Container flow forecasting through neural networks based on metaheuristics

M. Milenković[1,2] · N. Milosavljevic[3] · N. Bojović[2] · S. Val[1]

Abstract

In this paper we propose a fuzzy neural network prediction approach based on metaheuristics for container flow forecasting. The approach uses fuzzy if–then rules for selection between two different heuristics for developing neural network architecture, simulated annealing and genetic algorithm, respectively. These non-parametric models are compared with traditional parametric ARIMA technique. Time series composed from monthly container traffic observations for Port of Barcelona are used for model developing and testing. Models are compared based on the most important criteria for performance evaluation and for each of the data sets (total container traffic, loaded, unloaded, transit and empty) the appropriate model is selected.

1 Introduction

The aim of this paper is to emphasize the importance of container flow forecasting and provide a comprehensive tool for that purpose. Container manipulation process in terminals is characterized by high uncertainty which influences on regularity of intermodal transport service. Therefore, in order to reduce this uncertainty and to improve efficiency of intermodal transport service, forecasts of container throughput in terminals are required. Forecasting of container throughput represents an essential component of planning and control in port intermodal terminals. The accuracy of predicting container flow volume contributes to higher utilisation of assets,

✉ M. Milenković
mimilenkovic@zlc.edu.es

[1] Zaragoza Logistics Center, Zaragoza, Spain

[2] Division for Management in Railway, Rolling Stock and Traction, The Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade, Serbia

[3] Department for Mathematical Sciences, State University of Novi Pazar, Novi Pazar, Serbia

increased reliability and flexibility, lower lead time and costs of intermodal transport chain. Forecasting of future throughput is also important for planning the development of terminal infrastructure.

There are numerous forecasting techniques which are applicable to container throughput forecasting. In the context of this paper, all techniques may be classified in those handling only linear time series data and advanced methods capable to model complex nonlinear relationships. Time series models and causal analysis models generate accurate performances only in case of linear or near linear container flow time series. On the other side, Artificial Neural Networks (ANNs) can handle complex patterns and generate models which adequately reflect nonlinear relationships. In case of ANNs, there is no need to define an explicit model form. Model is adaptively determined based on the characteristics of the time series. However, potential issues in developing an efficient ANN structure may arise in determining an optimal set of ANN input variables and number of neurons in hidden layer. This limitation motivated authors to look for an improved container throughput forecasting methodology. Therefore, in this paper we propose a hybrid methodology for developing an optimal ANN architecture for the container throughput in port terminals. Two well-known global search heuristic methods are used for developing a neural network (NN) architecture, genetic algorithm (GA) and differential simulated annealing (SA) algorithm, respectively. Both methods use if–then rules for selecting the optimal NN architecture based on their performances.

Comparisons of proposed approaches, with each other and with traditional ARIMA method, are performed on a case study of Port of Barcelona. Performances were evaluated according to various metrics independently for total container traffic, volume of loaded, unloaded, transit and empty containers. At the time of our analysis a time series of monthly container throughput observations from January 2010 to December 2016 (84 months) were available.

Paper is organised as follows. Section 2 contains a comprehensive analysis of relevant literature. Section 3 describes the methodology used in the container flow time series modelling and forecasting. In Sect. 4 proposed models have been tested on five different time series related to total, transit, loaded, unloaded and empty container flows of Port of Barcelona. To investigate the performance of the proposed approaches, Sect. 4 compares the results obtained by the proposed neural network based approaches and ARIMA. Concluding remarks and future research directions are given in the last section.

2 Literature overview

Due to the importance of container ports for economy and of container throughput forecasting for efficient terminal management and intermodal transport chain planning many research efforts related to container flow forecasting were published in literature. Forecasting approaches for container throughput modeling can be categorized in: time series models, causal analysis models and nonlinear dynamics forecasting models. Time series models that are based on establishing a mathematical model only by historical throughput data, include Autoregressive Integrated

Moving Average (ARIMA) model (Box et al. 2008), exponential smoothing model (Schulze and Prinz 2009; Diaz et al. 2011), grey model (Liu et al. 2007; Peng and Chu 2009) and decomposition approach (X-11) (Chen and Chen 2010). In causal analysis models correlation between container flows and a set of explanatory variables is evaluated and the forecasting model is built based on relevant explanatory variables (Seabrooke et al. 2003; Hui et al. 2004). In contrast to these two classes, nonlinear dynamics forecasting models, such as Artificial Neural Networks (ANNs) or ANFIS method (Atsalakis and Valavanis 2008, 2009; Atsalakis et al. 2011; Lam et al. 2006; Lingras et al. 2000; Vlahogianni et al. 2004, 2005) represent data driven, self-adaptive methods which learn from examples and capture functional relationships within the time series even if the underlying relationships are complicated to describe (Zhang et al. 1998; Wei and Chen 2012). Remaining part of this section is dedicated to more detailed description of the relevant approaches related to container throughput forecasting. Fung (2002) provided a systematic approach to forecast the demand for Hong Kong container handling services. Author compared forecasting results of developed structural error correction model (SECM) and forecasts published by the Hong Kong Port and Maritime Board (PMB) and found out that the resulting forecasts were more accurate than that reported by government authority. Seabrooke et al. (2003) use regression analysis for predicting cargo growth and the development of the port of Hong Kong. Factors affected cargo throughput in Hong Kong are identified, qualitatively evaluated and then entered into forecast model. Hui et al. (2004) forecast Hong Kong's port cargo throughput by estimating a cointegrated error correction model, which addressed the non-stationary issue and the reliability difficulty of Port and Maritime Board's simple regression model.

Lam et al. (2004) proposed a neural network approach for forecasting port cargo throughput in Hong Kong. Results of the developed approach were compared with regression analysis and it was concluded that neural network model significantly outperforms the regression model. In his Ph.D. dissertation Chou (2004) applied the graded multiple integrals representation method to model the total import and export container volume for ports in Taiwan. Liu et al. (2007) applied a grey prediction model and a cubic polynomial curve prediction model combined with radial basis neural network to predict the port container throughput. Results have shown that combined forecast methods produced more accurate predictions than the simplex method. Chou et al. (2008) proposed a modified regression model for forecasting the volumes of Taiwan's import containers. Authors compared developed model with the traditional regression model and found out that the modified regression model exhibits higher prediction accuracy. Peng and Chu (2009) compared six univariate forecasting models in order to find the most accurate prediction of container throughput. Authors compared classical decomposition model, the trigonometric regression model, the regression model with seasonal dummy variables, the grey and hybrid grey model and the seasonal ARIMA (SARIMA) model. Based on performance measures authors have found that the classical decomposition model appears to be the best model for forecasting container throughput with seasonal variations. Schulze and Prinz (2009) considered container transshipment at German ports using the SARIMA model and the Holt–Winters exponential smoothing approach for forecasting the quarterly

data of the German container transshipment. Different models are identified and tested for the whole German container throughput and for the throughput with destinations Asia, Europe and North America. Comparisons of predictive performances have shown that SARIMA models perform slightly better than the Holt–Winter approach. Hui et al. (2010) proposed a nonlinear multivariate regression model for port throughput modeling. The model specified demonstrates high exploratory ability. Chen and Chen (2010) developed an optimal predictive model of volumes of container throughput at ports by using genetic programming, decomposition approach and SARIMA models. Validation of models has been performed on a historical data from Taiwan's major ports and results suggest that genetic programming is the optimal method for this case. Thirty-seven ANN models were built to model thirty-seven types of freight movement. Comparison results have shown that neural network models outperform the regression model applied by Port and Maritime Board. Zhang and Cui (2011) built a combination forecast model based on Elman neural network for forecasting the container throughput. Empirical analysis based on container flows in Shanghai port, demonstrates the accuracy of developed model. Diaz et al. (2011) evaluated Winters, Toiga group and United Nations methods for forecasting empty container volumes at container ports. Forecasting abilities of the three methods were tested in forecasting empty container volumes for the US container ports—Port of Long Beach, Port of Los Angeles and Port of Savannah. The evaluations found that the Winters method is a more accurate forecaster of empty container volumes than Toiga group and United Nations methods. Xie et al. (2013) combined traditional SARIMA, classical decomposition (CD) and structural decomposition (SD) methods with advanced least square support vector regression (LSSVR) model and proposed: SARIMA-LSSVR, SD-LSSVR and CD-LSSVR. In order to verify the effectiveness of the proposed hybrid approaches authors used the time series of container throughput at Shanghai Port and Shenzen Port. The proposed hybrid approaches were compared with each other and with individual approaches and the results suggested that they can achieve better forecasting performance. Huang et al. (2015) proposed a hybrid forecasting model combining projection pursuit regression (PPR) and genetic programming algorithm for forecasting container throughput in Qingdao Port. Results have shown that the proposed model significantly outperforms ANN, SARIMA and PPR models. Xiao et al. (2014) proposed a transfer forecasting model guided by discrete particle swarm optimization algorithm (TF-DPSO). Proposed model was tested on container throughput time series of Shanghai Port and Ningbo Port in China and the results have shown the effectiveness of proposed model. The model outperforms traditional analog complexing (AC) model, ARIMA modela and Elman network (ENN) model. Gao et al. (2016) demonstrated the application of six commonly used criteria for model selection (AIC, BIC, HQC, Cp, LooCu and soCV) and compared six frequentist model averaging (FMA) methods (S-AIC, S-BIC, S-HQC, JMA, LsoMA and AFTER) for monthly container port throughput forecasting. Results have shown that the LsoMA method is the best in the sense of achieving the lowest average of mean squared forecast errors. Moscoso-Lopez et al. (2016) proposed and compared different approaches based on artificial neural networks

and support vector machines for modeling the Ro–Ro transport flows. The performance of the models is evaluated on real data from Ro–Ro freight transportation in the Port of Algeciras Bay. It was concluded that the values obtained with the SVM models are fitted better with the observed values than the values obtained with ANNs. Pang and Gebka (2017) proposed a novel approach for forecasting the total port container throughput. In place of forecasting the total throughput directly they suggested to utilize data from individual port´s terminals, both individually and with account for interdependencies between terminals throughputs to obtain the aggregate total port´s throughput. Authors forecasted the demand for total container throughput at the Indonesia´s largest seaport employing SARIMA, the additive and multiplicative seasonal Holt–Winters (MSHW) and vector error correction model (VECM). MSHW model generated the most accurate forecasts of total container throughput whereas VECM provided the best model fits and forecasts for individual terminals. Twrdy and Batista (2016) applied a set of dynamic models to forecast container throughput in the North Adriatic ports of Koper, Trieste, Venice, Ravenna and Rijeka. Models applied are Markov-chain annual growth rate model, a time series trend model with periodical terms and the gray system model. Lee et al. (2017) provided a practical approach for estimating the market share and cargo volume through the seaports in South Korea. Proposed approach based on a combining a port choice and ARIMA model and tested under various port development scenarios demonstrates practical usability.

