8.86; 10.54]	7.2 - [5.77; 8.63]	7.5 - [7.15; 7.85]	8.9 - [7.98; 9.82]
`loglog`	10.6 - [10.02; 11.18]	9.2 - [8.63; 9.77]	8.8 - [8.06; 9.54]	10.3 - [9.00; 11.60]	**6.6** - [**6.43**; **6.77**]	7.9 - [6.81; 8.99]	7.4 - [6.69; 8.11]
`sech`	10.8 - [10.14; 11.46]	8.9 - [8.10; 9.70]	8.7 - [8.11; 9.29]	10.8 - [9.48; 12.12]	7.7 - [6.80; 8.60]	11.7 - [9.33; 14.07]	11.7 - [10.19; 13.21]
`sinc`	17.6 - [14.20; 21.00]	14.7 - [12.73; 16.67]	12.7 - [11.50; 13.90]	11.0 - [9.63; 12.37]	13.1 - [12.04; 14.16]	19.9 - [17.81; 21.99]	15.8 - [15.04; 16.56]
`wave`	8.7 - [7.95; 9.45]	9.0 - [7.89; 10.11]	7.7 - [7.17; 8.23]	9.3 - [8.28; 10.32]	8.3 - [7.57; 9.03]	9.1 - [8.05; 10.15]	8.8 - [7.92; 9.68]
`sincos`	8.1 - [7.66; 8.54]	**7.1** - [**6.59**; **7.61**]	6.1 - [5.82; 6.38]	**6.6** - [**6.15**; **7.05**]	6.9 - [6.32; 7.48]	14.9 - [11.11; 18.69]	8.5 - [7.93; 9.07]
`rootsig`	10.5 - [9.60; 11.40]	10.7 - [9.76; 11.64]	10.3 - [9.27; 11.33]	10.4 - [9.53; 11.27]	6.7 - [6.04; 7.36]	**5.4** - [**4.92**; **5.88**]	6.3 - [5.74; 6.86]
`logsigm`	9.1 - [8.46; 9.74]	10.6 - [9.93; 11.27]	7.8 - [7.01; 8.59]	8.4 - [7.62; 9.18]	6.9 - [6.27; 7.53]	7.5 - [7.04; 7.96]	**6.3** - [**5.96**; **6.64**]
Average no. of epochs	9	13	18	17	24	21	25
LM algorithm							
`logsig`	4.8 - [4.21; 5.39]	**3.8 - [3.79; 3.81]**	**3.8 - [3.79; 3.81]**	3.9 - [3.89; 3.91]	**3.8 - [3.79; 3.81]**	**3.8 - [3.79; 3.81]**	**3.9 - [3.88; 3.92]**
`tanh`	4.1 - [3.98; 4.22]	3.9 - [3.88; 3.92]	3.8 - [3.78; 3.82]	**3.8 - [3.79; 3.81]**	4.0 - [3.97; 4.03]	4.0 - [3.98; 4.02]	4.0 - [3.97; 4.03]
`cloglog`	**3.8 - [3.79; 3.81]**	3.8 - [3.78; 3.82]	3.9 - [3.87; 3.93]	**3.8 - [3.79; 3.81]**	**3.8 - [3.79; 3.81]**	3.9 - [3.87; 3.93]	**3.9 - [3.88; 3.92]**
`cloglogm`	3.9 - [3.86; 3.94]	4.0 - [3.94; 4.06]	3.9 - [3.88; 3.92]	3.9 - [3.89; 3.91]	3.9 - [3.85; 3.95]	3.9 - [3.88; 3.92]	4.0 - [3.97; 4.03]
`probit`	4.5 - [4.09; 4.91]	3.8 - [3.78; 3.82]	3.8 - [3.79; 3.81]	3.9 - [3.85; 3.95]	3.9 - [3.88; 3.92]	3.9 - [3.88; 3.92]	**3.9 - [3.88; 3.92]**
`loglog`	**3.8 - [3.79; 3.81]**	3.9 - [3.87; 3.93]	3.9 - [3.89; 3.91]	3.9 - [3.89; 3.91]	**3.8 - [3.79; 3.81]**	3.9 - [3.88; 3.92]	**3.9 - [3.88; 3.92]**
`sech`	3.9 - [3.87; 3.93]	4.0 - [3.97; 4.03]	3.9 - [3.87; 3.93]	3.9 - [3.89; 3.91]	4.0 - [3.97; 4.03]	4.0 - [3.98; 4.02]	4.2 - [4.11; 4.29]
`sinc`	7.0 - [6.56; 7.44]	6.0 - [5.59; 6.41]	4.9 - [4.63; 5.17]	5.7 - [5.30; 6.10]	4.5 - [4.30; 4.70]	4.2 - [4.16; 4.24]	4.1 - [4.07; 4.13]
`wave`	5.2 - [4.69; 5.71]	3.9 - [3.88; 3.92]	4.0 - [3.97; 4.03]	4.1 - [4.05; 4.15]	4.5 - [4.31; 4.69]	4.2 - [4.16; 4.24]	4.2 - [4.17; 4.23]
`sincos`	4.0 - [3.95; 4.05]	4.1 - [4.03; 4.17]	4.0 - [3.93; 4.07]	4.0 - [3.95; 4.05]	4.2 - [4.07; 4.33]	4.0 - [3.99; 4.01]	4.0 - [3.98; 4.02]
`rootsig`	3.9 - [3.88; 3.92]	3.8 - [3.79; 3.81]	4.0 - [3.97; 4.03]	3.9 - [3.89; 3.91]	3.9 - [3.88; 3.92]	3.9 - [3.88; 3.92]	**3.9 - [3.88; 3.92]**
`logsigm`	**3.8 - [3.79; 3.81]**	3.8 - [3.79; 3.81]	3.9 - [3.89; 3.91]	3.9 - [3.89; 3.91]	3.9 - [3.88; 3.92]	**3.8 - [3.78; 3.82]**	**3.9 - [3.88; 3.92]**
Average no. of epochs	17	15	43	23	25	16	11

