Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 152375, 12 pages
http://dx.doi.org/10.1155/2014/152375

Research Article

Smoothing Strategies Combined with ARIMA and Neural Networks to Improve the Forecasting of Traffic Accidents

Lida Barba,[1,2] Nibaldo Rodríguez,[1] and Cecilia Montt[1]

[1] *Pontificia Universidad Católica de Valparaíso, 2362807 Valparaíso, Chile*
[2] *Universidad Nacional de Chimborazo, 33730880 Riobamba, Ecuador*

Correspondence should be addressed to Lida Barba; lida_barba@hotmail.com

Received 26 April 2014; Revised 29 July 2014; Accepted 14 August 2014; Published 28 August 2014

Academic Editor: Cagdas Hakan Aladag

Two smoothing strategies combined with autoregressive integrated moving average (ARIMA) and autoregressive neural networks (ANNs) models to improve the forecasting of time series are presented. The strategy of forecasting is implemented using two stages. In the first stage the time series is smoothed using either, 3-point moving average smoothing, or singular value Decomposition of the Hankel matrix (HSVD). In the second stage, an ARIMA model and two ANNs for one-step-ahead time series forecasting are used. The coefficients of the first ANN are estimated through the particle swarm optimization (PSO) learning algorithm, while the coefficients of the second ANN are estimated with the resilient backpropagation (RPROP) learning algorithm. The proposed models are evaluated using a weekly time series of traffic accidents of Valparaíso, Chilean region, from 2003 to 2012. The best result is given by the combination HSVD-ARIMA, with a MAPE of $0:26\%$, followed by MA-ARIMA with a MAPE of $1:12\%$; the worst result is given by the MA-ANN based on PSO with a MAPE of $15:51\%$.

1. Introduction

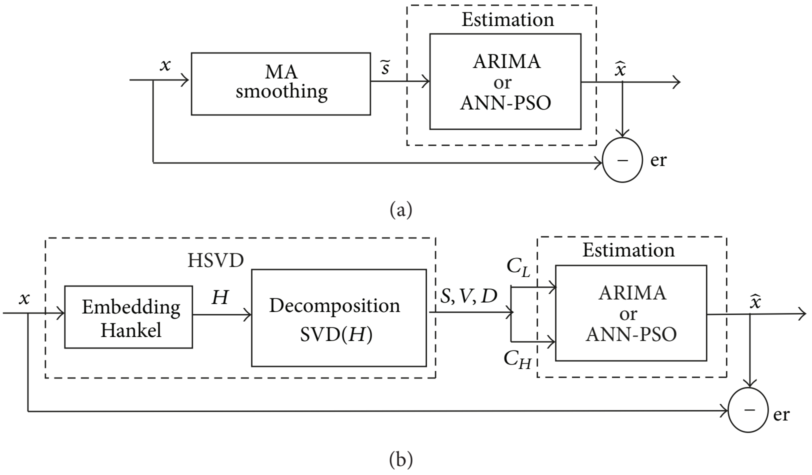
The traffic accidents occurrence is a matter of impact in the society, therefore a problem of priority public attention; the Chilean National Traffic Safety Commission (CONASET) periodically reports a high rate of sinister on roads; in Valparaíso from year 2003 to 2012 28595 injured people were registered. The accuracy in the projections enables the intervention by the government agencies in terms of prevention; another demandant of information is the insurance companies, who require this kind of information to determine new market policies.

In order to capture the dynamic of traffic accidents, during the last years some techniques have been applied. For classification, decision rules and trees [1, 2], latent class clustering and bayesian networks [3], and the genetic algorithm [4] have been implemented. For traffic accidents forecasting, autoregressive moving average (ARMA) and ARIMA models [5], state-space models [6, 7], extrapolation [8], dynamic harmonic regression combined with ARIMA, and dynamic transfer functions [9] have been implemented.

The smoothing strategies Moving Average (MA) and Singular Value Decomposition (SVD) have been used to identify the components in a time series. MA is used to extract the trend [10], while SVD extracts more components [11]; the SVD application is multivariate and in some works is applied for parameter calibration in dynamical systems [12, 13], in time series classification [14], or to switched linear systems [15]; typically SVD has been applied over an input data set to reduce the data dimensionality [16] or to noise reduction [17].

ARIMA is a linear conventional model for nonstationary time series; by differentiation the nonstationary time series is transformed in stationary; it is based on past values of the series and on the previous error terms for forecasting. ARIMA has been applied widely to model nonstationary data; some applications are the traffic noise [18], the daily global solar radiation [19], premonsoon rainfall data for western India [20], and aerosols over the Gangetic Himalayan region [21].

The autoregressive neural network (ANN) is a nonlinear method for forecasting that has been shown to be efficient in



(a)

(b)

Figure 1: Smoothing strategies: (a) moving average and (b) Hankel singular value decomposition.

solving problems of different fields; the capability of learning of the ANN is determined by the algorithm. Particle swarm optimization (PSO) is a population algorithm that has been found to be optimal; it is based on the behaviour of a swarm; this is applied to update the connections weights of the ANN; some modifications of PSO have been evaluated based on variants of the acceleration coefficients [22], others apply the adaptation of the inertia weight [23–26], also the usage of adaptive mechanisms for both inertia weight and the acceleration coefficients based on the behaviour of the particle at each iteration have been used [27, 28]. The combination of ANN-PSO has improved the forecasting over some classical algorithms like backpropagation (BP) [29–31] and least mean square (LMS) [32]. Another learning algorithm that has been shown to be better than backpropagation is RPROP and is also analyzed by its robustness, easy implementation, and fast convergence regarding the conventional BP [33, 34].

The linear and nonlinear models may be inadequate in some forecasting problems; consequently they are not considered universal models; then the combination of linear and nonlinear models could capture different forms of relationships in the time series data. The Zhang hybrid methodology that combines both ARIMA and ANN models is an effective way to improve forecasting accuracy; ARIMA model is used to analyze the linear part of the problem and the ANN models, the residuals from the ARIMA model [35]; this model has been applied for demand forecasting [36]; however some researchers believe that some assumptions of Zhang can degenerate hybrid methodology when opposite situation occurs; Kashei proposes a methodology that combines the linear and nonlinear models which has no assumptions of traditional Zhang hybrid linear and nonlinear models in order to yield the more general and the more accurate forecasting model [37].

Based on the arguments presented in this work, two smoothing strategies to potentiate the preprocessing stage of time series forecasting are proposed; 3-point MA and HSVD are used to smooth the time series; the smoothed values are forecasted with three models; the first is based on ARIMA model, the second in ANN is based on PSO, and the third in ANN is based on RPROP. The models are evaluated using the time series of injured people in traffic accidents occurring in Valparaíso, Chilean region, from 2003 to 2012 with 531 weekly registers. The smoothing strategies and the forecasting models are combined and six models are obtained and compared to determine the model that gives the major accuracy. The paper is structured as follows. Section 2 describes the smoothing strategies. Section 3 explains the proposed forecasting models. Section 4 presents the forecasting accuracy metrics. Section 5 presents the results and discussions. The conclusions are shown in Section 6.

2. Smoothing Strategies

2.1. Moving Average. Moving average is a smoothing strategy used in linear filtering to identify or extract the trend from a time series. MA is a mean of a constant number of observations that can be used to describe a series that does not exhibit a trend [38]. When 3-point MA is applied over a time series of length n, the $n-2$ elements of the smoothed series are computed with

$$\tilde{s}_k = \sum_{i=k-1}^{k+1} \frac{x_i}{3}, \quad (1)$$

where $\tilde{s}_k$ is the kth smoothed signal element, for $k = 2, \ldots, n-1, x_i$ is each observed element of original time series, and terms $\tilde{s}_1$ and $\tilde{s}_n$ have the same values of x_1 and x_n, respectively. The smoothed values given by 3-points MA will be used by the estimation process through the selected technique (ARIMA or ANN); this strategy is illustrated in Figure 1(a).