Xie et al. (2017) proposed a procedure of data characteristic analysis (DCA) and model selection within a decomposition-ensemble methodology which includes data decomposition, DCA of components, individual prediction and ensemble prediction. Based on the time series of container throughput at the Port of Singapore and the Port of Los Angeles the proposed approaches are illustrated and empirically compared with selected benchmark methods. Mo et al. (2018) proposed a hybrid forecasting model based on decomposition the container throughput time series into linear and nonlinear parts. The linear part is predicted by the SARIMA model. The model adopts three nonlinear single models, Support Vector Regression (SVR), Back Propagation (BP) neural network and Genetic Programming (GP) to predict the residual subseries. Then the model establishes selective combination by the improved Group Method of Data Handling (GMDH) neural network on the nonlinear subseries and obtains its combination forecasting results. The predictions of two parts, linear and nonlinear are integrated to obtain the forecasting results of the original container throughput time series. Niu et al. (2018) built a hybrid decomposition-ensemble model named VMD–ARIMA–HGWO–SVR for the purpose of improving the stability and accuracy of container throughput prediction. Variational mode decomposition (VMD) algorithm is employed to decompose original series in several modes, ARIMA models are built to forecast the low frequency components and the high frequency components are predicted by Support Vector Regression (SVR) which are optimized by hybridizing grey wolf optimization (HGWO). Historical container throughput data of Singapore port and Shanghai port are used for experimental simulations.

Based on detailed and comprehensive review of literature we may draw following conclusions:

- The problem of container flow estimation and forecasting represents a very important problem that may have significant implications on a broad set of actions within the port and also the port hinterland.
- The significance of the problem implied the development and testing a huge diversity of various techniques (parametric as well as non-parametric) for container flow assessment and prediction. Among the parametric techniques some researches (Peng and Chu 2009; Schulze and Prinz 2009) demonstrated the superiority of ARIMA (SARIMA) models, and that was the reason for selecting this kind of models as a benchmark against the developed technique. On the other side, some other research papers (like Tsai and Huang (2015) and Gosasang et al. (2011) for example) clearly emphasized the potential of neural networks for container throughput modelling which motivated authors to consider the possibility for improvement of existing ANN based container flow modelling approaches.
- However, all ANN based past research efforts (in the domain of container flow forecasting) treat the construction of neural network structure without finding the optimal neural architecture. Standard multi-layer ANNs are assumed with predefined functional form and time consuming experiments with alternative architectures are usually performed. This may lead to inappropriately designed network structure. Networks smaller than needed would be unable to learn, on the other side, networks larger than needed would probably end in over fitting the training samples. Therefore, determining an appropriate architecture of ANN for a particular problem is an important issue since the network topology directly affects its computational complexity and its generalization capability.

Based on these research findings we proposed a new and comprehensive methodology for port container flow forecasting (which also can be used on other traffic flow forecasting problems). Novelty is reflected in the fact, that as far as the knowledge of the authors goes, none of the existing literature has investigated and developed a new general framework for container throughput forecasting based on a fuzzy ANN (FANN) combined with metaheuristic algorithms. Therefore, in order to overcome the difficulty in ANN design architecture, and to make container flow forecasting more accurate, we hybridized FANN, with two very known metaheuristics, genetic algorithms and simulated annealing, respectively. Comprehensiveness of this approach is based on separate analysis, assessment and forecasting of total, loaded, unloaded, empty and transit container flows which generates a more detailed input to the port managers as a support for solving a diversity of decision making problems.

3 Methodology

This research evaluates two methods to evolve neural networks architectures, one carried out with genetic algorithm (GA) and second one carried out with differential simulated annealing (SA) algorithm (Michalewicz and Fogel 2000; Wu et al. 2008; Soares et al. 2013). Work in this area demonstrated that applying evolutionary optimization and simulated annealing can replace trial and error

methods to determine optimal ANN architecture and increase the speed, accuracy and efficiency of the training phase. Sexton et al. (1999) proposed simulated annealing and genetic algorithm for optimizing neural networks and compared their performances. According to the results of their research genetic algorithm is superior to simulated annealing for optimizing neural networks. Blanco et al. (2000) and Abdalla et al. (2014) applied GA for determining the optimal architecture and minimum number of required neurons in the ANN. Arifovic and Gencay (2001) used GA to determine the optimal ANN structure. Liao and Tsao (2006) proposed the chaos search genetic algorithm and simulated annealing method to tune the parameters of fuzzy neural networks for power-system load forecasting. The results of the research of Kaviani and MirRokni (2017) showed that the GA based approach for determining optimal state of ANN architecture may lead to increased speed, accuracy and efficiency of the training. Ghasemiyeh et al. (2017) proposed hybrid ANN models and metaheuristic algorithms to predict prices on stock exchange. According to statistical performances particle swarm optimization was a dominant metaheuristic approach that produced the best ANN architecture. Inspired by these research efforts we proposed a hybrid fuzzy ANN approach for container throughput volume forecasting. The approach is described in Sect. 3.1. Developed approach is compared with traditional ARIMA models which are briefly described in Sect. 3.2. Flow chart on Fig. 1 summarizes methodology applied in this paper. Alternative models are tested on a training data sample and their optimal configurations are chosen and compared on a test data sample. Fuzzy neural networks based metaheuristics include model initialization (expert knowledge matrix design), candidate models identification (neural fuzzy prediction network) and evaluation of candidate models (best network selected by GA-FANN or SA-FANN). Model which has the best performances for considered time series can be used for forecasting of container flow throughput in next period. ARIMA modelling follows Box and Jenkins methodology (Box et al. 2008) that includes model identification, estimation of parameters and model validation.
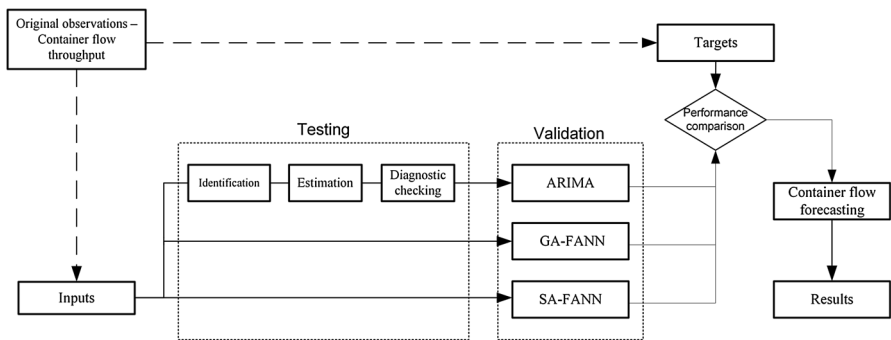

Fig. 1 Comparative analysis of ARIMA, GA-FANN and SA-FANN methods for container flow modelling

3.1 Fuzzy neural network based metaheuristics for time series analysis

3.1.1 Artificial neural networks and fuzzy neural networks

(a)   Artificial neural networks

Artificial Neural Networks (ANNs) represent computer programs whose main purpose is to simulate the way in which human brain processes information. ANNs learn (or are trained) through experience with appropriate learning exemplars and gather their knowledge by detecting the patterns and relationships in data (Agatonovic and Beresford 2000). ANNs are composed from simple interconnected processing elements, known as neurons under a defined topology or layers. Each neuron is connected to its neighbors with varying coefficients called weights in which the knowledge of ANN is stored (Ghasemiyeh et al. 2017). Neural networks can automatically adjust their weights to optimize their behavior as pattern recognizers, decision makers, system controllers, predictors, etc. Adaptivity allows the neural network to perform well even when the environment or the system being controlled varies over time. Besides their good capability in pattern recognition, ANNs have a limitation reflected mainly in explaining how they reach their decisions. In order to overcome this limitation, we combined ANNs with fuzzy logic.

(b)   Fuzzy neural networks

Fuzzy artificial neural networks (FANNs) represent a layered, feedforward network that handles fuzzy set signals and/or has fuzzy set weights. FANNs implement fuzzy if–then conditional statements in a constructive way. Though they cannot use the standard error back propagation algorithm for learning directly, they can be trained by steepest descent methods to learn the parameters of the membership functions which represent the linguistic terms in the rules (Herrera and Martinez 2000; Kosko 1992). In this paper a regular fuzzy neural network was used. It represents a neural network with fuzzy signals and sigmoidal transfer function. All operations are defined by Zadeh's extension principle (Zadeh 1965). The fuzzy neural net considered in this paper is presented in Fig. 2.