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 15 Results of the average performance of the forecasting values in models of neural networks for the "Brasil Telecom" time series

Activation function	2 Nodes $\overline{\text{MAPE}}$ - [95% CI]	4 Nodes $\overline{\text{MAPE}}$ - [95% CI]	6 Nodes $\overline{\text{MAPE}}$ - [95% CI]	8 Nodes $\overline{\text{MAPE}}$ - [95% CI]	12 Nodes $\overline{\text{MAPE}}$ - [95% CI]	16 Nodes $\overline{\text{MAPE}}$ - [95% CI]	20 Nodes $\overline{\text{MAPE}}$ - [95% CI]
CGF algorithm							
`logsig`	6.6 - [5.65; 7.55]	5.5 - [4.59; 6.41]	6.2 - [5.21; 7.19]	3.4 - [3.19; 3.61]	4.9 - [4.03; 5.77]	4.2 - [3.50; 4.90]	3.9 - [3.40; 4.40]
`tanh`	6.2 - [5.36; 7.04]	4.2 - [3.77; 4.63]	**3.0 - [2.92; 3.08]**	3.8 - [3.38; 4.22]	3.2 - [3.12; 3.28]	3.7 - [3.42; 3.98]	**3.0 - [2.93; 3.07]**
`cloglog`	7.5 - [6.32; 8.68]	5.5 - [4.71; 6.29]	4.3 - [3.68; 4.92]	3.4 - [3.21; 3.59]	**2.9 - [2.86; 2.94]**	3.2 - [3.12; 3.28]	4.1 - [3.53; 4.67]
`cloglogm`	**4.4 - [3.97; 4.83]**	**3.0 - [2.97; 3.03]**	3.1 - [3.04; 3.16]	3.1 - [3.04; 3.16]	3.0 - [2.96; 3.04]	3.4 - [3.24; 3.56]	3.7 - [3.36; 4.04]
`probit`	6.1 - [5.09; 7.11]	4.7 - [4.08; 5.32]	4.0 - [3.44; 4.56]	3.1 - [3.03; 3.17]	3.0 - [2.95; 3.05]	4.7 - [3.99; 5.41]	4.2 - [3.57; 4.83]
`loglog`	5.3 - [4.42; 6.18]	5.0 - [3.74; 6.26]	4.3 - [3.66; 4.94]	3.1 - [3.02; 3.18]	3.0 - [2.96; 3.04]	**2.9 - [2.88; 2.92]**	3.5 - [3.16; 3.84]
`sech`	8.1 - [7.07; 9.13]	5.1 - [4.25; 5.95]	5.4 - [4.33; 6.47]	5.7 - [4.87; 6.53]	3.6 - [3.26; 3.94]	4.5 - [3.81; 5.19]	4.0 - [3.56; 4.44]
`sinc`	5.7 - [5.20; 6.20]	5.0 - [4.38; 5.62]	5.0 - [4.45; 5.55]	5.3 - [4.76; 5.84]	4.3 - [3.95; 4.65]	4.2 - [3.73; 4.67]	4.7 - [4.13; 5.27]
`wave`	7.5 - [5.92; 9.08]	4.2 - [3.45; 4.95]	3.7 - [3.43; 3.97]	4.6 - [3.98; 5.22]	3.9 - [3.61; 4.19]	8.6 - [7.12; 10.08]	5.0 - [4.11; 5.89]
`sincos`	6.2 - [5.11; 7.29]	3.4 - [3.32; 3.48]	3.5 - [3.30; 3.70]	3.7 - [3.47; 3.93]	3.3 - [3.24; 3.36]	4.0 - [3.73; 4.27]	3.1 - [2.97; 3.23]
`rootsig`	5.6 - [4.57; 6.63]	4.8 - [3.75; 5.85]	3.1 - [3.06; 3.14]	3.0 - [2.94; 3.06]	3.2 - [3.13; 3.27]	3.0 - [2.96; 3.04]	3.1 - [3.01; 3.19]