2.2. Hankel Singular Value Decomposition. The proposed strategy HSVD is implemented during the preprocessing stage in two steps, embedding and decomposition. The time series is embedded in a trajectory matrix; then the structure of the Hankel matrix is applied, the decomposition process extracts the components of low and high frequency of the mentioned matrix by means of SVD, the smoothed values given by HSVD are used by the estimation process, and this strategy is illustrated in Figure 1(b).

The original time series is represented with x, H is the Hankel matrix, U, S, V are the elements obtained with SVD and will be detailed more ahead, C_L is the component of low frequency, C_H is the component of high frequency, $\hat{x}$ is the forecasted time series, and er is the error computed between x and $\hat{x}$ with

$$\widehat{er} = x - \hat{x}. \tag{2}$$

2.2.1. Embedding the Time Series. The embedding process is illustrated as follows:

$$H_{M\times L} = \begin{bmatrix} x_1 & x_2 & \cdots & x_L \\ x_2 & x_3 & \cdots & x_{L+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_M & x_{M+1} & \cdots & x_n \end{bmatrix}, \tag{3}$$

where H is a real matrix, whose structure is the Hankel matrix, $x_1, \ldots, x_n$ are the original values of the time series, M is the number of rows of H and also M is the number of components that will be obtained with SVD, L is the number of columns of H, and n is the length of the time series. The value of L is computed with

$$L = n - M - 1. \tag{4}$$

2.2.2. Singular Value Decomposition. The SVD process is implemented over the matrix H obtained in the last subsection. Let H be an $M \times N$ real matrix; then there exist an $M \times M$ orthogonal matrix U, an $N \times N$ orthogonal matrix V, and an $M \times N$ diagonal matrix S with diagonal entries $s_1 \geq s_2 \geq \cdots \geq s_p$, with $p = \min(M, N)$, such that $U^T HV = S$. Moreover, the numbers $s_1, s_2, \ldots, s_p$ are uniquely determined by H [39]:

$$H = U \times S \times V^T. \tag{5}$$

The extraction of the components is developed through the singular values s_i, the orthogonal matrix U, and the orthogonal matrix V, for each singular value is obtained one matrix A_i, with $i = 1, \ldots, M$:

$$A_i = s(i) \times U(:, i) \times V(:, i)^T. \tag{6}$$

Therefore the matrix A_i contains the ith component; the extraction process is

$$C_i = \left[A_i(1,:) \;\; A_i(2, N : M)^T\right], \tag{7}$$

where C_i is the ith component and the elements of C_i are located in the first row and last column of A_i.

The energy of the obtained components is computed with

$$E_i = \frac{s_i^2}{\sum_{i=1}^{M} s_i^2}, \tag{8}$$

where E_i is the energy of the ith component and s_i is the ith singular value. When $M > 2$, the component C_H is computed with the sum of the components from 2 to M, as follows:

$$C_H = \sum_{i=2}^{M} C_i. \tag{9}$$

3. Proposed Forecasting Models

3.1. Autoregressive Integrated Moving Average Model. The ARIMA model is the generalization of the ARMA model; ARIMA processes are applied on nonstationary time series to convert them in stationary, in ARIMA(P, D, Q) process; D is a nonnegative integer that determines the order and P and Q are the polynomials degrees [40].

The time series transformation process to obtain a stationary time series from a nonstationary is developed by means of differentiation; the time series x_t will be nonstationary of order d if $x_t = \Delta^d x_t$ is stationary; the transformation process is

$$\Delta x_t = x_t - x_{t-1}, \tag{10a}$$

$$\Delta^{j+1} x_t = \Delta^j x_t - \Delta^j x_{t-1}, \tag{10b}$$

where x is the time series, t is the time instant, and j is the number of differentiations obtained, that is, because the process is iterative. Once we obtained the stationary time series, the estimation is computed with

$$\hat{x}_t = \sum_{i=1}^{P} \alpha_i z_{t-i} + \sum_{i=1}^{Q} \beta_i e_{t-i} + e_t, \tag{11}$$

where α_i represents the coefficients of the AR terms of order P and β_i denotes the coefficients of the MA terms of order Q, z is the input regressor vector, which is defined in Section 3.2, and e is a source of randomness and is called white noise. The coefficients α_i and β_i are estimated using the maximum likelihood estimation (MLE) algorithm [40].

3.2. Neural Network Forecasting Model. The ANN has a common structure of three layers [41]; the inputs are the lagged terms contained in the regressor vector z; at hidden layer the sigmoid transfer function is applied, and at output layer the forecasted value is obtained. The ANN output is

$$\hat{x}(n) = \sum_{j=1}^{Q} v_j h_j, \tag{12a}$$

$$h_j = f\left[\sum_{i=1}^{K} w_{ji} z_i(n)\right], \tag{12b}$$

where $\hat{x}$ is the estimated value, n is the time instant, Q is the number of hidden nodes, v_j and w_{ji} are the linear and nonlinear weights of the ANN connections, respectively, z_i represents the ith lagged term, and $f(\cdot)$ is the sigmoid transfer function denoted by

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{13}$$

The lagged terms are the input of the ANN and they are contained in the regressor vector z, whose representation for MA smoothing is

$$Z(t) = [\tilde{s}(t-1), \tilde{s}(t-2), \ldots, \tilde{s}(t-K)], \quad (14)$$

where $K = P$ lagged terms and P and Q were defined in Section 3.1.

The representation of z for HSVD smoothing is

$$\begin{aligned} z(t) = [&C_L(t-1), \ldots, C_L(t-K), \\ &C_H(t-1), \ldots, C_H(t-K)], \end{aligned} \quad (15)$$

where $K = 2P$ lagged terms.

The ANN is denoted by ANN$(K, Q, 1)$, with K inputs, Q hidden nodes, and 1 output. The parameters v and w are updated with the application of two learning algorithms: one based on PSO and the other on RPROP.

3.2.1. Learning Algorithm Based on PSO. The weight of the ANN connections, w and v are adjusted with PSO learning algorithm. In the swarm the N_p particles have a position vector $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$ and a velocity vector $V_i = (V_{i1}, V_{i2}, \ldots, V_{iD})$; each particle is considered a potential solution in a D-dimensional search space. During each iteration the particles are accelerated toward the previous best position denoted by p_{id} and toward the global best position denoted by p_{gd}. The swarm has N_p rows and D columns, and it is initialized randomly; D is computed with $P \times N_h + N_h$; the process finishes when the lowest error is obtained based on the fitness function evaluation or when the maximum number of iterations is reached [42], as follows:

$$\begin{aligned} V_{id}^{l+1} = I^l \times V_{id}^l &+ c_1 \times rd_1 \left(p_{id}^l + X_{id}^l\right) \\ &+ c_2 \times rd_2 \left(p_{gd}^l + X_{id}^l\right), \end{aligned} \quad (16a)$$

$$X_{id}^{l+1} = X_{id}^l + V_{id}^{l+1}, \quad (16b)$$

$$I^l = I_{\max}^l - \frac{I_{\max}^l - I_{\min}^l}{\text{iter}_{\max}} \times l, \quad (16c)$$

where $i = 1, \ldots, N_p$, $d = 1, \ldots, D$; I denotes the inertia weight; c_1 and c_2 are learning factors, rd_1 and rd_2 are positive random numbers in the range $[0, 1]$ under normal distribution, and l is the lth iteration. Inertia weight has linear decreasing, $I_{\max}$ is the maximum value of inertia, $I_{\min}$ is the lowest, and iter$_{\max}$ is total of iterations.

The particle X_{id} represents the optimal solution, in this case the set of weights w and v for the ANN.