It is a one hidden layer network with constant structure of input and output layers. Input layer contains a number of neurons (p) that corresponds to realized container flow (daily, monthly) observations in considered time series (total, transit, loaded, unloaded, empty), say period $t - p$ to period $t - 1$. The output layer contains a single neuron which represents the container flows in period t. The approach applied in this paper is based on utilization of fuzzy if–then rules for learning of neural networks. The learning algorithm is summarized as follows (the details can be found in Ishibuchi et al. 1993a, b, c):

Each input x_i interacts with the weight w_i to produce the product $P_i = w_i x_i$, $i = 1,…,n$ where we use the extension principle to compute P_i. The input information P_i is aggregated, by standard extended addition, to produce the input:
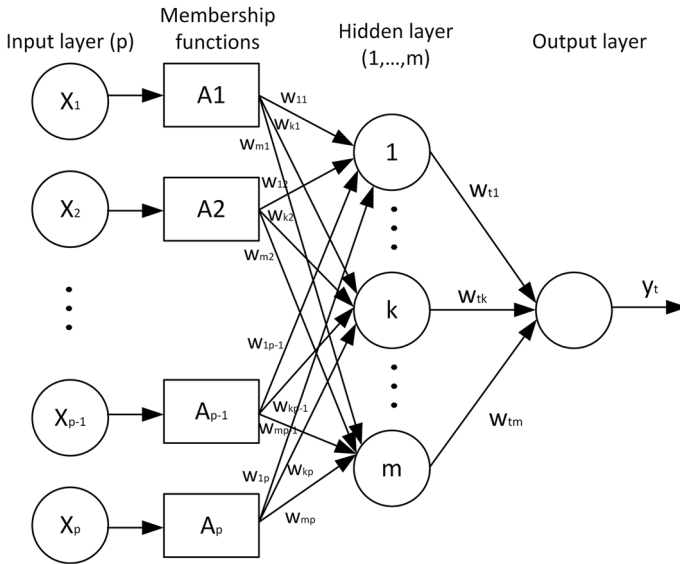
Fig. 2 Simple regular fuzzy neural net

$$\text{net} = P_1 + \cdots + P_n = w_1 \, x_1 + \cdots + w_n \, x_n \tag{1}$$

to the neuron. The neuron uses its transfer function f, which is a sigmoidal function, to compute the output:

$$y = f(\text{net}) = f(w_1 \, x_1 + \cdots + w_n \, x_n) \tag{2}$$

where f is a sigmoidal function. The membership function of the output fuzzy set y is computed by the extension principle. This approach uses following fuzzy fuzzy rules:

$$R_i : \quad \text{if} \quad x \text{ is } A_i \text{ then } \quad y \text{ is } B_i, \quad 1 \leq i \leq n \tag{3}$$

Given some data on x, say A′, the fuzzy expert system comes up with its final conclusion y is B′. Let's define $[\alpha_1, \alpha_2]$ that contains the support of all the A_i, plus the support of all the A′ we might have as input to the system. Also, let's assume that $[\beta_1, \beta_2]$ contains the support of all the B_i, plus the support of all the B′ that we can obtain as outputs from the system. Let $M \geq 2$ and $N \geq M$ be positive integers. Then,

$$x_j = \alpha_1 + (j-1)(\alpha_2 - \alpha_1)/(M-1) \tag{4}$$

for $1 \leq j \leq M$, and

$$y_i = \beta_1 + (i-1)(\beta_2 - \beta_1)/(N-1) \tag{5}$$

for $1 \leq i \leq N$. The exact version of the system (Ishibuchi et al. 1993a, b, c, 1994) is obtained by defuzzification of $B'(i), i = 1, \ldots, N$ to obtain output $b' = (B'(y_1), \ldots, B'(y_N))$ for every input

$$a' = (A'(x_1), \ldots, A'(x_M)) \tag{6}$$

In order to improve computational performance of proposed FANNs, genetic algorithm and simulated annealing, as the main representatives of multi-point searching methods are applied for more accurate learning the relations of fuzzy inputs and outputs.

3.1.2 Genetic algorithm

Genetic algorithm (GA) represents a stochastic global search technique that solves problems by imitating the processes observed during natural evolution (Ghasemiyeh et al. 2017). It decreases the effect of original values greatly through crossover and mutation operations, and it can easily find the global optimal results (Donate et al. 2011). With genetic algorithm, a population of candidate solutions (called individuals or phenotypes) to an optimization problem is continuously evolving toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered. Traditionally, solutions are represented in binary form as strings of 0 s and 1 s, however, other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated. The fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been reached, or a satisfactory fitness level has been obtained for the population (Ghosh et al. 2005; Yan 2012; Glisovic et al. 2015).

In this paper, we applied GA in order to determine the optimal architecture of fuzzy neural network. The main string (as the set of genes which represent one individual or number of neurons in case of this application) is composed of binary substrings which correspond to the neurons in hidden layer. All models from the training set are successively conveyed to the neural network input (Denai et al. 2007). The response of the neural network to the input data is calculated and the difference between the actual and the desired value of forecasted container flow is used to evaluate the error function over all models. This error represents the chromosome quality. GA parameters (mutation, fitness function and crossover) were varied in order to generate optimal values. The uniform crossing is used—every bit of descendants is with the probability 0.7 taken from one of the parents. Probability mutation is 0.05. One-point crossover was used (MacKay 2003). Pseudo code of GA-FANN algorithm developed for the purpose of container flow forecasting looks as follows:

If NetP$_n$ are parent chromosomes and netC$_n$ are daughter chromosomes at the nth iteration then pseudo GA-FANN is:

Initialize iteration it=0;
Specify the initial value of netP$_{it}$;
Estimate netP$_{it}$ using the fitness function;
while not (termination conditions) do
{
*   Cross netP$_{it}$ to produce netC$_{it}$;*
*   Mutate netC$_{it}$;*
*   Estimate netC$_{it}$ using the fitness function;*
*   Select netP$_{it+1}$ from netP$_{it}$ and netC$_{it}$;*
*   it= it+1;*
}

For given GA parameters, the initial ANN architecture is developed. Then the phasification is performed and thus obtained *netP$_{it}$* for the network defined on the randomly selected input data belonging to a particular node to the hidden layer. *netC$_{it}$* represents a new population obtained from old population according *netP$_{it}$* according to their fitness.

3.1.3 Simulated annealing

Simulated Annealing (SA) algorithm represents a heuristic search algorithm which is widely used in a variety of combinatorial optimization problems (Szu and Hartley 1987; Lee et al. 2008). SA algorithm mimics the solid cooling process. It begins in a high initial temperature T and gradually decreased, to optimize the results by iteration (Chibante 2010). Updated conditions during the optimization process for the solution is the objective function whose value is better than the original value.

A measure of network performance is given by the "energy" as follows:

$$E = \sum_j \left[t_j - O_j \right]^2 \tag{7}$$

t_j is the target output value, O_j is predicted value. The weight accepts the change with probability:

$$P = \exp \left[-(E_{new} - E_{old})/T \right] \tag{8}$$

if $E_{new} > E_{old}$, and with probability $P = 1$ if the change decreases the energy. The parameter T is initially set at some relatively large value, and then decreased according to the prescription:

$$T \rightarrow \gamma T, \ \gamma < 1 \tag{9}$$

after a fixed number of sweeps N_{sw}.

When the temperature T is set to zero from the start, the annealing algorithm reduces to a kind of iterated improvement. Changes in the network are accepted only if they decrease the energy, otherwise the network remains unchanged. In

this research cooling parameter is $\gamma = 0.9$ and the starting temperature is $T_0 = 100$ (Chibante 2010).

While simulated annealing is slower than other training algorithms, it outperforms greedy methods and remains popular when minimum error in the trained network is critical. SA algorithm serves as an alternative way to GA for developing an optimal FANN architecture. In case of this algorithm, every change of number of neurons in hidden layer has been executed separately. Current solution represents a random combination of input neurons or past observations of container flows) which are used for hidden layer/neurons. The basic idea of the method is to generate a random neural network configuration (trial point) iteratively through perturbation, and evaluate the objective function and the constraints after determining the state variables by using the simulator. The method uses temperature and other annealing parameters by trial and error to attain near-optimal solutions. Pseudo code of developed SA-FANN algorithm looks as follows:

Initialize a vector Woldof the weights of the Neural Network
Fopt = F(Wopt)
T= Tmax
Tmin, iter = 0, maxIter
while(T>Tmin) {
 while(iter<maxIter){
 Wnew = newOf (Wold)
 if(Δf=F(Wnew)-F(Wold)<0) then{
  If(F(Wnew)>F(Wold)) then{
   Wold=Wnew
   actualize Fold
  }
 } else
  If($E_{new} > E_{old}$)then
  P=1
  else
  {
   $$P = \exp\left[-(E_{new} - E_{old})\,/\,T\right]$$
   Wold = Wnew
  }
 *} T=T*γ*
}

For given SA parameters, the initialization of ANN parameters (architectures) is performed. After that the phasification follows and thus the *netP_{it}* is obtained for the network defined on the randomly selected input data belonging to a particular node to the hidden layer. *Wold* represents the weights of *netP_{it}*.

3.2 ARIMA and SARIMA models

ARIMA method represents one of the most popular parametric univariate time series modelling techniques. It is composed from Auto Regressive (AR) model, Moving Average (MA) model and combination of AR and MA, ARMA models (Suhartono 2011). AR model includes lagged terms on the time series itself and MA model includes lagged terms on the noise or residuals (Milenkovic et al. 2016). For applying the ARIMA models, it is required that the time series data to be stationary or can be differenced to make them stationary. The letter "I" (Integrated) means that the first order difference is applied in order to make the given time series stationary.

The equation representing an ARIMA(p, d, q) model for time sequence Y_t can be expressed as:

$$\phi_p(B)(1 - B)^d \, Y_t = \theta_q(B)\varepsilon_t \tag{10}$$

where p is the order of AR process, d is the order of differencing and q is the order of MA process. ε_t is a white noise sequence assumed to be normal random variable with zero mean and variance σ^2. B is the backshift operator, whose effect on a time series Y_t can be summarized as $B^d \, Y_t = Y_{t-d}$. By extending the ARIMA model to incorporate seasonal variation in time series a SARIMA model is obtained that can be formulated as follows (Smith et al. 2002):

$$\phi_p(B)\Phi_P(B^s)(1 - B)^d(1 - B^s)^D \, Y_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t \tag{11}$$

Notation is represented by SARIMA(p, d, q) $\times$ (P, D, Q)$_s$ where Φ and Θ are the seasonal ARMA coefficients and seasonal differencing operator $(1 - B^s)^D$ of order D is used to eliminate seasonal patterns.

Fitting the ARIMA and SARIMA models involves model identification, estimation of parameters and validation (Box et al. 2008; Ruiz-Aguilar et al. 2014; Taskaya-Temizel and Casey 2005). By analyzing the behavior of the autocorrelation function (ACF) and partial autocorrelation function (PACF) appropriate model can be identified. To cope with subjectivity present in this step and to improve the determination of the final orders of the ARMA, Akaike Information Criteria (AIC), Bayesian Information Criterion (BIC) and normalized version of BIC have been used. Estimation step includes fitting the model to the time series and estimation of parameters. This is done by using the conditional sum of squares or maximum likelihood method. The last step, validation of selected model, is performed within the diagnostic checking by analysis of stationarity, invertibility and the presence of redundancy in model parameters (Milenkovic et al. 2016).