`logsigm`	7.0 - [5.94; 8.06]	5.9 - [4.75; 7.05]	4.7 - [3.82; 5.58]	**3.0 - [2.97; 3.03]**	3.0 - [2.98; 3.02]	3.4 - [3.11; 3.69]	3.0 - [2.96; 3.04]
Average no. of epochs	12	10	10	9	8	8	9
LM algorithm							
`logsig`	2.9 - [2.88; 2.92]	**2.8 - [2.80; 2.80]**	**2.8 - [2.80; 2.80]**	3.0 - [2.90; 3.10]	**2.8 - [2.80; 2.80]**	**2.9 - [2.90; 2.90]**	**2.8 - [2.80; 2.80]**
`tanh`	3.5 - [3.12; 3.88]	2.9 - [2.90; 2.90]	2.9 - [2.90; 2.90]	2.9 - [2.87; 2.93]	2.9 - [2.90; 2.90]	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]
`cloglog`	4.1 - [3.32; 4.88]	2.9 - [2.90; 2.90]	2.9 - [2.90; 2.90]	**2.9 - [2.90; 2.90]**	2.9 - [2.87; 2.93]	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]
`cloglogm`	3.6 - [3.25; 3.95]	2.9 - [2.90; 2.90]	**2.8 - [2.80; 2.80]**	**2.9 - [2.90; 2.90]**	3.0 - [2.93; 3.07]	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]
`probit`	4.0 - [3.29; 4.71]	2.9 - [2.89; 2.91]	3.1 - [2.93; 3.27]	2.9 - [2.88; 2.92]	2.9 - [2.88; 2.92]	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]
`loglog`	3.0 - [2.92; 3.08]	2.9 - [2.90; 2.90]	2.9 - [2.90; 2.90]	2.9 - [2.88; 2.92]	**2.8 - [2.80; 2.80]**	**2.9 - [2.90; 2.90]**	3.0 - [2.90; 3.10]
`sech`	3.8 - [3.33; 4.27]	2.9 - [2.89; 2.91]	2.9 - [2.90; 2.90]	2.9 - [2.89; 2.91]	2.9 - [2.90; 2.90]	**2.9 - [2.90; 2.90]**	2.9 - [2.86; 2.94]
`sinc`	5.8 - [4.72; 6.88]	2.9 - [2.88; 2.92]	2.9 - [2.89; 2.91]	2.9 - [2.89; 2.91]	3.1 - [2.99; 3.21]	**2.9 - [2.89; 2.91]**	3.2 - [3.07; 3.33]
`wave`	2.9 - [2.87; 2.93]	2.9 - [2.89; 2.91]	2.9 - [2.89; 2.91]	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]	3.0 - [2.95; 3.05]	2.9 - [2.87; 2.93]
`sincos`	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]	2.9 - [2.90; 2.90]	2.9 - [2.87; 2.93]	2.9 - [2.89; 2.91]	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]
`rootsig`	3.7 - [3.15; 4.25]	2.9 - [2.89; 2.91]	2.9 - [2.90; 2.90]	2.9 - [2.87; 2.93]	2.9 - [2.90; 2.90]	**2.9 - [2.90; 2.90]**	2.9 - [2.90; 2.90]
`logsigm`	3.9 - [3.22; 4.58]	**2.8 - [2.80; 2.80]**	**2.8 - [2.80; 2.80]**	**2.9 - [2.90; 2.90]**	**2.8 - [2.80; 2.80]**	**2.9 - [2.90; 2.90]**	3.0 - [2.92; 3.08]
Average no. of epochs	29	33	51	33	49	39	35