3.2.2. Learning Algorithm Based on Resilient Backpropagation. RPROP is an efficient learning algorithm that performs a direct adaptation of the weight step based on local gradient information; it is considered a first-order method. The update rule depends only on the sign of the partial derivative of the arbitrary error regarding each weight of the ANN. The individual step size Δ_{ij} is computed for each weight using this rule [33], as follows:

$$\Delta_{ij}^{(t)} := \begin{cases} \eta^{+} \cdot \Delta_{ij}^{t-1} & \text{if } \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} \dfrac{\partial E}{\partial w_{ij}}^{(t)} > 0, \\ \eta^{-} \cdot \Delta_{ij}^{t-1} & \text{if } \dfrac{\partial E}{\partial w_{ij}}^{(t-1)} \dfrac{\partial E}{\partial w_{ij}}^{(t)} < 0, \\ \Delta_{ij}^{t-1} & \text{else,} \end{cases} \quad (17)$$

where $0 < \eta^{-} < 1 < \eta^{+}$. If the partial derivative $\partial E / \partial w_{ij}$ has the same sign for consecutive steps, the step size is slightly increased by the factor η^{+} in order to accelerate the convergence, whereas if it changes the sign, the step size is decreased by the factor η^{-}. Additionally in the case of a change in the sign, there should be no adaptation in the succeeding step; in the practice this can be done by setting $\partial E / \partial w_{ij} = 0$ in the adaptation rule Δ_{ij}. Finally the weight update and the adaptation are performed after the gradient information of all the weights is computed.

4. Forecasting Accuracy Metrics

The forecasting accuracy is evaluated with the metrics root mean squared error (RMSE), generalized cross validation (GCV), mean absolute percentage error (MAPE), and relative error (RE):

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{N_v} \sum_{i=1}^{N_v} (x_i - \hat{x}_i)^2}, \\ \text{GCV} &= \frac{\text{RMSE}}{(1 - K/N_v)^2}, \\ \text{MAPE} &= \left[\frac{1}{N_v} \sum_{i=1}^{N_v} \left| \frac{(x_i - \hat{x}_i)}{x_i} \right| \right] \times 100, \\ \text{RE} &= \sum_{i=1}^{N_v} \frac{(x_i - \hat{x}_i)}{x_i}, \end{aligned} \quad (18)$$

where N_v is the validation (testing) sample size, x_i is the ith observed value, $\hat{x}_i$ is the ith estimated value, and K is the length of the input regressor vector.

5. Results and Discussions

The data used for forecasting is the time series of injured people in traffic accidents occurring in Valparaíso, from 2003 to 2012; they were obtained from CONASET, Chile [43]. The data sampling period is weekly, with 531 registers as shown in Figure 2(a); the series was separated for training and testing, and by trial and error the 85% for training and the 15% for testing were determined.

5.1. ARIMA Forecasting

5.1.1. Moving Average Smoothing. The raw time series is smoothed using 3-point moving average, whose obtained

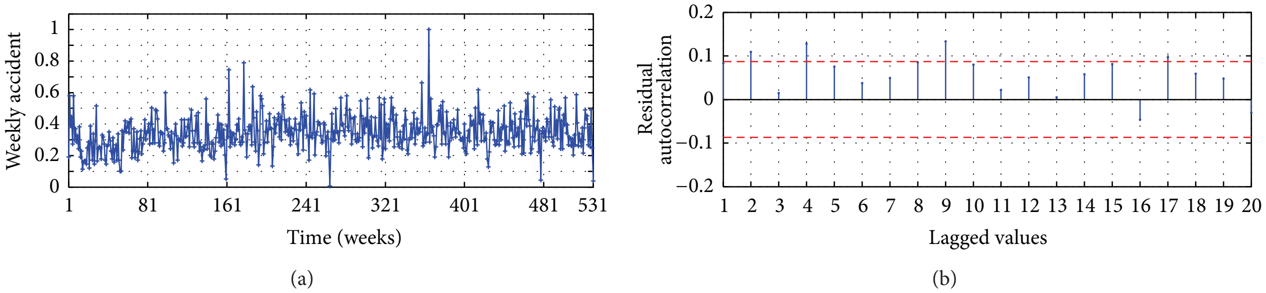


Figure 2: Accidents time series: (a) raw data and (b) autocorrelation function.

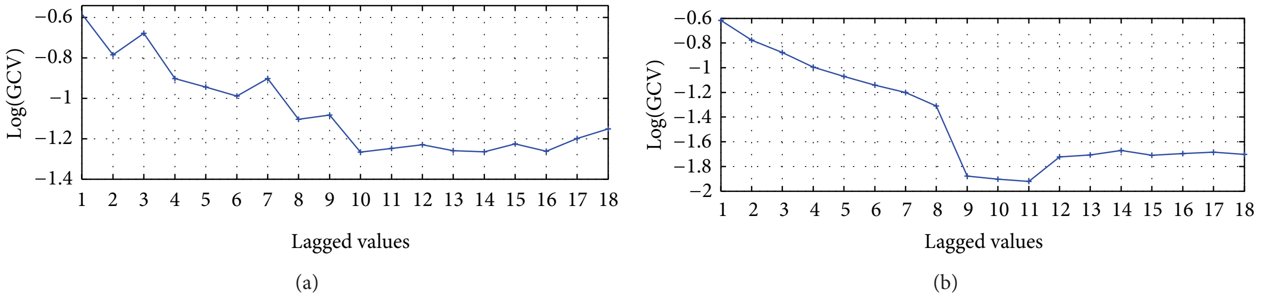


Figure 3: (a) MA smoothing and (b) HSVD smoothing.

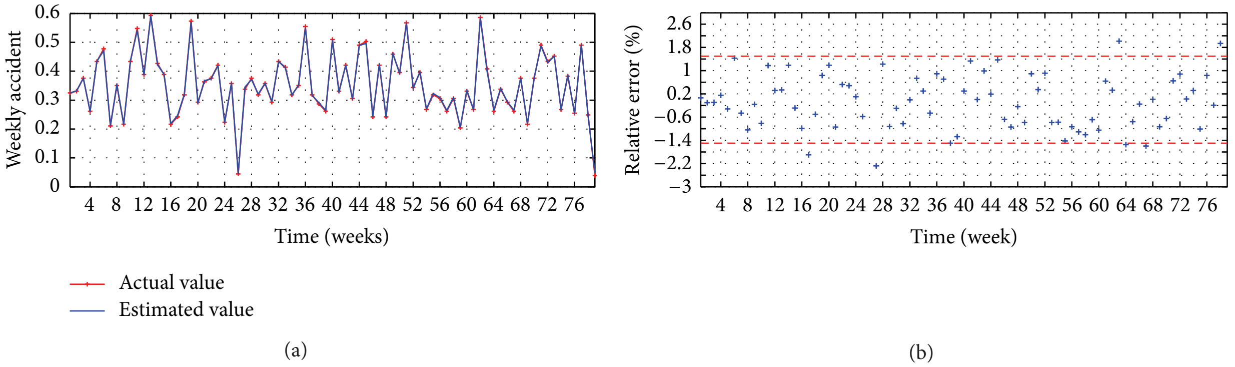


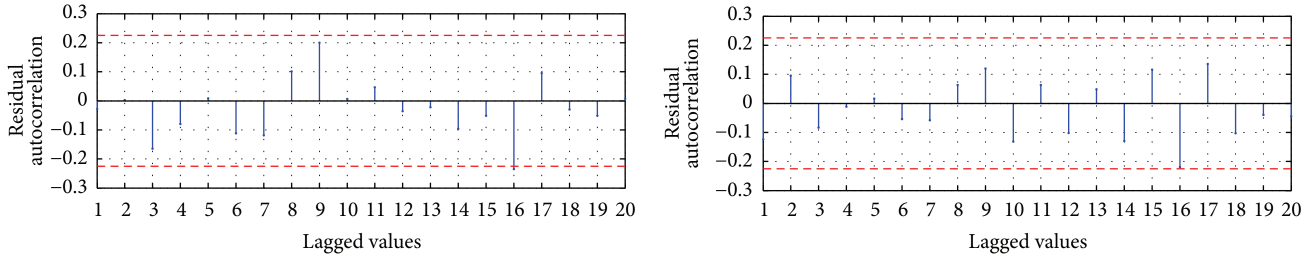
Figure 4: MA-ARIMA(9,0,10), (a) observed versus estimated (b) relative error.

values are used as input of the forecasting model ARIMA(P, D, Q); this is presented in Figure 1(a). The effective order of the polynomial for the AR terms is found to be $P = 9$ and the differentiation parameter is found to be $D = 0$; those values were obtained from the autocorrelation function (ACF) shown in Figure 2(b); to set the order Q of MA terms, is evaluated the metric GCV versus the Q Lagged values. The results of the GCV are presented in Figure 3(a); it shows that the lowest GCV is achieved with 10 lagged values. Therefore the configuration of the model is denoted by AM-ARIMA(9,0,10).