4 Application and results

Port of Barcelona (PoB) realized a cargo throughput of 48.6 million tons in 2016. Barcelona is after Valencia and Algeciras, the third largest container port in Spain. With a volume of 2.2 million TEU, Barcelona is ninth largest container port in Europe. The port handles four different types of cargo, of which containers is the most important with a share of more than 66%. The other main cargo types are liquid bulk, cars and dry bulk. Since this study focuses on the observation and prediction of 20 feet containers throughput time series, the historical data for this type of cargo is presented on Fig. 3. The time series data are obtained from website of PoB (Port of Barcelona 2017). The sample data are monthly observations of container throughput covering the period from January 2010 to December 2016 with a total of 84 observations. First 72 monthly observations are used as a training data set whereas the remaining 12 observations served for verification the forecasting capability of selected models. All components of container traffic—transit, loaded, unloaded and empty, together with total container traffic were independently investigated and appropriate models estimated.

In this study, nonlinear SA-FANN and GA-FANN models are developed and compared with traditional linear ARIMA model are used for predicting the volume of container flows in Port of Barcelona. SA-FANN and GA-FANN are implemented in C# programming language. The parameters used by SA and GA for each of the time series are the same and are presented in Table 1. ARIMA is implemented by the use of R software package. Following sections present the forecasting results of applied methods.
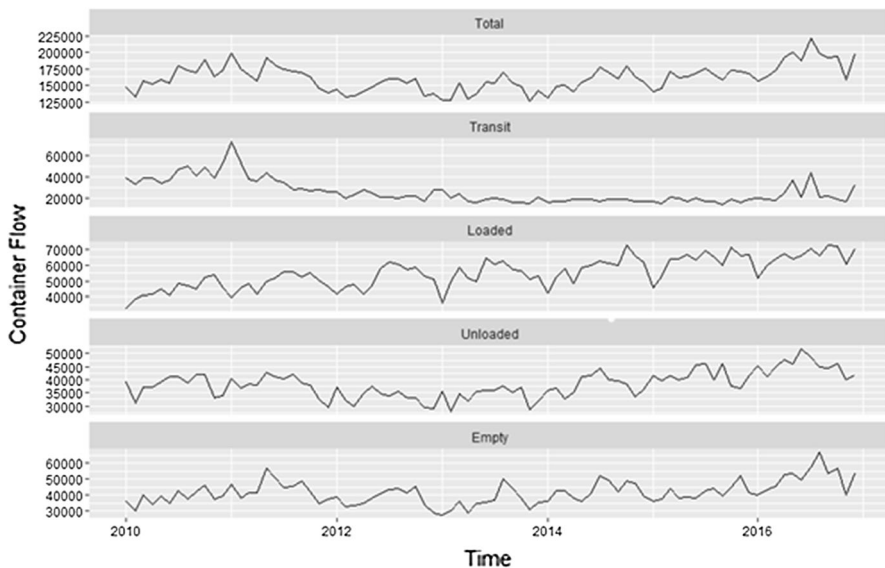

Fig. 3 Time series of monthly container flows for Port of Barcelona (January 2010–December 2016)

Table 1 Parameter description

Genetic algorithm parameters		Simulated annealing parameters	
Population size	500	Initial temperature	1000
Crossover probability	0.7	Cooling parameter	0.9
Mutation probability	0.05	Epoch length	500
		Stopping condition (s)	10

4.1 Total container flow

(a) SA-FANN and GA-FANN results

SA-FANN generates the best topology with 7 neurons in hidden layer whereas the SA-FANN has 9 neurons in middle layers (Table 2). Table 2 summarizes the results obtained by differing the number of neurons in the hidden layer for total container flow time series. The stopping criteria was the architecture of the network for which the respective errors were the smallest (bold values).

For the total container flow, and all the subsequent time series we have also conducted sensitivity analysis. The analysis of model sensitivity can show that the simulation results are more sensitive to some parameters than others. This study uses a single-value sensitivity index to evaluate the effect degree of parameters of FANN on simulation results. The index is defined as:

$$S = \frac{O_2 - O_1}{I_2 - I_1} \cdot \frac{I_{ave}}{O_{ave}} \tag{12}$$

Table 2 Varying number of neurons in hidden layer for total container flow

Neurons in middle layer	SA-FANN			GA-FANN		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
1	2464.27	1.35	3324.63	2700.84	1.48	4922.91
2	2547.5	1.39	3353.57	2244.67	1.25	2873.22
3	2630.83	1.44	3394.58	2328	1.3	2912.71
4	2380.33	1.3	3124.1	2449.67	1.38	3161.36
5	2234.58	1.22	3002.67	2283	1.29	3101.57
6	2151.25	1.18	2975.63	1949.67	1.11	2558.59
7	**2072.5**	**1.15**	**2597.08**	1933	1.1	2557.74
8	2119.75	1.16	2959.91	1902.17	1.09	2545.66
9	2177.25	1.19	3079.81	**1749.17**	**0.98**	**2280.73**
10	2260.58	1.24	3105.94	1885.5	1.08	2543.56

Table 3 Scenarios for sensitivity analysis

Parameters of FANN	Numbers of scenarios	Scenarios
Training times	3	100, 300, 500
Learning velocity	2	0.01, 0.0001

where S is the sensitivity index; I_1 and I_2 are the smallest and the largest input values respectively; O_1 and O_2 are the model output values corresponding to I_1 and I_2 respectively; I_{ave} and O_{ave} are the average I_1 and I_2 and the average O_1 and O_2, respectively. The greater the absolute value of S, the greater the effect an input parameter has on a particular output.

Number of scenarios and comparison of different scenarios with various training times and scenarios with various learning velocities for total container flow time series are presented in Table 3. Table 3 shows the scenarios with different parameters of FANN for sensitivity analysis, 3 training times, 2 learning velocities. When using the base mentioned above, the R-square (R^2) is measured for parameter training and validation. The result shows that statistical models, such as FANN technique, can be a flexible tool for the prediction of container flows.

To identify the sensitivity of parameters, the sensitivity index sensitivity, Eq. (12), is useful, although this index cannot be applied for the scenarios with different transfer functions. Table 4 lists the values of sensitivity index for parameter "training times" and "learning velocity". A higher sensitivity index value shows a higher sensitivity of these parameters for which it is counted. In this case the simulation results shown are more sensitive to parameter "training times" than parameter "learning velocity". It is significant to comprehend the parameters of FANN to minimize waste of time in the process of parameter training.

(b) ARIMA results

From the historical data for total container flow (Fig. 1) it can be observed that time series shows nonstationary character with strong seasonality pattern and increasing trend. Long term trend is proven by very sharp decrease of autocorrelation values in first four lags (0.64, 0.51, 0.41, 0.30). Annual lag also reported significant autocorrelation (0.24) which points on existence of seasonality (Fig. 2). For additional proving of seasonality, we applied TBATS estimation procedure for identifying and handling a variety of seasonal patterns which relies on a Trigonometric functions, Box–Cox transform, ARMA errors, Trend and Seasonal components (De Livera et al. 2010). Augmented Dickey Fuller (ADF) test also gives very high p value (0.8846).

Therefore, in order to eliminate trend and seasonal pattern, first lag difference and seasonal difference terms are included in model. ADF test of new time series gives p value equal to 0.01 which means that the new differenced and seasonally differenced time series is stationary. Based on these conclusions SARIMA$(p, 1, q) \times (P, 1, Q)_{12}$

Table 4 Prediction accuracy analysis for different parameters of FANN in case of total container flow time series

Comparison of different scenarios with various training times and scenarios with various learning velocities				Sensitivity analysis by using sensitivity index	
GA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.329162
R^2 (Parameter training)	0.87	0.87	0.91	Learning velocity	0.271981
R^2 (Parameter validation)	0.74	0.78	0.83		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.88	0.9	0.93		
R^2 (Parameter validation)	0.87	0.88	0.9		
SA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.301163
R^2 (Parameter training)	0.82	0.83	0.9	Learning velocity	0.259121
R^2 (Parameter validation)	0.67	0.68	0.8		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.85	0.89	0.9		
R^2 (Parameter validation)	0.77	0.84	0.89		

was selected as the basic structure of alternative SARIMA models. Among a number of alternative statistical models SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$ was chosen as the most appropriate with the lowest AICc of 1295.02 and Mean Absolute Percent Error (MAPE) of 4.64. 69.2% of the variance of time series was covered by this model (Table 5). No outliers were identified by fitting this model. The model has the following equation:

$$(1 - 0.22B)(1 - 0.14B^{12})(1 - B)(1 - B^{12})Y_t = (1 + 0.62B)(1 + 0.99B^{12})\varepsilon_t$$
(13)

Diagnostic checks prove correct specification of SARIMA model. According to Ljung–Box Q-test and p value (Ljung–Box Q = 11.1 and p value = 0.80) residuals

Table 5 Akaike Information Criterion with correction (AICc), mean absolute percentage error (MAPE) and stationary R^2 of SARIMA models for total container flow

Models	MAPE	AICc	Stationary R^2
SARIMA$(0, 1, 1) \times (0, 1, 1)_{12}$	5.04	1295.51	0.63
SARIMA$(0, 1, 1) \times (0, 1, 0)_{12}$	6.38	1303.05	0.57
SARIMA$(0, 1, 0) \times (0, 1, 1)_{12}$	5.59	1296.25	0.60
SARIMA$(1, 1, 1) \times (0, 1, 1)_{12}$	4.99	1295.66	0.64
SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$	4.64	1295.02	0.69

were white noise and therefore there is no significant autocorrelation between residuals at different lag times.

We have also conducted an additional analysis. Namely, we applied the Box–Jenkins procedure on the whole data set. In this case SARIMA$(0, 1, 1) \times (0, 1, 2)_{12}$ was selected as the best model. It reported slightly lower AICc (1559.4) then SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$ in case of applying it on the whole data set (AICc = 1561.28).

4.2 Transit container flow

(a) SA-FANN and GA-FANN results

In case of transit container flows topology with best results has 6 neurons in the middle layers for SA-FANN whereas for GA-FANN it has 7 neurons in middle layers. Table 6 summarizes the results obtained by differing the number of neurons in the hidden layer for transit container flow time series. As in previous case, the stopping criteria was the architecture of the network for which the respective errors were the smallest (bold values).

Values for each scenario and sensitivity (Table 2) for transit container flow time series are given in Table 7. In this case the simulation results shown are more sensitive to parameter "training times" than to parameter "learning velocity".