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

Table 16 Results of the average performance of the forecasting values in models of neural networks for the "Petrobras" time series

Activation function	2 Nodes MAPE - [95% CI]	4 Nodes MAPE - [95% CI]	6 Nodes MAPE - [95% CI]	8 Nodes MAPE - [95% CI]	12 Nodes MAPE - [95% CI]	16 Nodes MAPE - [95% CI]	20 Nodes MAPE - [95% CI]
CGF algorithm							
`logsig`	4.0 - [3.34; 4.66]	3.7 - [3.44; 3.96]	8.6 - [5.41; 11.79]	15.1 - [11.95; 18.25]	11.0 - [7.31; 14.69]	17.8 - [11.69; 23.91]	3.5 - [3.16; 3.84]
`tanh`	4.7 - [3.93; 5.47]	3.2 - [2.75; 3.65]	**2.4 - [2.37; 2.43]**	2.6 - [2.52; 2.68]	**2.5 - [2.46; 2.54]**	3.4 - [2.90; 3.90]	2.6 - [2.49; 2.71]
`cloglog`	6.5 - [5.54; 7.46]	10.5 - [7.72; 13.28]	9.6 - [7.76; 11.44]	17.2 - [13.88; 20.52]	24.3 - [18.15; 30.45]	20.6 - [13.25; 27.95]	27.1 - [16.94; 37.26]
`cloglogm`	**2.7 - [2.45; 2.95]**	2.7 - [2.60; 2.80]	2.5 - [2.44; 2.56]	**2.4 - [2.34; 2.46]**	**2.5 - [2.46; 2.54]**	5.1 - [4.29; 5.91]	2.6 - [2.49; 2.71]
`probit`	6.5 - [5.11; 7.89]	3.5 - [3.19; 3.81]	3.3 - [3.06; 3.54]	3.1 - [2.96; 3.24]	2.9 - [2.78; 3.02]	3.1 - [3.02; 3.18]	3.1 - [2.98; 3.22]
`loglog`	4.9 - [4.21; 5.59]	3.3 - [3.07; 3.53]	3.0 - [2.83; 3.17]	2.9 - [2.77; 3.03]	7.7 - [5.35; 10.05]	2.7 - [2.61; 2.79]	2.6 - [2.49; 2.71]
`sech`	5.8 - [5.07; 6.53]	2.9 - [2.75; 3.05]	3.5 - [3.12; 3.88]	6.7 - [4.34; 9.06]	3.1 - [2.96; 3.24]	3.6 - [3.33; 3.87]	4.6 - [4.10; 5.10]
`sinc`	9.0 - [7.39; 10.61]	3.5 - [3.37; 3.63]	8.1 - [6.82; 9.38]	5.2 - [4.42; 5.98]	8.3 - [6.53; 10.07]	6.7 - [5.67; 7.73]	8.9 - [8.22; 9.58]
`wave`	6.0 - [5.30; 6.70]	3.5 - [3.36; 3.64]	3.8 - [3.52; 4.08]	3.5 - [3.24; 3.76]	6.4 - [5.26; 7.54]	8.1 - [6.87; 9.33]	12.0 - [9.23; 14.77]
`sincos`	4.4 - [3.94; 4.86]	5.8 - [4.88; 6.72]	6.2 - [5.16; 7.24]	9.4 - [7.00; 11.80]	4.0 - [3.41; 4.59]	3.9 - [3.45; 4.35]	6.3 - [5.58; 7.02]
`rootsig`	3.7 - [3.45; 3.95]	3.9 - [3.62; 4.18]	2.8 - [2.64; 2.96]	2.7 - [2.63; 2.77]	2.5 - [2.35; 2.65]	**2.6 - [2.55; 2.65]**	**2.5 - [2.44; 2.56]**