The evaluation executed in the testing stage is presented in Figures 4 and 5(a) and Table 1. The observed values versus the estimated values are illustrated in Figure 4(a), reaching a good accuracy, while the relative error is presented in Figure 4(b), which shows that the 87% of the points present an error lower than ±1.5%.

For the evaluation of the serial correlation of the model errors the ACF is applied, whose values are presented in Figure 5(a); it shows that ACF for a lag of 16 is slightly lower than the 95% confidence limit; however the rest of the coefficients are inside the confidence limit; therefore in the errors of the model AM-ARIMA(9,0,10) there is no serial correlation; we can conclude that the proposed model explains efficiently the variability of the process.

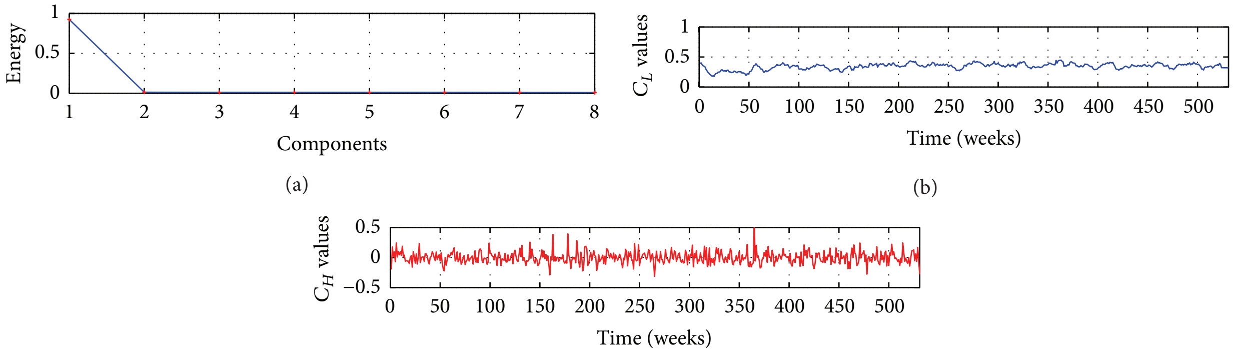
5.1.2. HSVD Smoothing. In this section the forecasting strategy presented in Figure 1(b) is evaluated; to implement this strategy in first instance the time series is mapped using the Hankel matrix, after the SVD process is executed to obtain



(a)

(b)

Figure 5: Residual ACF: (a) MA-ARIMA(9,0,10) and (b) SVD-ARIMA(9,0,11).



(a)

(b)

(c)

Figure 6: Accidents time series: (a) components energy, (b) low frequency component, and (c) high frequency component.

Table 1: Forecasting with ARIMA.

	MA-ARIMA	HSVD-ARIMA
RMSE	0.0034	0.00073
MAPE	1.12%	0.26%
GCV	0.006	0.0013
RE ± 1.5%	87%	—
RE ± 0.5%	—	95%

the M components. The value of M is found through the computation of the singular values of the decomposition; this is presented in Figure 6(a); as shown in Figure 6(a), the major quantity of energy is captured by the two first components; therefore in this work only two components have been selected with $M = 2$. The first component extracted represents the long-term trend (C_L) of the time series, while the second represents the short-term component of high frequency fluctuation (C_H). The components C_L and C_H are shown in Figures 6(b) and 6(c), respectively.

To evaluate the model, in this section $P = 9$ and $D = 0$ are used, and Q is evaluated using the GCV metric for $1 \leq Q \leq 18$; then the effective value $Q = 11$ is found, as shown in Figure 3(b); therefore the forecasting model is denoted by HSVD-ARIMA(9,0,11).

Once P and Q are found, the forecasting is executed with the testing data set, and the results of HSVD-ARIMA(9,0,11) are shown in Figures 7(a), 7(b), and 5(b) and Table 1. Figure 7(a) shows the observed values versus the estimates vales, and a good adjusting between them is found. The relative errors are presented in Figure 7(b); it shows that the 95% of the points present an error lower than ±0.5%.

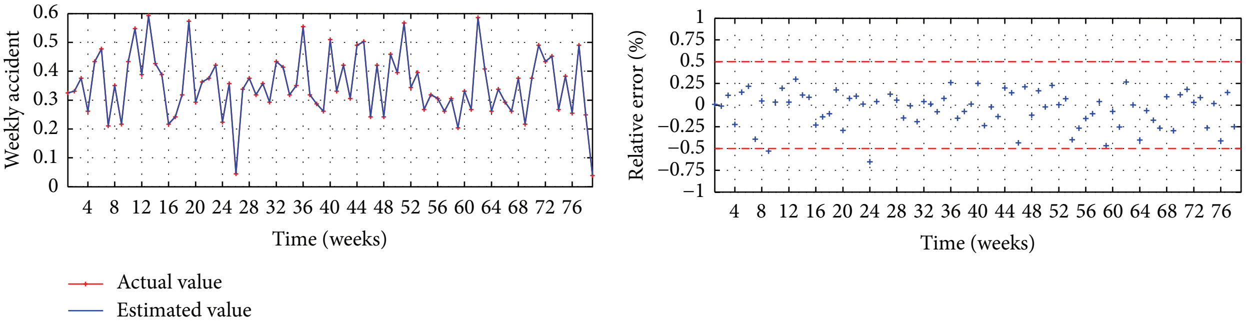
For the evaluation of the serial correlation of the model errors the ACF is applied, whose values are presented in Figure 5(b); it shows that all the coefficients are inside the confidence limit; therefore in the model errors there is no serial correlation; we can conclude that the proposed model HSVD-ARIMA(9,0,11) explains efficiently the variability of the process.

The results presented in Table 1 show that the major accuracy is achieved with the model HSVD-ARIMA(9,0,11), with a RMSE of 0.00073 and a MAPE of 0.26%; the 95% of the points have a relative error lower than ±0.5%.

5.2. ANN Forecasting Model Based on PSO

5.2.1. Moving Average Smoothing. The raw time series is smoothed using the moving average of order 3, whose obtained values are used as input of the forecasting model presented in Figure 1(a). The calibration executed in Section 5.1.1 is used for the neural network and then an ANN$(K, Q, 1)$ is used, with $K = 9$ inputs (lagged values), $Q = 10$ hidden nodes, and 1 output.

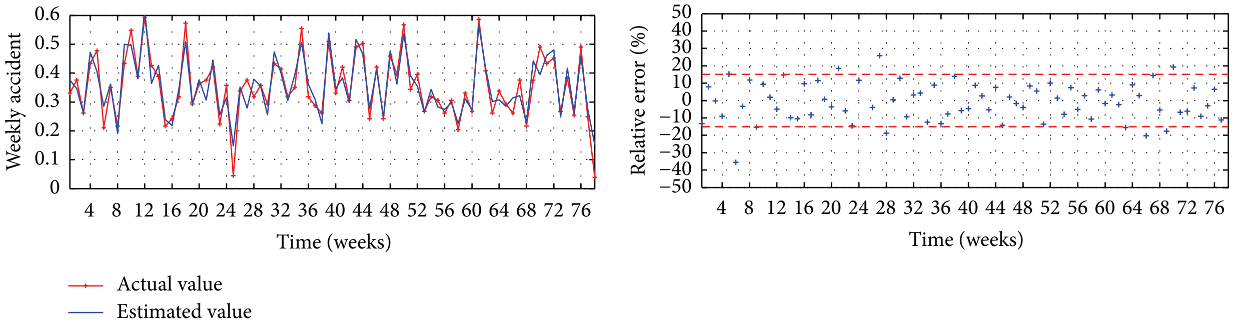
The evaluation executed in the testing stage is presented in Figures 8 and 9(a) and Table 2. The observed values versus the estimated values are illustrated in Figure 8(a), reaching a good accuracy, while the relative error is presented in Figure 8(b),