(b) ARIMA results

Historical data for transit container flow (Fig. 1) exhibit nonstationary character without seasonality pattern. ADF test of differenced time series gives p value equal to 0.01 which means that the new time series is stationary. In this case

Table 6 Varying number of neurons in hidden layer for transit container flow

Neurons in middle layer	SA-FANN			GA-FANN		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
1	1261	4.7	1755.83	1115.5	4.78	1353.44
2	1217.67	4.58	1743.45	1222.17	5.24	1480.3
3	1344.33	5.1	1834.65	1151.5	5.02	1456.24
4	1177.67	4.72	1613.88	1126.5	4.88	1445.24
5	1094.33	4.46	1538.65	1138.83	4.9	1361.17
6	**1011**	**3.7**	**1342.18**	946.33	4.17	1210.33
7	1269.33	5.03	1735.01	**828.67**	**3.31**	**1096.11**
8	1102.67	4.36	1410.41	863	3.83	1061.32
9	1019.33	3.96	1306.24	1160.33	5.05	1456.9
10	1077.67	4.09	1377.28	979.67	4.08	1302.67

Table 7 Prediction accuracy analysis for different parameters of FANN in case of transit container flow time series

Comparison of different scenarios with various training times and scenarios with various learning velocities				Sensitivity analysis by using sensitivity index	
GA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.330122
R^2 (Parameter training)	0.86	0.87	0.9	Learning velocity	0.279989
R^2 (Parameter validation)	0.73	0.75	0.82		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.88	0.89	0.92		
R^2 (Parameter validation)	0.86	0.87	0.9		
SA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.319933
R^2 (Parameter training)	0.8	0.82	0.9	Learning velocity	0.26112
R^2 (Parameter validation)	0.61	0.67	0.79		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.86	0.9	0.91		
R^2 (Parameter validation)	0.79	0.87	0.87		

Table 8 Akaike Information Criterion with correction (AICc), mean absolute percentage error (MAPE) and stationary R^2 of SARIMA models for transit container flow

Models	MAPE	AICc	Stationary R^2
ARIMA(1,1,2)	13.38	1429.99	0.79
ARIMA(2,1,1)	13.03	1426.14	0.80
ARIMA(1,1,1)	13.12	1429.13	0.78
ARIMA(2,1,2)	12.78	1426.08	0.81

ARIMA(p, 1, q) was selected as the basic structure. Based on observations from correlograms, we started from ARIMA(1, 1, 2) model and compared it with similar configurations based on AICc criteria. The best model (lowest AICc) which shows also statistically insignificant evidence of non-zero autocorrelation in the residuals is ARIMA(2, 1, 2) model (Table 8):

$$(1 - 0.66B + 0.61B^2)(1 - B)Y_t = (1 - 0.99B + 0.56B^2)\varepsilon_t \qquad (14)$$

Further, if we consider the presence of outliers in selected ARIMA model we may observe following four outliers: Three temporary change outliers (June 2010, December 2010 and December 2012) and an additive outlier (January 2011). Therefore, we implement a procedure described by Chen and Liu (1993) for

identifying the best ARIMA model taking into account presence of outliers. The procedure consists from two main stages:

- Detection of outliers upon a chosen ARIMA model;
- Selection and/or refit of ARIMA model including the outliers detected in the previous step and removing those outliers that are not significant in the new fit.

After a series of iterations, ARIMA(1, 1, 1) (AICc = 1362.35) is selected and the series is adjusted for the detected outliers. The model with explicitly incorporated outlier effects looks like follows:

$$
\begin{aligned}
Y_t = {} & 9892 \frac{1}{1 - B} I_t(7) + 29353 \frac{1}{1 - 0.7B} I_t(13) - 17251 \frac{1}{1 - B} I_t(14) \\
& + 8710 \frac{1}{1 - 0.7B} I_t(36) + \frac{(1 + 0.62B)}{(1 + 0.32B)(1 - B)} \varepsilon_t
\end{aligned}
\tag{15}
$$

Additionally, when we take the full data set for model estimation we obtain ARIMA(1, 1, 0) as the best model (AICc = 1596.4). In this case, following seven outliers were detected: two level shift outliers (July 2010, March 2011), two additive outliers (January 2011, July 2016), temporary change outlier (December 2012), two innovational outliers (May 2016 and December 2016). The model has the following equation:

$$
\begin{aligned}
Y_t = {} & 12336 \frac{1}{1 - B} I_t(7) + 25733 I_t(13) + 25855 I_t(79) - 12964 \frac{1}{1 - B} I_t(15) \\
& + 10382 \frac{1}{1 - 0.7B} I_t(36) + 12094 \frac{1}{(1 + 0.59B)(1 - B)} I_t(77) \\
& + 13755 \frac{1}{(1 + 0.59B)(1 - B)} I_t(84) + \frac{1}{(1 + 0.59B)(1 - B)} \varepsilon_t
\end{aligned}
\tag{16}
$$

4.3 Loaded container flow

(a) SA-FANN and GA-FANN results

The results obtained by differing the number of neurons in the hidden layer for loaded container flow time series are given in Table 9. In case of SA-FANN topology with the best results for loaded container flows has 10 neurons in the middle layer. GA-FANN produces best results with 9 neurons in the middle layers. Criteria for this selection were the minimum values of MAE, MAPE and RMSE (bold values in the Table 9).

In next table the values for each scenario and sensitivity (Table 2) for transit loaded flow time series are given. The simulation results are more sensitive to parameter "training times" than to parameter "learning velocity" (Table 10).

Table 9 Varying number of neurons in hidden layer for loaded container flow

Neurons in middle layer	SA-FANN			GA-FANN		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
1	781.75	1.23	932.05	965.75	1.5	1092.79
2	865.08	1.37	986.17	954.58	1.48	1044.68
3	823.41	1.31	885.13	1046.25	1.61	1129.02
4	898.42	1.41	982.93	962.91	1.49	1072.39
5	790.08	1.21	870.43	915.75	1.44	1044.66
6	745.53	1.14	844.48	907.42	1.43	1022.19
7	703.91	1.07	841.26	949.08	1.5	1203.43
8	749.92	1.14	868.17	907.41	1.44	1174.18
9	699.92	1.07	840.31	**896.25**	**1.4**	**964.97**
10	**698.42**	**1.11**	**796.79**	904.58	1.41	974.19

(b) ARIMA results

Time series for loaded container flow is characterized by an increasing trend and seasonal pattern. The trend is characterized by significant autocorrelation values on first four lags (0.682, 0.505, 0.415, 0.342). Annual lags and its multiplies also have significant values. Therefore, in order to eliminate trend and seasonality, the first differencing (d = 1) and seasonal differencing

Table 10 Prediction accuracy analysis for different parameters of FANN in case of loaded container flow time series

Comparison of different scenarios with various training times and scenarios with various learning velocities				Sensitivity analysis by using sensitivity index	
GA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.290107
R^2 (Parameter training)	0.89	0.89	0.9	Learning velocity	0.289989
R^2 (Parameter validation)	0.84	0.86	0.87		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.89	0.9	0.9		
R^2 (Parameter validation)	0.87	0.87	0.88		
SA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.373013
R^2 (Parameter training)	0.9	0.9	0.93	Learning velocity	0.31899
R^2 (Parameter validation)	0.87	0.88	0.9		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.91	0.93	0.94		
R^2 (Parameter validation)	0.89	0.9	0.92		

Table 11 Akaike Information Criterion with correction (AICc), mean absolute percentage error (MAPE) and stationary R^2 of SARIMA models for loaded container flow

Models	MAPE	AICc	Stationary R^2
SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$	4.22	1166.26	0.8724
SARIMA$(1, 1, 2) \times (1, 1, 2)_{12}$	4.25	1166.43	0.8733
SARIMA$(1, 1, 1) \times (1, 1, 2)_{12}$	4.23	1164.22	0.8722
SARIMA$(2, 1, 1) \times (2, 1, 2)_{12}$	4.22	1168.19	0.8760
SARIMA$(1, 1, 3) \times (1, 1, 1)_{12}$	4.21	1161.83	0.8740

$(S = 12, D = 1)$ are included into the model. Then, in this case we may suggest SARIMA$(p, 1, q) \times (P, 1, Q)_{12}$ model structure. Among the set of selected alternative configurations SARIMA$(1, 1, 3) \times (1, 1, 1)_{12}$ was selected as the most appropriate (Table 11). There are no identified outliers in this time series. The model looks as follows:

$$(1 + 0.71B)(1 - 0.30B^{12})(1 - B)(1 - B^{12})Y_t = (1 + 0.07B - 0.76B^2 - 0.30B^3)(1 - B^{12})\varepsilon_t$$

(17)

Selected SARIMA model appears to be stationary and invertible without redundant parameters. Ljung–Box Q-test and p value (Ljung–Box Q = 10.718, p value = 0.8266) prove that residuals are white nose and therefore there is no significant autocorrelation between residuals at different lags. Coefficient of determination of selected model is 0.87. If we use the full dataset for model building SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$ has the lowest AICc value (1397.30).

4.4 Unloaded container flow

(a) SA-FANN and GA-FANN results

In case of unloaded container flow, optimal FANN architecture contains 7 neurons in middle layer in both case, SA-FANN and GA-FANN, respectively. The proof for this conclusion are minimum values of MAE, MAPE and RMSE, which are highlighted in Table 12.

In Table 13 the values for each scenario and sensitivity (Table 2) for unloaded flow time series are given. As in case of previous time series the simulation results are more sensitive to "training times" then to "learning velocity".