`logsigm`	3.5 - [3.05; 3.95]	**2.7 - [2.63; 2.77]**	2.7 - [2.60; 2.80]	3.0 - [2.79; 3.21]	6.9 - [4.49; 9.31]	3.7 - [3.47; 3.93]	4.1 - [3.96; 4.24]
Average no. of epochs	13	13	12	11	11	10	10
LM algorithm							
`logsig`	**2.2 - [2.19; 2.21]**	**2.2 - [2.19; 2.21]**	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]
`tanh`	**2.2 - [2.19; 2.21]**	**2.2 - [2.19; 2.21]**	2.2 - [2.19; 2.21]	2.3 - [2.23; 2.37]	**2.1 - [2.10; 2.10]**	2.1 - [2.09; 2.11]	**2.1 - [2.10; 2.10]**
`cloglog`	**2.2 - [2.19; 2.21]**	**2.2 - [2.19; 2.21]**	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.1 - [2.09; 2.11]	2.2 - [2.19; 2.21]	**2.1 - [2.10; 2.10]**
`cloglogm`	**2.2 - [2.19; 2.21]**	2.2 - [2.18; 2.22]	**2.1 - [2.10; 2.10]**	2.2 - [2.18; 2.22]	2.2 - [2.19; 2.21]	2.1 - [2.09; 2.11]	**2.1 - [2.10; 2.10]**
`probit`	**2.2 - [2.19; 2.21]**	**2.2 - [2.19; 2.21]**	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]
`loglog`	**2.2 - [2.19; 2.21]**	**2.2 - [2.19; 2.21]**	2.2 - [2.18; 2.22]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.20; 2.20]	2.2 - [2.19; 2.21]
`sech`	2.3 - [2.26; 2.34]	**2.2 - [2.19; 2.21]**	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.1 - [2.09; 2.11]	2.2 - [2.18; 2.22]	2.2 - [2.18; 2.22]
`sinc`	8.5 - [6.58; 10.42]	3.1 - [2.71; 3.49]	2.6 - [2.45; 2.75]	2.2 - [2.16; 2.24]	2.2 - [2.18; 2.22]	**2.1 - [2.10; 2.10]**	2.2 - [2.19; 2.21]
`wave`	4.5 - [3.05; 5.95]	2.3 - [2.22; 2.38]	2.2 - [2.19; 2.21]	2.2 - [2.17; 2.23]	2.2 - [2.19; 2.21]	2.2 - [2.20; 2.20]	2.2 - [2.19; 2.21]
`sincos`	**2.2 - [2.19; 2.21]**	**2.2 - [2.19; 2.21]**	2.2 - [2.18; 2.22]	**2.1 - [2.09; 2.11]**	2.1 - [2.09; 2.11]	2.2 - [2.19; 2.21]	2.1 - [2.09; 2.11]
`rootsig`	**2.2 - [2.19; 2.21]**	2.2 - [2.18; 2.22]	2.2 - [2.19; 2.21]	2.2 - [2.20; 2.20]	2.2 - [2.18; 2.22]	**2.1 - [2.10; 2.10]**	**2.1 - [2.10; 2.10]**
`logsigm`	**2.2 - [2.19; 2.21]**	**2.2 - [2.19; 2.21]**	2.2 - [2.19; 2.21]	2.2 - [2.18; 2.22]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]	2.2 - [2.19; 2.21]
Average no. of epochs	15	21	16	15	14	13	17