(a)

(b)

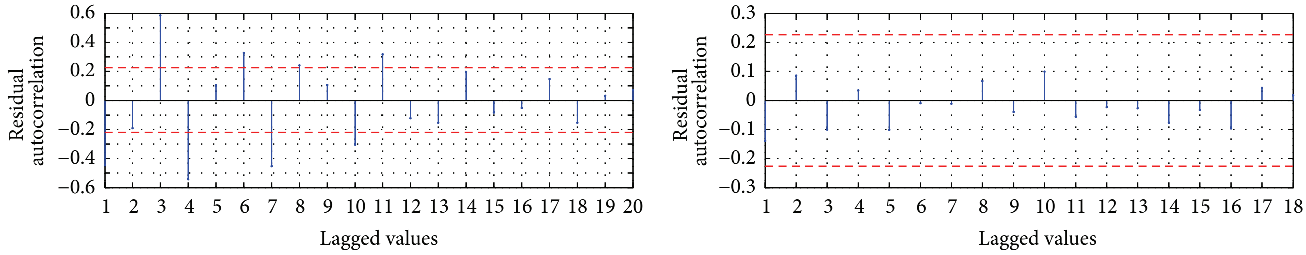
Figure 7: SVD-ARIMA(9,0,11): (a) observed versus estimated and (b) relative error.



(a)

(b)

Figure 8: MA-ANN-PSO(9,10,1): (a) observed versus estimated and (b) relative error.



(a)

(b)

Figure 9: Residual ACF: (a) MA-ANN-PSO(9,10,1) and (b) HSVD-ANN-PSO(9,11,1).

Table 2: Forecasting with ANN-PSO.

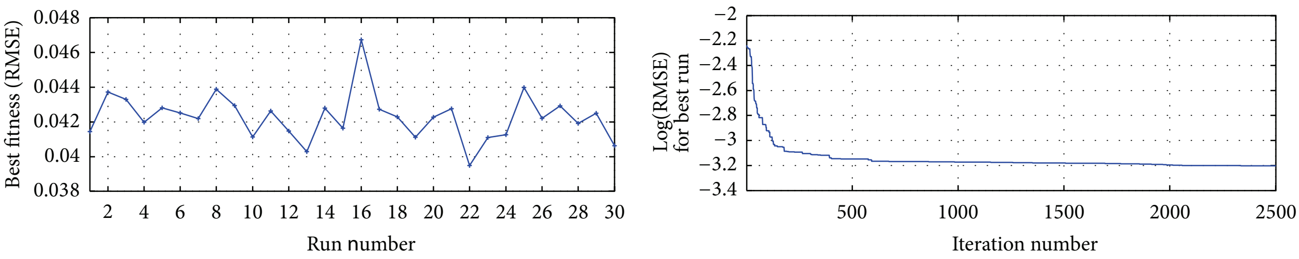
	MA-ANN-PSO	HSVD-ANN-PSO
RMSE	0.04145	0.0123
MAPE	15.51%	5.45%
GCV	0.053	0.022
RE ± 15%	85%	—
RE ± 4%	—	95%

which shows that the 85% of the points present an error lower than ±15%.

For the evaluation of the serial correlation of the model errors the ACF is applied, whose values are presented in Figure 9(a); it shows that there are values with significative difference from zero to 95% of the confidence limit; by example the three major values are obtained when the lagged value is equal to 3, 4, and 7 weeks. Therefore in the residuals there is serial correlation; this implies that the model MA-ANN-PSO(9,10,1) is not recommended for future usage and probably other explanatory variables should be added in the model.

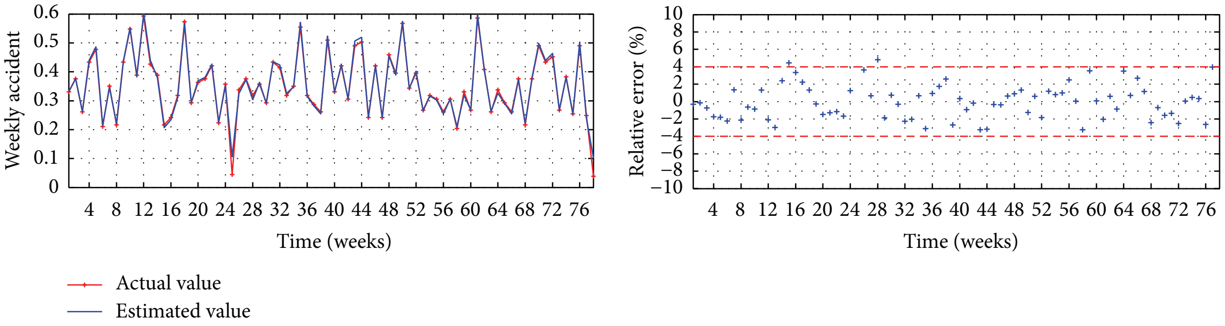
The process was run 30 times and the best result was reached in the run 22 as shown in Figure 10(a); Figure 10(b) presents the RMSE metric for the best run.

5.2.2. HSVD Smoothing. In this section the forecasting strategy presented in Figure 1(b) is evaluated; the HSVD smoothing strategy is applied using the same calibration



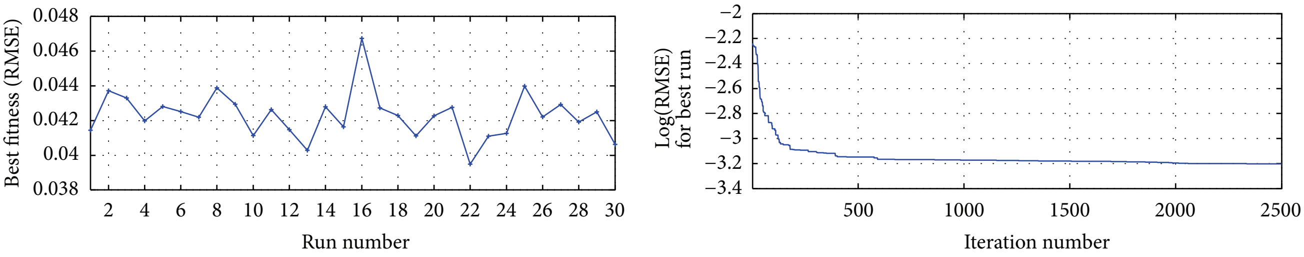
(a) (b)

Figure 10: MA-ANN-PSO(9,10,1): (a) run versus fitness for 2500 iterations and (b) iterations number for the best run.



(a) (b)

Figure 11: HSVD-ANN-PSO(9,11,1): (a) observed versus estimated and (b) relative rrror.



(a) (b)

Figure 12: HSVD-ANN-PSO(9,11,1): (a) run versus fitness for 2500 iterations and (b) iterations number for the best run.

explained in Section 5.1.2; then an ANN(K, Q, 1) is used, with $K = 9$ inputs (lagged values), $Q = 11$ hidden nodes, and 1 output.

The evaluation executed in the testing stage is presented in Figures 11 and 9(b) and Table 2. The observed values versus the estimated values are illustrated in Figure 11(a), reaching a good accuracy, while the relative error is presented in Figure 11(b), which shows that the 95% of the points present an error lower than ±4%.

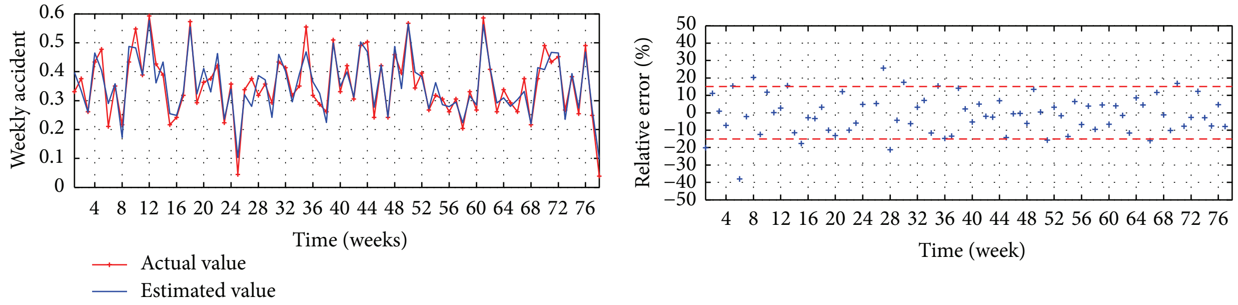
For the evaluation of the serial correlation of the model errors the ACF is applied, whose values are presented in Figure 9(b); it shows that all the coefficients are inside the confidence limit of 95% and statistically are equal to zero; therefore in the model errors there is no serial correlation; we can conclude that the proposed model HSVD-ANN-PSO(9,11,1) explains efficiently the variability of the process.