(b) ARIMA results

Time series related to unloaded containers shows variable trend (decreasing in period from January 2010 to June 2014, increasing afterwards). To validate the need for first differencing we applied Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test for level and trend stationarity and results (KPSS Level = 0.57863, p value = 0.02458; KPSS Trend = 0.47013, p value = 0.01) suggest that first

Table 12 Varying number of neurons in hidden layer for unloaded container flow

Neurons in middle layer	SA-FANN			GA-FANN		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
1	926.75	2.11	1283.39	915.08	2.06	1280.74
2	843.42	1.91	1063.3	856.75	1.94	1258.92
3	918.41	2.07	1104.06	831.75	1.88	1247.81
4	835.08	1.89	1006.04	706.75	1.58	1016.26
5	811.91	1.84	1000.27	623.42	1.4	799.55
6	786.91	1.78	983.76	606.75	1.36	715.93
7	**760.08**	**1.71**	**883.61**	**581.75**	**1.29**	**635.84**
8	770.75	1.74	975.51	598.42	1.35	711.96
9	853.25	1.95	1079.41	590.08	1.33	701.21
10	786.58	1.79	933.77	923.41	2.16	1432.84

Table 13 Prediction accuracy analysis for different parameters of FANN in case of unloaded container flow time series

Comparison of different scenarios with various training times and scenarios with various learning velocities				Sensitivity analysis by using sensitivity index	
GA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.355177
R^2 (Parameter training)	0.88	0.89	0.92	Learning velocity	0.301981
R^2 (Parameter validation)	0.84	0.86	0.88		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.9	0.9	0.93		
R^2 (Parameter validation)	0.87	0.87	0.91		
SA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.341163
R^2 (Parameter training)	0.87	0.87	0.89	Learning velocity	0.277101
R^2 (Parameter validation)	0.77	0.78	0.82		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.88	0.88	0.91		
R^2 (Parameter validation)	0.83	0.84	0.89		

differencing is required (p value < 0.05). The Osborn–Chui–Smith–Birchenhall (Osborn et al. 1988) seasonal unit root test is used for proving the necessity for seasonal differencing (OCSB test returns positive results, output $= 1$). After performed first and seasonal differencing (d $= 1$, D $= 1$) a stationary time series has been obtained (KPSS test: p value $= 0.1$). Therefore, for unloaded

Table 14 Akaike Information Criterion with correction (AICc), mean absolute percentage error (MAPE) and stationary R^2 of SARIMA models for unloaded container flow

Models	MAPE	AICc	Stationary R^2
SARIMA$(1,0,2) \times (0,0,1)_{12}$	10.10	1441.92	0.32
SARIMA$(1,0,1) \times (1,0,1)_{12}$	9.82	1440.91	0.34
SARIMA$(2,0,2) \times (2,0,2)_{12}$	9.27	1446.96	0.42
SARIMA$(1,0,0) \times (1,0,0)_{12}$	9.91	1437.14	0.33

container time series SARIMA$(p, 1, q) \times (P, 1, Q)_{12}$ model is suggested. Assessment of alternative configurations based on AICc criteria led to selection of SARIMA$(1, 1, 2) \times (1, 1, 1)_{12}$ as the most appropriate model (Table 14). No outliers were detected. The model looks as follows:

$$(1 - 0.78B)(1 + 0.08B^{12})(1 - B)(1 - B^{12})Y_t = (1 - 1.67B + 0.82B^2)(1 - 0.47B^{12})\varepsilon_t$$

(18)

Diagnostic checks show that residuals are white noise and that there is no significant autocorrelation between them at different lag times. Coefficient of determination of selected model is 0.75. Additional tests based on model building performed on the whole dataset point on the SARIMA$(1, 1, 1) \times (0, 1, 1)_{12}$ as the most appropriate model (AIC$_c$ = 1332.41).

4.5 Empty container flow

(a) SA-FANN and GA-FANN results

Table 15 presents the topology with the best results based on the smallest errors (bold values in the table). In case of SA-FANN 5 neurons exist in the middle layer whereas the topology for GA-FANN has 8 neurons in the middle layer.

The values for each scenario and sensitivity (Table 2) for empty container flow time series are given in Table 16. As in previous cases, the simulation results of training times are more sensitive to parameter "learning velocity".

(b) ARIMA results

The last time series considers empty container flows. Large p values of KPSS test (p value = 0.1) point on stationarity of training dataset. OCSB test also returns negative results regarding the need for seasonal differencing. Reason for this lies in a possibility (evident from time series plot) that seasonality does not exists in all 12 months but just for some months (seasonal pulse in month 8 beginning from year 2011). Therefore, a SARIMA$(p, 0, q) \times (P, 0, Q)_{12}$ model is selected as a general structure for empty container flow time series. ACF/PACF observations point on SARIMA$(1, 0, 2) \times (0, 0, 1)_{12}$ as a first tentative model.

Table 15 Varying number of neurons in hidden layer for empty container flow

Neurons in middle layer	SA-FANN			GA-FANN		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
1	1362.08	2.79	1544.07	1725.42	3.46	1863.68
2	1328.75	2.73	1516.73	1642.08	3.26	1768.65
3	1245.42	2.54	1374.22	1567.08	3.13	1666.61
4	1162.08	2.34	1331.53	1533.75	3.05	1634.01
5	**964.42**	**1.96**	**1245.31**	1608.75	3.21	1687.17
6	1078.75	2.14	1235.98	1633.75	3.26	1714.04
7	995.41	1.98	1145.78	1550.41	3.05	1638.83
8	1014.42	2.01	1227.86	**1496.75**	**3.03**	**1711.86**
9	1064.42	2.1	1298.63	1500.41	2.96	1586.81
10	1006.08	1.97	1270.38	1512.58	3.01	1640.76

Table 16 Prediction accuracy analysis for different parameters of FANN in case of unloaded container flow time series

Comparison of different scenarios with various training times and scenarios with various learning velocities				Sensitivity analysis by using sensitivity index	
GA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.300122
R^2 (Parameter training)	0.86	0.87	0.9	Learning velocity	0.229989
R^2 (Parameter validation)	0.71	0.73	0.82		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.85	0.89	0.9		
R^2 (Parameter validation)	0.83	0.87	0.86		
SA-FANN					
Training times	100	300	500	Parameters	Sensitivity index
Learning velocity	0.01	0.01	0.01	Training times	0.319863
R^2 (Parameter training)	0.88	0.88	0.91	Learning velocity	0.240011
R^2 (Parameter validation)	0.8	0.81	0.87		
Training times	100	300	500		
Learning velocity	0.0001	0.0001	0.0001		
R^2 (Parameter training)	0.9	0.9	0.92		
R^2 (Parameter validation)	0.87	0.86	0.9		

Alternatives based on slight modifications of parameters result in a conclusion that the $SARIMA(1, 0, 0) \times (1, 0, 0)_{12}$ is the model with minimum AICc (Table 17).

As in the case of transit container flows, we considered the presence of outliers and identified a temporary change (TC) outlier in May 2011. $SARIMA(1, 0, 0) \times (1, 0, 0)_{12}$ model selected in first phase remains the

Table 17 Akaike Information Criterion with correction (AICc), mean absolute percentage error (MAPE) and stationary R^2 of SARIMA models for empty container flow

Models	MAPE	AICc	Stationary R^2
SARIMA$(1,0,2) \times (0,0,1)_{12}$	10.10	1441.92	0.32
SARIMA$(1,0,1) \times (1,0,1)_{12}$	9.82	1440.91	0.34
SARIMA$(2,0,2) \times (2,0,2)_{12}$	9.27	1446.96	0.42
SARIMA$(1,0,0) \times (1,0,0)_{12}$	9.91	1437.14	0.33

same (considering the effect of TC outlier) but with slightly improved AICc (AICc = 1428.17). The model looks like follows:

$$Y_t = 14968 \frac{1}{1 - 0.7B} I_t(17) + \frac{1}{(1 - 0.47B)(1 - 0.28B^{12})} \varepsilon_t \tag{19}$$

Diagnostic check proves validity of selected model. Ljung–Box Q test and p value (Ljung–Box Q = 12.043, p value = 0.741) prove validity of the model. If we take the whole dataset for ARIMA model building we obtain SARIMA$(0,1,1) \times (1,0,0)_{12}$ (AICc = 1659.23). The model identifies two outliers (temporary change outlier—May 2011 and additive outlier—November 2016) and looks as follows:

$$Y_t = 13614 \frac{1}{1 - 0.7B} I_t(17) - 18685I_t(83) + \frac{(1 - 0.51B)}{(1 - B)(1 - 0.36B^{12})} \varepsilon_t \tag{20}$$

4.6 Discussion of results

Each of the forecasting methods has been applied for computing the predicted values of container flows for the test period from January 2016 to December 2016. One of the most important practical issues for obtaining a good forecast is the size of the sample used to build the prediction model. There is no specific rule that can be followed, however, a larger sample will in general provide a better chance for any model to adequately approximate the underlying data structure. In case of time series forecasting methods, like ARIMA is, Box and Jenkins (1976) have suggested at least 50 or even better 100 observations for building linear ARIMA models. For nonlinear models, like SA-FANN and GA-FANN are, larger sample sizes are desirable. A larger sample provides a better chance for neural networks to adequately approximate the underlying data structure. On the other side, if data in the sample are not homogeneous or the underlying data generating process in a time series changes over time, then a larger sample may even hurt performance of neural networks. Besides the data quantity, data quality is important too. The key is to use training data that generally span the problem data space. In concrete case, at the time of our analysis a time series of monthly container throughput observations from January 2010 to December 2016 (84 months) were available. However, despite the "limited data availability" the data set does not contain a lot of random variation (standard deviation versus mean ratio is much lower than 1 for all time series even with anomalies included) so the authors consider the data set is sufficient for reliable container flow modeling and short-term forecast generation even for dealing with periodical effects like such as seasonality.
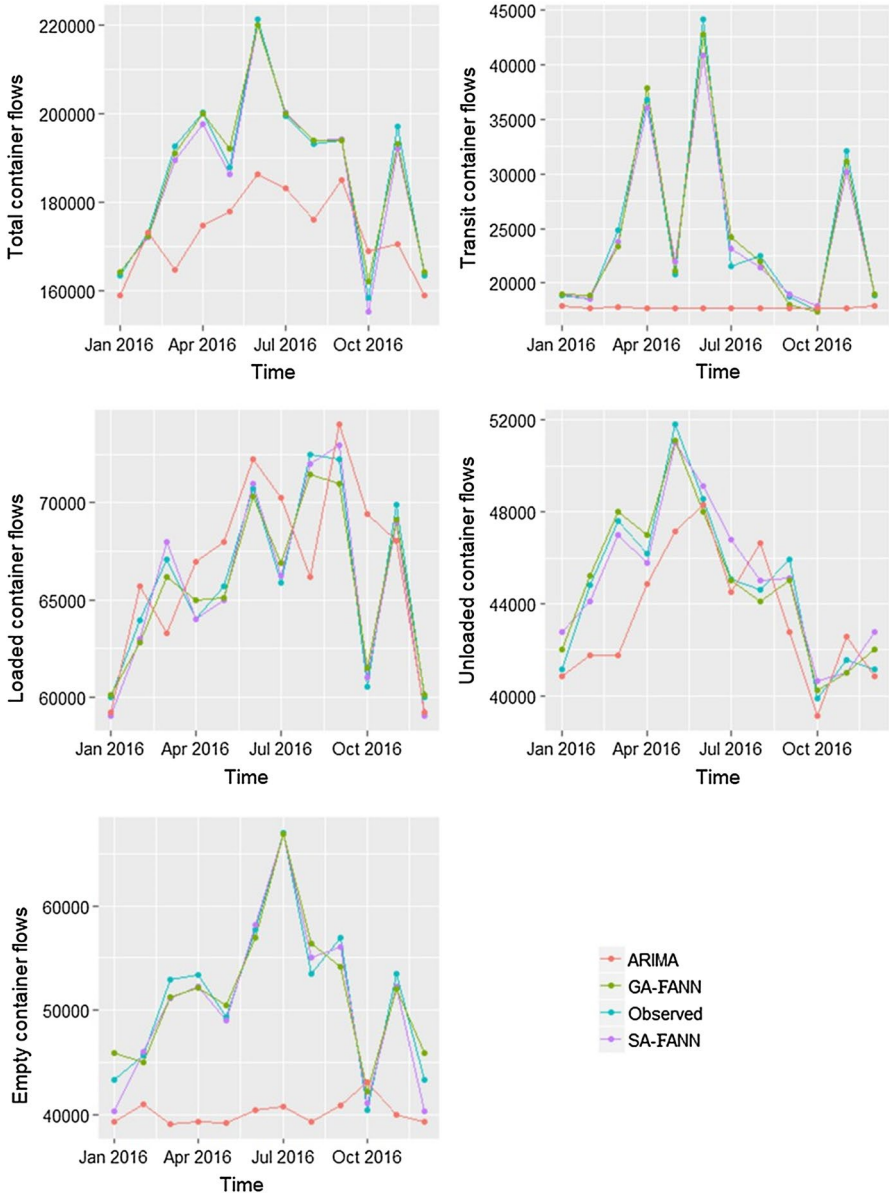
Fig. 4 Container flows at Barcelona Port—fitting performances of SA-FANN, GA-FANN and ARIMA