CGF conjugate gradient backpropagation with Fletcher–Reeves updates, *LM* Levenberg–Marquadt, *CI* confidence interval

The "HSBC" stock market series, presents a very high degree of negative asymmetry (skewness = −2.31), and the results of the performance for this series are presented in Table 12. In many situations, the use of the CGF algorithm did not improve the results compared with the results obtained by the AR model. However, with an architecture of six hidden nodes or less, the best results were obtained by the model with cloglogm activation function, using 8 and 12 hidden nodes, the best performance observed was by the model with probit activation function. The 95% CI for the $\overline{MAPE}$ corroborates with these facts. Using the LM algorithm, the best results were verified for the models with rootsig, sincos and logsigm activation functions. However, the results for the other models were very close to these.

The "TAM" and "Itau" stock market series presented a small degree of positive asymmetry, 0.21 and 0.34 respectively (Table 3). The results of the performance for these time series are presented in Tables 13 and 14. For the "TAM" time series, just the result from the model with sinc activation function and two hidden nodes, using the CGF algorithm presented a higher average performance than the AR model (Table 4). For the "Itau" series, most results using the same algorithm were higher than the AR model. In Table 13, we can observe that for networks with smaller structure (six hidden nodes or less), the best results were obtained by the models with cloglogm, probit and cloglog activation functions, using the algorithm CGF. Using the LM algorithm, the results are equivalent and very close to the results found with the model that used the cloglogm activation function with two hidden nodes using the algorithm CGF. In Table 14, we can see that the model with cloglogm activation function, with two and six hidden nodes, using the CGF algorithm, is still the one with better performance. Using the LM algorithm, the best results among the networks with just two hidden nodes are obtained by the models with cloglog, cloglogm and logsigm activation functions. Other results are similar with increasing numbers of hidden nodes in the neural network.

The "Brasil Telecom" and "Petrobras" stock market series presented a higher degree of positive asymmetry, 1.49 and 1.75, respectively. The results for these series are presented in Tables 15 and 16. For the "Brasil Telecom" time series, the model with cloglogm activation function was the only one that presented better results than the AR model for all the configurations of networks used. For the "Petrobras" time series, in some situations, the results obtained by the AR model was better than the results obtained by the models with logsig, clolog, loglog, sinc, wave, sincos and logitm activation functions. In Table 15, we can see that the model with cloglogm activation function presented better results with smaller networks. Using the CGF algorithm, the model with tanh activation function also presented good results with six hidden nodes and the models with cloglog and loglog activation functions presented the best results for networks with 12 and 16 hidden nodes. Using the LM algorithm, the results were equivalent to the ones described previously. In Table 16, we can observe that, using the CGF algorithm, the model with cloglogm activation function presented the best results through networks with two hidden nodes, the model with logsigm activation function also presented good results with four hidden nodes and the model with tanh activation function, presented good results with networks of six hidden nodes. Using the LM algorithm, the results obtained were equivalent to the ones obtained with the CGF algorithm.

4 Conclusions

This paper presents a comparison between the average performance of neural networks models with different activation functions and two different learning algorithms. Besides, it compares these results with the results obtained by AR models. To validate the results we executed exhaustive experiments. In general, the average performances obtained by the neural networks models are better than those obtained by the AR models. However, the models in which the performance was inferior, some activation functions were observed more constantly, which is the case of the following functions: sech, sinc, wave and sincos.

The first conclusion we can take from this study is that, using CGF algorithm, the activation functions proposed in this study, i.e., cloglog, cloglogm, probit and loglog, obtained the best average performance with smaller networks, $q = \{2, 4, 6\}$. We can also conclude that the models with the cloglog and cloglogm activation functions improved significantly their performances, especially with the series known to be asymmetric and through simple descriptive measures we could identify such characteristic. The models with logsig, tanh and rootsig activation functions showed better performance with larger networks, $q = \{8, 12, 16, 20\}$.

The second conclusion is that, using LM algorithm, the models with logsig activation function reached better results due to the use of this particular algorithm. The proposed activation functions also showed good performance. However, it is important to say that these good results had already been observed when using the CGF algorithm. The importance of this fact is that the LM algorithm is more expensive than the CGF algorithm. Moreover, the good performance of the models with the new activation functions are not as much influenced by changing of the training algorithm as the other functions in literature. This fact was specially noticed in the larger bases used in this study. The fact is that having more examples (patterns) available is enough for obtaining the desired performance through LM algorithm.

Therefore, in this paper, from these results, we recommend using the new activation functions, `cloglog`, `cloglogm`, `probit` and `loglog`, especially for financial time series and networks with small structure, if this is an important factor to be considered by the specialist. By identifying the type of asymmetry, through descriptive statistics of time series, we can choose the activation function suitable for every situation. For symmetrical data, we recommend that the activation function `probit` and for asymmetric data the activation function `cloglogm`. We also strongly recommend the use of these functions in the models of neural networks that use the CGF algorithm, due to the fact that the LM algorithm is computationally too expensive, requires more memory space and requires great programming skills from the specialist for its correct implementation.

Acknowledgments The authors would like to thank CNPq and FACEPE for their financial support. The authors gratefully acknowledge the contribution of the anonymous referees comments in improving the clarity of the work.