The process was run 30 times and the best result was reached in the run 11 as shown in Figure 12(a); Figure 12(b) presents the RMSE metric for the best run.

The results presented in Table 2 show that the major accuracy is achieved with the model HSVD-ANN-PSO(9,11,1), with a RMSE of 0.0123 and a MAPE of 5.45%; the 95% of the points have a relative error lower than ±4%.

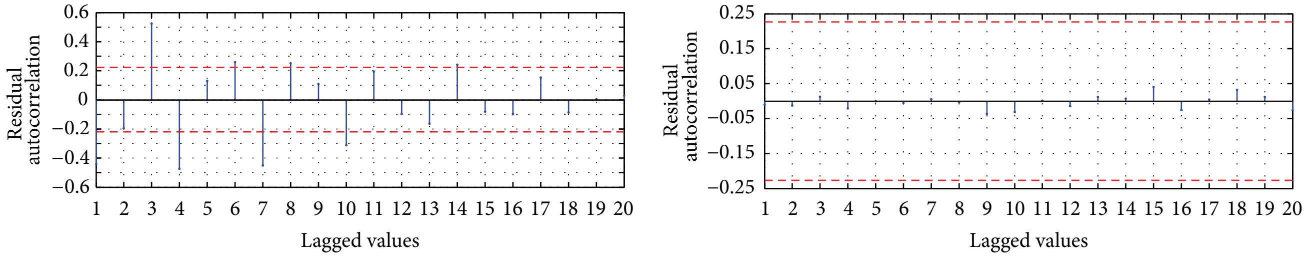
5.3. *ANN Forecasting Model Based on RPROP*

5.3.1. Moving Average Smoothing. The raw time series is smoothed using the moving average of order 3, whose obtained values are used as input of the forecasting



(a) (b)

Figure 13: MA-ANN-RPROP(9,10,1): (a) observed versus estimated and (b) relative error.



(a) (b)

Figure 14: Residual ACF: (a) MA-ANN-RPROP(9,10,1) and (b) HSVD-ANN-RPROP(9,11,1).

Table 3: Forecasting with ANN-RPROP.

	MA-ANN-RPROP	HSVD-ANN-RPROP
RMSE	0.0384	0.024
MAPE	12.25%	8.08%
GCV	0.0695	0.045
RE ± 15%	81%	—
RE ± 4%	—	96%

model presented in Figure 1(a). The calibration executed in Section 5.1.1 is used for the neural network; then an ANN$(K, Q, 1)$ is used, with $K = 9$ inputs (lagged values), $Q = 10$ hidden nodes, and 1 output.

The evaluation executed in the testing stage is presented in Figures 13 and 14(a) and Table 3. The observed values versus the estimated values are illustrated in Figure 13(a), reaching a good accuracy, while the relative error is presented in Figure 13(b), which shows that the 81% of the points present an error lower than ±15%.

For the evaluation of the serial correlation of the model errors the ACF is applied, whose values are presented in Figure 14(a); it shows that there are values with significative difference from zero to 95% of the confidence limit; by example the three major values are obtained when the lagged value is equal to 3, 4, and 7 weeks. Therefore in the residuals there is serial correlation; this implies that the model MA-ANN-RPROP(9,10,1) is not recommended for future usage and probably other explanatory variables should be added in the model.

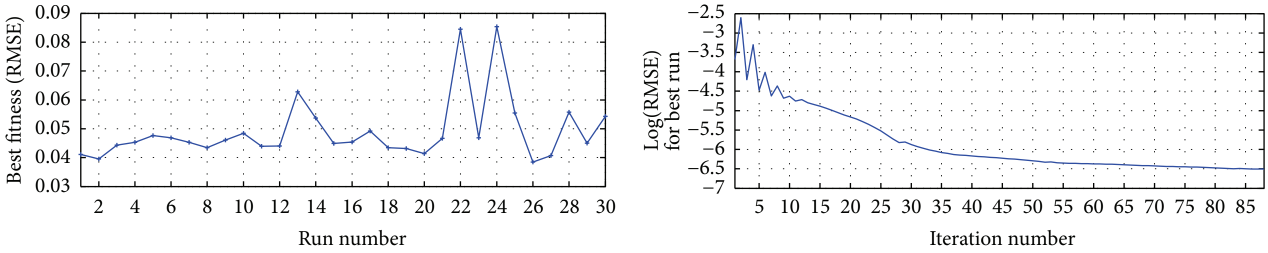
The process was run 30 times, and the best result was reached in the run 26 as shown in Figure 15(a); Figure 15(b) presents the RMSE metric for the best run.

5.3.2. HSVD Smoothing. In this section the forecasting strategy presented in Figure 1(b) is evaluated, the HSVD smoothing strategy is applied using the same calibration explained in Section 5.1.2, and then an ANN$(K, Q, 1)$ is used, with $K = 9$ inputs (lagged values), $Q = 11$ hidden nodes, and 1 output.

The evaluation executed in the testing stage is presented in Figures 16 and 14(b) and Table 3. The observed values versus the estimated values are illustrated in Figure 16(a), reaching a good accuracy, while the relative error is presented in Figure 16(b), which shows that the 96% of the points present an error lower than ±4%.

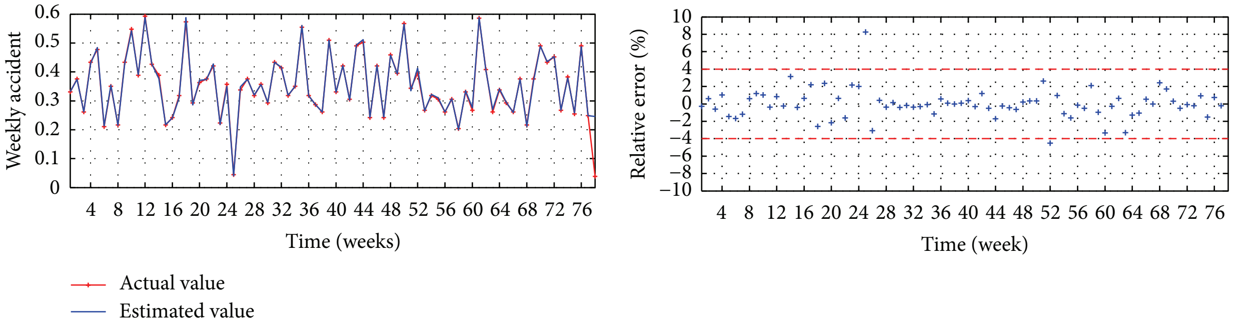
For the evaluation of the serial correlation of the model errors the ACF is applied, whose values are presented in Figure 14(b); it shows that all the coefficients are inside the confidence limit and statistically are equal to zero; therefore in the model errors there is no serial correlation; we can conclude that the proposed model HSVD-ANN-RPROP(9,11,1) explains efficiently the variability of the process. The process was run 30 times and the first best result was reached in the run 21 as shown in Figure 17(a); Figure 17(b) presents the RMSE metric for the best run.

The results presented in Table 3 show that the major accuracy is achieved with the model HSVD-ANN-RPROP(9,11,1), with a RMSE of 0.024 and a MAPE of 8.08%; the 96% of the points have a relative error lower than ±4%.