Actual observations as well as the predictions generated by each of proposed models for total, transit and loaded, unloaded and empty container flows are graphically illustrated on Fig. 4.

In order to test forecasting performances, we selected the mean average error (MAE), the and mean absolute percent error (MAPE) and the root mean squared error (RMSE) defined as follows (Xie et al. 2013):

$$MAE = \frac{\sum_{t=1}^{n} (Y_t - \overline{Y_t})}{n} \qquad (21)$$

$$MAPE = \frac{100 \sum_{t=1}^{n} \left| 1 - \frac{\overline{Y_t}}{Y_t} \right|}{n} \qquad (22)$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (Y_t - \overline{Y_t})^2}{n}} \qquad (23)$$

where Y_t and $\overline{Y_t}$ represent the actual and the predicted values of the time series in period t, respectively.

Table 18 contains comparison of forecasting accuracy of proposed methods for all time series considered. Values of MAE, MAPE and RMSE clearly show that ARIMA model has much lower performance level than SA-FANN and GA-FANN for each of the time series. In case of transit (MAPE = 22.49) and empty container flows (MAPE = 20.83) this difference is obvious. When we look at the transit container flow time series we may note that the beginning of the series obviously behaves very differently from later parts. Also, we have three large spikes during the test period which are not forecastable by ARIMA model. Empty container flow time series has a more "regular" pattern and that was the reason for slightly better shape of ARIMA model, but still much worse than in case of total (MAPE = 8.03), loaded (MAPE = 4.75) and unloaded (MAPE = 4.48) data samples. In this case the influence of outliers was not decisional, but the pattern itself.

Table 18 Performance of proposed methods for forecasting container flows at Barcelona Port

	Total container flow			Transit container flow			Loaded container flow		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
SA-ANN	1776.43	1.15	2404.43	866.57	3.70	1242.61	598.64	1.11	737.69
GA-ANN	1499.29	0.97	2111.54	710.93	3.31	1014.81	768.21	1.40	893.39
ARIMA	15,646.23	8.03	18,809.58	7042.03	22.49	10,671.28	3109.93	4.75	3883.79

	Unloaded container flow			Empty container flow		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
SA-ANN	651.5	1.71	818.7	826.64	1.96	1152.94
GA-ANN	498.64	1.29	588.68	1282.93	3.03	1584.88
ARIMA	2084.56	4.48	2699.69	11,444.3	20.83	13,423.66

The reason for this lies in embedded nonlinearity of considered time series. In general, to check for nonlinearity of time series and therefore necessity to use nonlinear dynamics forecasting models, one approach can be to fit a general linear model to the data (ARIMA model in this case) and then to use some statistics computed from the residuals. Existing analysis in this paper has been extended by the BDS test for detecting nonlinear serial dependence in time series. The BDS test produces a viable test of linearity against the alternative of non-linearity when the data are prefiltered by ARIMA fit. According to the results of BDS test for all container flow time series, the null hypothesis (data are i.i.d) is rejected and hence, the series are nonlinear. In other words, fitted linear time series models have left certain aspects of the data unexplained—residuals dataset after fitting ARIMA contains

Table 19 Non linearity testing for ARIMA residuals of container flow time series

Parameter ε/σ	Dimension (m=2)		Dimension (m=3)	
	Statistic	Probability	Statistic	Probability
Total container flows				
0.5	2.2943	0.0218	5.2050	0.0000
1.0	− 1.7847	0.0743	− 0.4378	0.6616
1.5	− 2.0863	0.0370	− 1.3018	0.1930
2.0	− 2.0921	0.0364	− 1.4595	0.1444
Transit container flows				
0.5	8.5184	0.0000	10.3138	0.0000
1.0	8.6847	0.0000	10.0408	0.0000
1.5	7.1944	0.0000	8.7226	0.0000
2.0	3.0425	0.0023	5.6219	0.0000
Loaded container flows				
0.5	4.7451	0.0000	8.4215	0.0000
1.0	2.0946	0.0362	1.4787	0.1392
1.5	0.6356	0.5250	− 0.2618	0.7935
2.0	− 0.3487	0.7273	− 0.5223	0.6014
Unloaded container flows				
0.5	4.3497	0.0000	9.9094	0.0000
1.0	1.0649	0.2869	1.5427	0.1229
1.5	− 0.4874	0.6260	− 0.4792	0.6318
2.0	− 0.0128	0.9898	− 0.6568	0.5113
Empty container flows				
0.5	− 2.3816	0.0172	− 4.4955	0.0000
1.0	− 1.5062	0.132	− 1.8122	0.0700
1.5	− 1.4839	0.1378	− 1.3565	0.1749
2.0	− 3.2268	0.0013	− 2.9551	0.0031

m represents the embedding dimension. ε is equal to 0.5, 1.0, 1.5 and 2.0 times the standard deviation. The critical value for confidence level of 5% is 1.96. The results presented in Table 19 demonstrate that the null hypothesis of i.i.d. for the residuals can be rejected at the 5% level of confidence

nonlinear component and so can be properly modeled through an FANN. The results of BDS test are given in Table 19.

If we compare performances of two non-parametric approaches, the results revealed different performances. More precisely, for total, transit and unloaded container flow GA-ANN obtained a slightly better performance then SA-FANN. On the other side, for loaded and empty container flows SA-FANN has lower values against all performance measures. The reason for relative difference lies in the algorithmic differences between simulated annealing and genetic algorithm, specific characteristics of considered time series, and the set of parameters used and their values.

Based on presented results we may apply the best method for predicting the future volume of container flows. Therefore, we apply GA-FANN method to generate forecasts for total, transit and unloaded container flows and SA-FANN for forecasting of loaded and empty container flows. Superiority of developed nonlinear techniques is aligned with results of comparison of nonlinear and linear techniques for container flow forecasting from literature (Lam et al. 2004; Chen and Chen 2010; Huang et al. 2015; Xiao et al. 2014).

Forecasts generated by the developed methods are essential inputs for the control and scheduling of operations in any port system, and also for terminal operators in decision making and planning. Acquisition of additional equipment and material as well as the allocation and arrangement of workers and machines can be performed more efficiently. Synchronization between sea and land transportation modes can significantly be improved and on that way also the competitiveness of intermodal transportation. Good prediction model is essential for the terminal operators to make decisions for planning and modernization of building structure and other port facilities.

5 Concluding remarks

Long term strategic plan of port terminals is to gain higher market shares through improved intermodal transport services. Forecasts of future container flows represent important mean for better planning of intermodal supply chains. In this paper we proposed a hybrid methodology based on metaheuristics for designing an optimal fuzzy ANN architecture for container flow forecasting. Proposed approach is combines fuzzy neural network method and two sophisticated heuristics for developing the neural network architecture. We compared the performance of the developed methods against the traditional ARIMA technique for forecasting container throughput volumes. SA-FANN, GA-FANN and ARIMA have been applied to model container throughput based on the monthly observations of Port of Barcelona, 9th largest port in Europe which tends to become one of the main Euroregion distribution centers in the Mediterranean. Five container flow related time series: total, transit, loaded, unloaded and empty container flows were independently modeled. For each time series we compared the predicting accuracy of the models by calculating MAE, RMSE and MAPE for each of the models and selected the models which have the best performances. Developed non-parametric methods significantly outperform ARIMA technique for all time series. Between those two, GA-FANN outperforms

SA-FANN in case of total, transit and unloaded container flows, whereas SA-FANN has much lower values of performance measures for loaded and empty container flows. Having in mind that container flows are highly dependent on a number of uncertainty factors, future research will be based on the development of more complex multivariate AI-based modeling tools.