References

1. Box GEP, Jenkins GM (1976) Time series analysis, forecasting and control. Holden Day, San Francisco
2. Engle RF (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. Econometrica 50:987–1008
3. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2:303–314
4. White H (1990) Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings. Neural Net 3:535–550
5. Gallant AR, White H (1992) On learning the derivatives of an unknown mapping with multilayer feedforward networks. Neural Net 5:129–138
6. Huang W, Lai K, Nakamori Y, Wang S, Yu L (2007) Neural networks in finance and economics forecasting. Int J Inf Technol Decision Making 6(1):113–140
7. Bollerslev T (1990) Modeling the coherence in short-run nominal exchange rates: a multivariate generalized arch model. Rev Econ Statist 72:498–505
8. Chandra P, Singh Y (2004) An activation function adapting training algorithm for sigmoidal feedforward networks. Neurocomputing 61:429–437
9. Duch W, Jankowski N (1999) Survey of neural transfer functions. Neural Comput Appl 2:163–212
10. Duch W, Jankowski N (2001) Transfer functions: hidden possibilities for better neural networks. In 9th European symposium on artificial neural networks, pp 81–94
11. Singh Y and Chandra P (2003) A class +1 sigmoidal activation functions for FFANNs. J Econ Dynamic Control 28(1):183–187
12. Pao YH (1989) Adaptive pattern recognition and neural networks, 2nd edn. Addison-Wesley, New York
13. Hartman E, Keeler JD, Kowalski JM (1990) Layered neural networks with gaussian hidden units as universal approximations. Neural Comput Appl 2(2):210–215
14. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Net 4(2):251–257
15. Hornik K (1993) Some new results on neural network approximation. Neural Net 6(9):1069–1072
16. Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Net 6(6):861–867
17. Leung H, Haykin S (1993) Rational function neural network. Neural Comput Appl 5(6):928–938
18. Giraud B, Lapedes A, Lon C, Lemm J (1995) Lorentzian neural nets. Neural Net 8(5):757–767
19. Skoundrianos EN, Tzafestas SG (2004) Modelling and FDI of dynamic discrete time systems using a MLP with a new sigmoidal activation function. J Intell Robotics Syst 41(1):19–36
20. Ma L, Khorasani K (2005) Constructive feedforward neural networks using hermite polynomial activation functions. IEEE Trans Neural Net 16(4):821–833
21. Wen C, Ma X (2005) A max-piecewise-linear neural network for function approximation. Neurocomputing 71:843–852
22. Efe MO (2008) Novel neuronal activation functions for feedforward neural networks. Neural Process Lett 28:63–79
23. Gomes GSS, Ludermir TB (2008) Complementary log-log and probit: activation functions implemented in artificial neural networks. In: 8th International conference on hybrid intelligent systems. IEEE Computer Society, pp 939–942
24. Fletcher R, Reeves CM (1964) Function minimization by conjugate gradients. Comput J 7:149–154
25. Hagan MT, Demuth HB, Beale MH (1996) Neural network design. PWS Publishing, Boston
26. Hagan MT, Menhaj M (1994) Training feed-forward networks with the marquardt algorithm. IEEE Trans Neural Net 5(6):989–993
27. Marquardt D (1963) An algorithm for least-squares estimation of nonlinear parameters. SIAM J Appl Math 11: 431–441
28. Cybenko G (1988) Continuous valued neural networks with two hidden layers are sufficient technical report, Department of Computer Science, Tufts University, Medford, MA
29. Hara K, Nakayamma K (1994) Comparison of activation functions in multilayer neural network for pattern classification. In: International conference on neural networks vol 5, IEEE World Congress on computational intelligence, pp 2997–3002
30. Nelder JA, Wedderburn WM (1972) Generalized linear models. J Royal Statist Soc 3:370–384
31. McCullagh P, Nelder JA (1989) Generalized linear models, 2nd edn. Chapman and Hall, London
32. Bliss CI (1935) The calculation of the dosage-mortality curve. Ann Appl Biol 22:134–167
33. Collett D (1994) Modelling survival data in medical research. Chapman and Hall, London
34. Dobson AJ (2002) An introduction to generalized linear models, 2nd edn. Chapman and Hall, New York
35. McCullagh P, Nelder JA (1989) Generalized linear models, 2nd ed. Chapman and Hall, London
36. Haykin S (2001) Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall, New Jersey