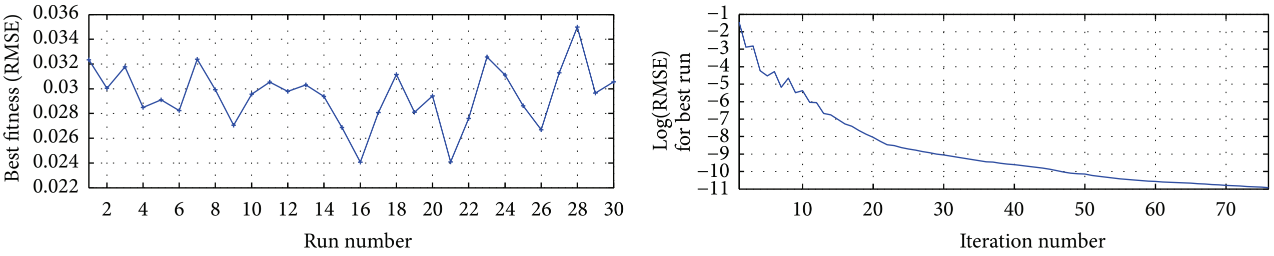
(a) (b)

FIGURE 15: MA-ANN-RPROP(9,10,1): (a) run versus fitness for 85 iterations and (b) iterations number for the best run.



(a) (b)

FIGURE 16: HSVD-ANN-RPROP(9,11,1): (a) observed versus estimated and (b) relative error.



(a) (b)

FIGURE 17: HSVD-ANN(9,11,1): (a) run versus fitness for 70 iterations and (b) iterations number for the best run.

Finally, Pitman's correlation test [44] is used to compare all forecasting models in a pairwise fashion. Pitman's test is equivalent to testing if the correlation (Corr) between Υ and Ψ is significantly different from zero, where Υ and Ψ are defined by

$$\Upsilon = e_1(n) + e_2(n), \quad n = 1, 2, \ldots, N_v, \tag{19a}$$

$$\Psi = e_1(n) - e_2(n), \quad n = 1, 2, \ldots, N_v, \tag{19b}$$

where e_1 and e_2 represent the one-step-ahead forecast error for model 1 and model 2, respectively. The null hypothesis is significant at the 5% significance level if $|\text{Corr}| > 1.96/\sqrt{N_v}$.

The evaluated correlations between Υ and Ψ are presented in Table 4.

The results presented in Table 4 show that statistically there is a significant superiority of the HSVD-ARIMA forecasting model, regarding the rest of models. The results are presented from left to right, where the first is the best model and the last is the worst model.

6. Conclusions

In this paper were proposed two strategies of time series smoothing to improve the forecasting accuracy. The first smoothing strategy is based on moving average of order 3, while the second is based on the Hankel singular value decomposition. The strategies were evaluated with the time

Table 4: Pitman's correlation (Corr) for pairwise comparison six models at 5% of significance and the critical value 0.2219.

Models	M1	M2	M3	M4	M5	M6
M1 := HSVD-ARIMA	—	−0.9146	−0.9931	−0.9983	−0.9994	−0.9993
M2 := MA-ARIMA	—	—	−0.8676	−0.9648	−0.9895	−0.9887
M3 := HSVD-ANN-PSO	—	—	—	−0.6645	−0.8521	−0.8216
M4 := HSVD-ANN-RPRO	—	—	—	—	−0.5129	−0.4458
M5 := ANN-RPROP	—	—	—	—	—	0.1623
M6 := MA-ANN-PSO	—	—	—	—	—	—

series of traffic accidents occurring in Valparaíso, Chile, from 2003 to 2012.

The estimation of the smoothed values was developed through three conventional models, ARIMA, an ANN based on PSO, and an ANN based on RPROP. The comparison of the six models implemented shows that the first best model is HSVD-ARIMA, as it obtained the major accuracy, with a MAPE of 0.26% and a RMSE of 0.00073, while the second best is the model MA-ARIMA, with a MAPE of 1.12% and a RMSE of 0.0034. On the other hand, the model with the lowest accuracy was MA-ANN-PSO with a MAPE of 15.51% and a RMSE of 0.041. Pitman's test was executed to evaluate the difference of the accuracy between the six proposed models and the results show that statistically there is a significant superiority of the forecasting model based on HSVD-ARIMA. Due to the high accuracy reached with the best model, in future works, it will be applied to evaluate new time series of other regions and countries.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by Grant CONICYT/FONDECYT/Regular 1131105 and by the DI-Regular project of the Pontificia Universidad Católica de Valparaíso.

References

[1] J. Abellán, G. López, and J. de Oña, "Analysis of traffic accident severity using decision rules via decision trees," *Expert Systems with Applications*, vol. 40, no. 15, pp. 6047–6054, 2013.

[2] L. Chang and J. Chien, "Analysis of driver injury severity in truck-involved accidents using a non-parametric classification tree model," *Safety Science*, vol. 51, no. 1, pp. 17–22, 2013.

[3] J. de Oña, G. López, R. Mujalli, and F. J. Calvo, "Analysis of traffic accidents on rural highways using Latent Class Clustering and Bayesian Networks," *Accident Analysis and Prevention*, vol. 51, pp. 1–10, 2013.

[4] M. Fogue, P. Garrido, F. J. Martinez, J. Cano, C. T. Calafate, and P. Manzoni, "A novel approach for traffic accidents sanitary resource allocation based on multi-objective genetic algorithms," *Expert Systems with Applications*, vol. 40, no. 1, pp. 323–336, 2013.

[5] M. A. Quddus, "Time series count data models: an empirical application to traffic accidents," *Accident Analysis and Prevention*, vol. 40, no. 5, pp. 1732–1741, 2008.

[6] J. J. F. Commandeur, F. D. Bijleveld, R. Bergel-Hayat, C. Antoniou, G. Yannis, and E. Papadimitriou, "On statistical inference in time series analysis of the evolution of road safety," *Accident Analysis and Prevention*, vol. 60, pp. 424–434, 2013.

[7] C. Antoniou and G. Yannis, "State-space based analysis and forecasting of macroscopic road safety trends in Greece," *Accident Analysis and Prevention*, vol. 60, pp. 268–276, 2013.

[8] W. Weijermars and P. Wesemann, "Road safety forecasting and ex-ante evaluation of policy in the Netherlands," *Transportation Research A: Policy and Practice*, vol. 52, pp. 64–72, 2013.

[9] A. García-Ferrer, A. de Juan, and P. Poncela, "Forecasting traffic accidents using disaggregated data," *International Journal of Forecasting*, vol. 22, no. 2, pp. 203–222, 2006.

[10] R. Gençay, F. Selçuk, and B. Whitcher, *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*, Academic Press, 2002.

[11] N. Abu-Shikhah and F. Elkarmi, "Medium-term electric load forecasting using singular value decomposition," *Energy*, vol. 36, no. 7, pp. 4259–4271, 2011.

[12] C. Sun and J. Hahn, "Parameter reduction for stable dynamical systems based on Hankel singular values and sensitivity analysis," *Chemical Engineering Science*, vol. 61, no. 16, pp. 5393–5403, 2006.

[13] H. Gu and H. Wang, "Fuzzy prediction of chaotic time series based on singular value decomposition," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1171–1185, 2007.

[14] X. Weng and J. Shen, "Classification of multivariate time series using two-dimensional singular value decomposition," *Knowledge-Based Systems*, vol. 21, no. 7, pp. 535–539, 2008.

[15] N. Hara, H. Kokame, and K. Konishi, "Singular value decomposition for a class of linear time-varying systems with application to switched linear systems," *Systems and Control Letters*, vol. 59, no. 12, pp. 792–798, 2010.

[16] K. Kavaklioglu, "Robust electricity consumption modeling of Turkey using singular value decomposition," *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 268–276, 2014.

[17] W. X. Yang and P. W. Tse, "Medium-term electric load forecasting using singular value decomposition," *NDT & E International*, vol. 37, pp. 419–432, 2003.

[18] K. Kumar and V. K. Jain, "Autoregressive integrated moving averages (ARIMA) modelling of a traffic noise time series," *Applied Acoustics*, vol. 58, no. 3, pp. 283–294, 1999.