References

Abdalla OA, Osman A, AlMurtadha YM (2014) Optimizing the multilayer feed-forward aritificial neural networks architecture and training parameters using genetic algorithms. Int J Comput Appl 96(10):42–48

Agatonovic KS, Beresford R (2000) Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. J Pharm Biomed Anal 22:717–727

Arifovic J, Gencay R (2001) Using genetic algorithms to select architecture of feedforward artificial neural network. Physica 289:574–594

Atsalakis G, Valavanis K (2008) Surveying stock market forecasting techniques—Part II: soft computing methods. Expert Syst Appl 36:5932–5941

Atsalakis G, Valavanis K (2009) Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert Syst Appl 36:10696–10707

Atsalakis G, Dimitrakakis E, Zopounidis C (2011) Elliot wave theory and neuro-fuzzy systems, in stock market prediction. Expert Syst Appl 38(8):9196–9206

Blanco A, Delgado M, Pegalajar MC (2000) A genetic algorithm to obtain the optimal recurrent neural network. Int J Approx Reason 23:67–83

Box GEP, Jenkins GM (1976) Time series analysis: forecasting and control. Holden Day, San Francisco

Box GE, Jenkins GM, Reinsel GC (2008) Time series analysis, forecasting and control. Wiley, New Jersey

Chen SH, Chen JN (2010) Forecasting container throughputs at ports using genetic programming. Expert Syst Appl 37(3):2054–2058

Chen C, Liu LM (1993) Joint estimation of model parameters and outlier effects in time series. J Am Stat Assoc 88(421):284–297

Chibante R (2010) Simulated annealing theory with applications. Sciyo, Croatia

Chou CC (2004) A study on forecasting the container volume of international ports in Taiwan area. Ph.D. dissertation, Department of Shipping and Transportation Management, National Taiwan Ocean University

Chou CC, Chu CW, Liang GS (2008) A modified regression model for forecasting the volumes of Taiwan's import containers. Math Comput Model 47(9–10):797–807

Denai MA, Palis F, Zeghbib A (2007) Modeling and control of non-linear systems using soft computing techniques. Appl Soft Comput 7(3):728–738

Diaz R, Talley W, Tulpule M (2011) Forecasting empty container volumes. Asian J Shipp Logist 27(2):217–236

Donate JP, Li X, Sanchez GG, Miguel ASD (2011) Time series forecasting by evolving artificial neural networks with genetic algorithms, differential evolution and estimation of distribution algorithm. Neural Comput Appl 22(1):11–20

Fung MK (2002) Forecasting Hong Kong's container throughput: an error-correction model. J Forecast 21(1):69–80

Gao Y, Luo M, Zou G (2016) Forecasting with model selection or model averaging: a case study for monthly container port throughput. Transportmetrica A Transp Sci 12(4):366–384

Ghasemiyeh R, Moghdani R, Sana SS (2017) A hybrid artificial neural network with metaheuristic algorithms for predicting stock price. Cybern Syst 48:365–392

Ghosh R, Ghosh M, Yearwood J, Bagirov A (2005) Comparative analysis of genetic algorithm, simulated annealing and cutting angle method for artificial neural networks, pp 62–70. In: Machine learning and data mining in pattern recognition: proceedings of the 4th international conference, MLDM 2005, Germany

Glisovic N, Milenkovic M, Bojovic N, Svadlenka L, Avramovic Z (2015) A hybrid model for forecasting the volume of passenger flows on Serbian railways. Oper Res Int J 16(2):271–285

Gosasang V, Chandraprakaikul W, Kiattisin S (2011) A comparison of traditional and neural networks forecasting techniques for container throughput at Bangkok port. Asian J Shipp Logist 27(3):463–482

Herrera F, Martinez L (2000) A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Trans Fuzzy Syst 8(6):746–752

Huang A, Lai KK, Li Y, Wang S (2015) Forecasting container throughput of Qingdao port with a hybrid model. J Syst Sci Complex 28(1):105–121

Hui ECM, Seabrooke W, Wong GKC (2004) Forecasting cargo throughput for the Port of Hong Kong: error correction model approach. J Urban Plan Dev 130(4):195–203

Hui ECM, Ng MH, Xu JJ, Yip TL (2010) The cargo throughput response to factor cost differentials—an analysis for the port of Hong Kong. Transportmetrica 6(4):235–248

Ishibuchi H, Kwon K, Tanaka H (1993) Implementation of fuzzy IF-THEN rules by fuzzy neural networks with fuzzy weights, In: Proceedings of the EUFIT'93 conference, Aachen, Germany, pp 209–215

Ishibuchi H, Fujioka R, Tanaka H (1993b) Neural networks that learn from fuzzy IF-THEN rules. IEEE Trans Fuzzy Syst 1:85–97

Ishibuchi H, Tanaka H, Okada H (1993) Fuzzy neural networks with fuzzy weights and fuzzy biases, In: Proceedings of the IEEE international conference on neural networks, San Francisco, pp 447–452

Ishibuchi H, Tanaka H, Okada H (1994) Interpolation of fuzzy IF THEN rules by neural networks. Int J Approx Reason 10:3–27

Kaviani M, MirRokni SM (2017) Applying genetic algorithm in architecture and neural network training. Int J Comput Sci Netw Secur 17(6):118–124

Kosko B (1992) Neural networks and fuzzy systems. Prentice-Hall, Englewood Cliffs

Lam WHK, Ng PLP, Seabrooke W, Hui ECM (2004) Forecasts and reliability analysis of port cargo throughput in Hong Kong. J Urban Plan Dev 130(3):133–144

Lam W, Tang Y, Chan K, Tam M (2006) Short-term hourly traffic forecasts using Hong Kong annual traffic census. Transportation 33(3):291–310

Lee LW, Wang HF, Chen SM (2008) Temperature prediction & TAIFEX forecasting based on high-order fuzzy logical relationships & genetic simulated annealing techniques. Expert Syst Appl 34(1):328–336

Lee SY, Lim H, Kim HJ (2017) Forecasting container port volume: implications for dredging. Marit Econ Logist 19(2):296–314

Liao GC, Tsao TP (2006) Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting. IEEE Trans Evol Comput 10(3):330–340

Lingras P, Sharma SC, Osborne P, Kalyar I (2000) Traffic volume time-series analysis according to the type of road use. Comput Aided Civ Infrastruct Eng 15(5):365–373

De Livera AM, Hyndman RJ, Snyder RD (2010) Forecasting time series with complex seasonal patterns using exponential smoothing. J Am Sta Assoc 106:1513–1527

Liu Z, Ji L, Ye Y, Geng Z (2007) Combined forecast method of port container throughput based on RBF neural network. J Tongji Univ (Nat Sci) 35(5):739

MacKay JCD (2003) Information theory, inference and learning algorithms. Cambridge University Press, Cambridge

Michalewicz Z, Fogel DB (2000) How to solve it: modern heuristics. Springer, Berlin

Milenkovic M, Svadlenka L, Melichar V, Bojovic N, Avramovic Z (2016) SARIMA modelling approach for railway passenger flow forecasting. Transport 33(5):1113–1120

Mo L, Xie L, Jiang X, Teng G, Xu L, Xiao J (2018) GMDH-based hybrid model for container throughput forecasting: selective combination forecasting in nonlinear subseries. Appl Soft Comput 62:478–490

Moscoso-Lopez JA, Turias IJ, Jimenez-Cone MJ, Ruiz-Aguilar JJ, Cerban M (2016) Short-term forecasting of intermodal freight using ANNs and SVR: case of the Port of Algeciras Bay. Transp Res Procedia 18:108–114

Niu M, Hu Y, Sun S, Liu Y (2018) A novel hybrid decomposition-ensemble model based on VMD and HGWO for container throughput forecasting. Appl Math Model 57:163–178

Osborn DR, Chui APL, Smith JP, Birchenhall CR (1988) Seasonality and the order of integration for consumption. Oxford Bull Econ Stat 50:361–377

Pang G, Gebka B (2017) Forecasting container throughput using aggregate or terminal-specific data. The case of Tanjung Priok Port, Indonesia. Int J Prod Res 55:2454–2469

Peng WY, Chu CW (2009) A comparison of univariate methods for forecasting container throughput volumes. Math Comput Model 50(7–8):1045–1057

Port of Barcelona (2017) Statistical reports 2010–2016. http://www.portdebarcelona.cat/en/web/autoritat-portuaria/estadisticas. Accessed 16 Jan 2017

Ruiz-Aguilar JJ, Turias IJ, Jiménez-Come MJ (2014) Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting. Transp Res Part E Logist Transp Rev 67:1–13

Schulze PM, Prinz A (2009) Forecasting container transshipment in Germany. Appl Econ 41(22):2809–2815

Seabrooke W, Lam WHK, Wong GKC (2003) Forecasting cargo growth and regional role of the port of Hong Kong. Cities 20(1):51–64

Sexton RS, Dorsey RE, Johnson JD (1999) Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing. Eur J Oper Res 114:589–601

Smith BL, Williams BM, Keith Oswald R (2002) Comparison of parametric and nonparametric models for traffic flow forecasting. Transp Res Part C 10(4):303–321

Soares S, Antunes CH, Araujo R (2013) Comparison of genetic algorithm and simulated annealing for automatic neural network ensemble development. Neurocomputing 119:498–521

Suhartono S (2011) Time series forecasting by using seasonal autoregressive integrated moving average: subset: multiplicative or additive model. J Math Stat 7(1):20–27

Szu H, Hartley R (1987) Fast simulated annealing. Phys Lett A 122(3–4):157–162

Taskaya-Temizel T, Casey MC (2005) A comparative study of autoregressive neural network hybrids. Neural Netw 18(5–6):781–789

Tsai FM, Huang LJW (2015) Using artificial neural networks to predict container flows between the major ports of Asia. Int J Prod Res 55(17):5001–5010

Twrdy E, Batista M (2016) Modeling of container throughput in Northern Adriatic ports over the period 1990–2013. J Transp Geogr 52:131–142

Vlahogianni EI, Golia JC, Karlaftis MG (2004) Short-term traffic forecasting: overview of objectives and methods. Transport Rev 24(5):533–557

Vlahogianni EI, Karlaftis MG, Golias JC (2005) Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transp Res Part C Emerg Technol 13(3):211–234

Wei Y, Chen MC (2012) Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transp Res Part C 21(1):148–162

Wu TH, Chang CC, Chung SHA (2008) Simulated annealing algorithm for manufacturing cell formation problems. Expert Syst Appl 34(3):1609–1617

Xiao J, Xiao Y, Fu J, Lai KK (2014) A transfer forecasting model for container throughput guided by discrete PSO. J Syst Sci Complex 27(1):181–192

Xie G, Wang S, Zhao Y, Lai KK (2013) Hybrid approaches based on LSSVR model for container throughput forecasting: a comparative study. Appl Soft Comput 13(5):2232–2241

Xie G, Zhang N, Wang S (2017) Data characteristic analysis and model selection for container throughput forecasting within a decomposition-ensemble methodology. Transp Res Part E 108:160–178

Yan W (2012) Toward automatic time-series forecasting using neural networks. IEEE Trans Neural Netw Learn Syst 23(7):1028–1039

Zadeh LA (1965) Fuzzy set. Inf Control 8(3):338–353

Zhang PL, Cui Y (2011) Research on combination forecast of port container throughput based on Elman neural network. In: Proceedings of the 3rd international conference on communication software and networks (ICCSN), pp 567–570

Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: the state of the art. Int J Forecast 14:35–62

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.