[19] J. Hassan, "ARIMA and regression models for prediction of daily and monthly clearness index," *Renewable Energy*, vol. 68, pp. 421–427, 2014.

[20] P. Narayanan, A. Basistha, S. Sarkar, and S. Kamna, "Trend analysis and ARIMA modelling of pre-monsoon rainfall data for western India," *Comptes Rendus Geoscience*, vol. 345, no. 1, pp. 22–27, 2013.

[21] K. Soni, S. Kapoor, K. S. Parmar, and D. G. Kaskaoutis, "Statistical analysis of aerosols over the gangetichimalayan region using ARIMA model based on long-term MODIS observations," *Atmospheric Research*, vol. 149, pp. 174–192, 2014.

[22] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[23] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A modified particle swarm optimizer with dynamic adaptation," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1205–1213, 2007.

[24] M. S. Arumugam and M. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 324–336, 2008.

[25] B. K. Panigrahi, V. Ravikumar Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy Conversion and Management*, vol. 49, no. 6, pp. 1407–1415, 2008.

[26] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing Journal*, vol. 11, no. 4, pp. 3658–3670, 2011.

[27] X. Jiang, H. Ling, J. Yan, B. Li, and Z. Li, "Forecasting electrical energy consumption of equipment maintenance using neural network and particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 194730, 8 pages, 2013.

[28] J. Chen, Y. Ding, and K. Hao, "The bidirectional optimization of carbon fiber production by neural network with a GA-IPSO hybrid algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 768756, 16 pages, 2013.

[29] J. Zhou, Z. Duan, Y. Li, J. Deng, and D. Yu, "PSO-based neural network optimization and its utilization in a boring machine," *Journal of Materials Processing Technology*, vol. 178, no. 1–3, pp. 19–23, 2006.

[30] M. A. Mohandes, "Modeling global solar radiation using Particle Swarm Optimization (PSO)," *Solar Energy*, vol. 86, no. 11, pp. 3137–3145, 2012.

[31] L. F. De Mingo López, N. Gómez Blas, and A. Arteta, "The optimal combination: grammatical swarm, particle swarm optimization and neural networks," *Journal of Computational Science*, vol. 3, no. 1-2, pp. 46–55, 2012.

[32] A. Yazgan and I. H. Cavdar, "A comparative study between LMS and PSO algorithms on the optical channel estimation for radio over fiber systems," *Optik*, vol. 125, no. 11, pp. 2582–2586, 2014.

[33] M. Riedmiller and H. Braun, "A direct adaptive me thod for faster backpropagation learning: the RPROP algorithm," in *Proceedings of the IEEE International Conference of Neural Networks*, E. H. Ruspini, Ed., pp. 586–591, 1993.

[34] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," *Neurocomputing*, vol. 50, pp. 105–123, 2003.

[35] P. G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[36] L. Aburto and R. Weber, "Improved supply chain management based on hybrid demand forecasts," *Applied Soft Computing Journal*, vol. 7, no. 1, pp. 136–144, 2007.

[37] M. Khashei and M. Bijari, "A new hybrid methodology for nonlinear time series forecasting," *Modelling and Simulation in Engineering*, vol. 2011, Article ID 379121, 5 pages, 2011.

[38] R. A. Yafee and M. McGee, *An Introduction to Time Series Analysis and Forecasting: With Applications of SAS and SPSS*, Academic Press, New York, NY, USA, 2000.

[39] TS. Shores, *Applied Linear Algebra and Matrix Analysis*, Springer, 2007.

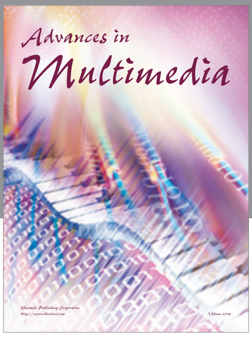
[40] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, Springer, Berlin, Germany, 2nd edition, 2002.

[41] J. A. Freeman and D. M. Skapura, *Neural Networks, Algorithms, Applications, and Programming Techniques*, Addison-Wesley, 1991.

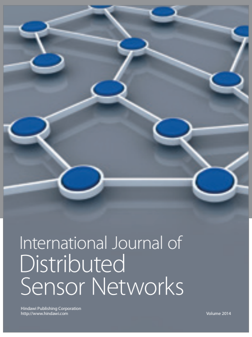
[42] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*, Morgan Kaufmann, 2001.

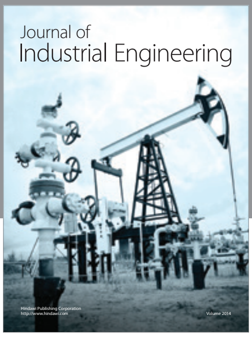
[43] Conaset, 2014, http://www.conaset.cl.

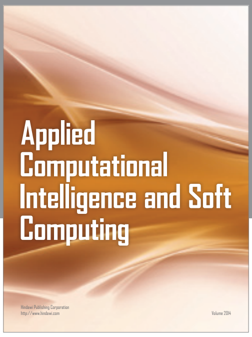
[44] K. Hipel and A. McLeod, *Time Series Modelling of Water Resources and Environmental Systems*, Elsevier, 1994.

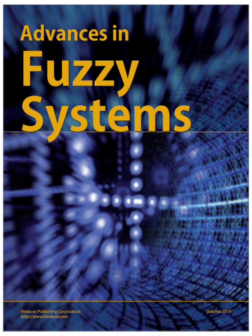

Advances in
Multimedia

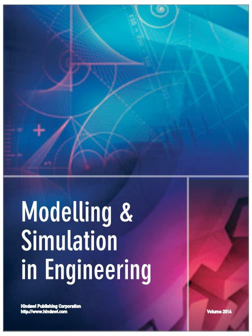
The Scientific
World Journal

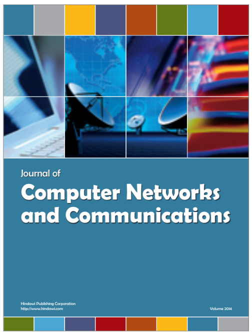

International Journal of
Distributed
Sensor Networks

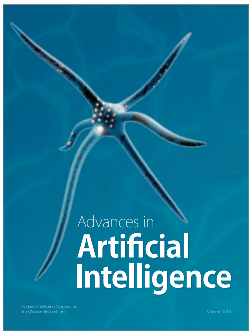

Journal of
Industrial Engineering

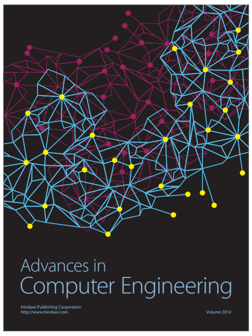

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

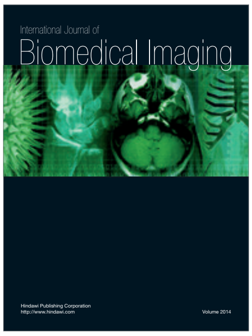

Modelling &
Simulation
in Engineering

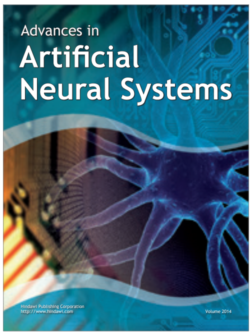

Journal of
Computer Networks
and Communications

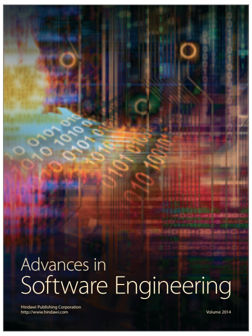

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

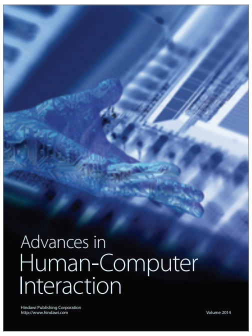
International Journal of
Computer Games
Technology

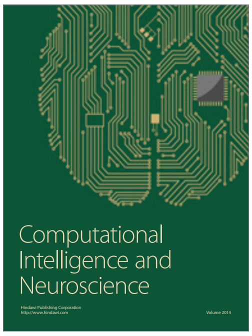

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering