A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative...

Article *in* Expert Systems with Applications · April 2009

DOI: 10.1016/j.eswa.2008.08.058 · Source: DBLP

Tuğba Efendigil
Yildiz Technical University

Cengiz Kahraman
Istanbul Technical University

ARTICLE IN PRESS

Expert Systems with Applications xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis

Tuğba Efendigil [a,*], Semih Önüt [a], Cengiz Kahraman [b]

[a] Yildiz Technical University, Mechanical Faculty, Department of Industrial Engineering, 34349 Yildiz, Istanbul, Turkey
[b] Istanbul Technical University, Faculty of Management, Department of Industrial Engineering, 34367 Macka, Istanbul, Turkey

ARTICLE INFO

Keywords:
Supply chain
Demand forecasting
Fuzzy inference systems
Neural networks

ABSTRACT

An organization has to make the right decisions in time depending on demand information to enhance the commercial competitive advantage in a constantly fluctuating business environment. Therefore, estimating the demand quantity for the next period most likely appears to be crucial. This work presents a comparative forecasting methodology regarding to uncertain customer demands in a multi-level supply chain (SC) structure via neural techniques. The objective of the paper is to propose a new forecasting mechanism which is modeled by artificial intelligence approaches including the comparison of both artificial neural networks and adaptive network-based fuzzy inference system techniques to manage the fuzzy demand with incomplete information. The effectiveness of the proposed approach to the demand forecasting issue is demonstrated using real-world data from a company which is active in durable consumer goods industry in Istanbul, Turkey.

1. Introduction

A supply chain (SC) has a dynamic structure involving the constant flow of information, product, and funds between different stages (Chopra & Meindl, 2001). Supply chain process has three important stages; *supply*, *production,* and *distribution* including not only manufacturer and suppliers, but also transporters, warehouses, retailers, and customers themselves. The flow of information, knowledge, product, or resources between and among these entities is to be managed appropriately to maximize the overall profitability. Specifically, information flow between departments is the most important connection links for a SC's success.

Forecasting is a part of the supply management picture and directly affects both quantity and delivery. Forecasts of usage, supply, market conditions, technology, prices, and so on, are always necessary to make good decisions (Leernders, Fearon, Flynn, & Johnson, 2002). To have an available decision making system is becoming a crucial issue for organizations in a constantly fluctuating environment where the economic uncertainty needs the mathematical models. Forecasting the expected demand for a certain period of time with one or more products is one of the most relevant targets in an enterprise. It is unavoidable to be able to know or predict the future demand as close to reality as possible. In spite of the need for accurate forecasting to enhance the commercial competitive advantage, there is no standard approach.

To this end, this paper suggests a comparative forecasting approach for collaborative organizations to create a supply chain frame that has dynamic characteristics using real-time information in production–distribution systems. Section 2 presents a critical view of past work on forecasting studies in SC and artificial intelligence. Section 3 describes the techniques used in the proposed methodology. A real-world case study from Istanbul is presented in Section 4. Section 5 gives the results of the neural techniques. Section 6 concludes this paper by giving important extensions and future directions of this work.

2. Literature review

2.1. Forecasting and supply chain

There are many forecasting techniques that can be classified into four main groups: (1) *Qualitative methods* are primarily subjective; they rely on human judgment and opinion to make a forecast. (2) *Time-series methods* use historical data to make a forecast. (3) *Causal methods* involve assuming that the demand forecast is highly correlated with certain factors in the environment (e.g., the state of the economy, interest rate). (4) *Simulation methods* imitate the consumer choices that give rise to demand to arrive at a forecast (Chopra & Meindl, 2001).

Most prior studies have been applied to predict the customer demand primarily based on *time-series models,* such as

* Corresponding author. Tel.: +90 212 383 2872; fax: +90 212 258 5928.
E-mail addresses: tfdolas@yildiz.edu.tr (T. Efendigil), onut@yildiz.edu.tr (S. Önüt), kahramanc@itu.edu.tr (C. Kahraman).

0957-4174/$ - see front matter Crown Copyright © 2008 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.08.058

moving-average, exponential smoothing, and the Box-Jenkins method, and *casual models,* such as regression and econometric models. Lee, Padmanabhan, and Whang (1997) analyzed the demand variability amplification along a SC from retailers to distributors, and named this amplification effect the bullwhip effect. Chen, Drezner, Ryan, and Simchi-Levi (2000) quantified the bullwhip effect for simple, two-stage SCs consisting of a single retailer and a single manufacturer. Their model included two factors assumed to cause bullwhip effect; demand forecasting, and order lead times. They supposed that the retailer used a simple moving-average forecast to estimate the mean and variance of demand and to form a simple order-up-to inventory policy. They extended the results to general multiple-stage SCs with and without centralized customer demand information and demonstrated that the bullwhip effect could be reduced by centralizing demand information. Chen, Ryan, and Simchi-Levi (2000) demonstrated that the use of an exponential smoothing forecast by the retailer could cause the bullwhip effect. The authors implied that magnitude of the increase in variability dependent on both the nature of the customer demand process and the forecasting technique used by retailer. Zhao, Xie, and Leung (2002) investigated the impact of forecasting models on SC performance and the value of sharing information in a SC with one capacitated supplier and multiple retailers under demand uncertainty via a computer simulation model. They examined demand forecasting and inventory replenishment decisions by the retailers and production decisions by the supplier under different demand patterns and capacity tightness.

Although the quantitative methods mentioned above perform well, they suffer from some limitations. First, lack of expertise might cause a mis-specification of the functional form linking the independent and dependent variables together, resulting in a poor regression. Secondly, a large amount of data is often required to guarantee an accurate prediction. Thirdly, non-linear patterns are difficult to capture. Finally, outliers can bias the estimation of the model parameters. Some of these limitations can be overcome by the use of neural networks, which have been mathematically demonstrated to be universal approximates of functions (Garetti & Taisch, 1999).

2.2. Forecasting and artificial intelligence in supply chain

Generally the artificial neural networks (ANN) are increasingly used by utilities to forecast short or long term demands for electric load (Al-Saba & El-Amin, 1999; Beccali, Cellura, Lo Brano, & Marvuglia, 2004), energy use (Hobbs, Helman, Jitprapaikulsarn, Konda, & Maratukulam, 1998; Sözen, Arcaklioğlu, & Özkaymak, 2005) and tourism (Law, 2000; Law & Au, 1999). For more application area of ANN, see Ayata, Çam, and Yıldız (2007), Cavalieri, Maccarrone, and Pinto (2004), Efendigil, Önüt, and Kongar (2008), Metaxiotis and Psarras (2003), Sabuncuoglu (1998), Vellido, Lisboa, and Vaughan (1999), Wong, Bodnovich, and Selvi (1997), Wong and Selvi (1998), Wong, Lai, and Lam (2000).

Developing better forecasting approaches to reduce or eliminate inventories affecting the total cost of SC is becoming an important issue nowadays. As a new tool, ANN has already been used in demand forecasting systems or as data pre-processors for smoothing and classifying noisy data to match the relationships between complicated functions.

A supply chain can be modeled as various levels, such as material processing, manufacturing, distributing, customers, etc. in where neural networks are considered as the primary and auxiliary problem solving methodology. The areas where NNs are used in supply chain are; optimization (in transportation management, resources allocation and scheduling), forecasting (for vague states in one echelon is bound to be propagated to the others in the chain), modeling and simulation (for dynamics of supply chain using techniques such as discrete event simulation and dynamic systems theory), globalization (for increasing coordination between activities happening in different centers) and decision support (for management and analysis of data for the support of a decision) (Leung, 1995).

Luxhøj, Riis, and Stensballe (1996) presented a hybrid econometric NN model for forecasting total monthly sales of a Danish company. This model attempted to integrate the structural characteristic of econometric models with the non-linear pattern recognition features of NN. Aburto and Weber (2007) presented a hybrid intelligent system combining autoregressive integrated moving-average models and NN for demand forecasting in SCM and developed a replenishment system for a Chilean supermarket. Du and Wolfe (1997) presented details of the implementation of neural networks and/or fuzzy logic systems in industry, especially in the areas of scheduling and planning, inventory control, quality control, group technology and forecasting. Gaafar and Choueiki (2000) applied ANN to the problem of lot-sizing in material resource planning for the case of a deterministic time-varying demand pattern over a fixed planning horizon. Shervais, Shannon, and Lendaris (2003) employed a set of neural networks to select an optimal set of both transport and inventory policies for a multi-product, multi-echelon, multi-model physical distribution system in a non-stationary environment. Chiu and Lin (2004) showed how collaborative agents and ANN could work together to enable collaborative SC planning with a computational framework for mapping the supply, production and delivery resources to the customer orders.

Due to the increasing market complexity and ambiguity, demand forecasting issue has been studied with collaborative techniques to have satisfactory results. Combinations of neural networks and fuzzy systems are one of those techniques. In SC literature very few studies have been considered using fuzzy neural networks (FNN) to forecast demand. Escoda, Ortega, Sanz, and Herms (1997) focused on the development and representation of linguistic variables to qualify the product demand by means of ANN and FNN. Kuo (1998) proposed a decision support system for the stock market via fuzzy Delphi and FNN. In another studies, Kuo and Xue (1998)) and Kuo, Wu, and Wang (2002) developed an intelligent sales forecasting system which considered quantitative and qualitative factors by integrating ANN and FNN. However, still there is a lack of implementing neuro-fuzzy techniques in demand forecast.

Statistical methods are only efficient for data having seasonal or trend patterns, while artificial neural techniques are also efficient for data which are influenced by the special case, like promotion or extreme crisis. Artificial neural techniques have been recently employed and successful results have been obtained in demand/sales forecasting area. While there were a limited number of publications using fuzzy neural networks to forecast demand, there is no evidence that any was applied to the issue of demand forecasting in SC using a comparative approach of ANN and neuro-fuzzy techniques, specifically. Thus, this study is a first attempt to develop a comparative forecasting methodology coping with the fuzziness of data via ANN and neuro-fuzzy systems. The proposed model including ANN and neuro-fuzzy techniques is explained in the following section.

3. Proposed model

Artificial intelligence forecasting techniques have been receiving much attention lately in order to solve problems that are hardly solved by the use of traditional methods. They have been cited to have the ability to learn like humans, by accumulating knowledge through repetitive learning activities. Therefore the objective of the paper is to propose new forecasting techniques via the artificial approaches to manage demand in a fluctuating environment. In this study, a comparative analysis based on neural techniques is presented for customer demands in a multi-level supply chain structure. The artificial techniques used in this study are explained as follows.

3.1. Artificial neural networks

Artificial neural networks (ANN) were developed in attempts to simulate the animal brain's cognitive learning process. However, in the past decade, they also attracted substantial attention in business industry. ANNs are proved to be efficient in modeling complex and poorly understood problems for which sufficient data are collected (Dhar & Stein, 1997). ANN is a technology that has been mainly used for prediction, clustering, classification, and alerting to abnormal patterns (Haykin, 1994). The capability of learning examples is probably the most important property of neural networks in applications and can be used to train a neural network with the records of past response of a complex system (Wei, Zhang, & Li, 1997).

The basic element in an ANN is a neuron. The model of a neuron is depicted in Fig. 1 (Haykin, 1994). In mathematical terms, a neuron k can be described as in Eqs. (1) and (2):

$$u_k = \sum_{j=1}^{p} w_{kj} x_j, \tag{1}$$

$$y_k = \varphi(u_k - \theta_k), \tag{2}$$

where $x_1, x_2, \ldots, x_p$ are the *input signals*; $w_{k1}, w_{k2}, \ldots, w_{kp}$ are the *synaptic weights* of neuron k, and, u_k is the *linear combiner output* while θ_k denotes the *threshold*. Furthermore, $\Phi(\cdot)$ is the *activation function*; and y_k is the *output signal* of the neuron (Haykin, 1994).

Each neuron is linked to some of its neighbors with varying coefficients of connectivity representing the strengths of these connections. Learning is accomplished by adjusting their strength so that neurons can then be grouped into layers. The input layer consists of neurons that receive input from the external environment. The output layer consists of neurons that communicate the output of the system to the user or external to the environment. There are usually a number of hidden layers between these two layers. The hidden layer of an ANN model acts as a black box to link the relationship between input and output (Choy, Lee, & Lo, 2003). When the relationship between the input and output variables is non-linear, a hidden layer helps in extracting higher level features and facilitates the generalization of outputs (Koskivaara, 2004). Fig. 2 exhibits the general structure of the one-hidden-layer ANN (Kuo & Xue, 1998).

The ANN's input layer with some neurons represents the previous sales data, for generic period $t - p$ to period $t - 1$, which are connected to the hidden layer. The hidden layer with some neurons is connected with the output layer with one single neuron which represents the sales for period t. T and O are the desired and actual outputs, respectively, and p indicates the sample number.

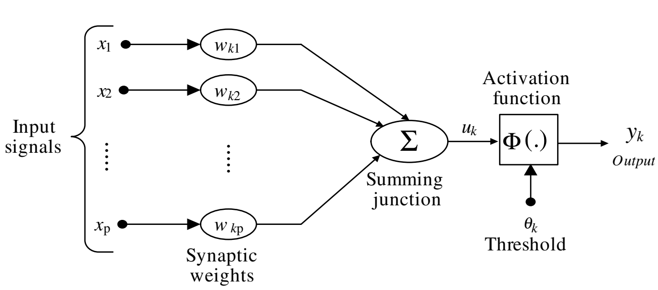


Fig. 1. Non-linear model of a neuron.

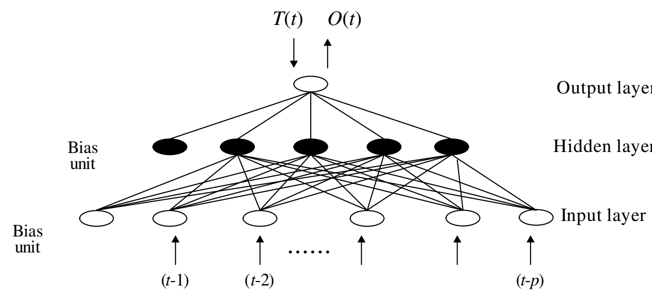


Fig. 2. The ANN structure.

The learning method can be divided into two categories, namely, *unsupervised learning* and *supervised learning*. A *backpropagation supervised learning model* is designed in this study. The error between the expected output and the calculated output is computed. Then a minimization procedure is used to adjust the weights between two connection layers starting backwards from the output layer to input layer. There are a number of variations of minimization procedures that are based on different optimization methods, such as *gradient descent*, *conjugate gradient*, *Quasi-Newton*, and *Levenberg–Marquardt* methods. The differences between them are based on various weight adjustments (Haykin, 1994; Manevitz, Bitar, & Givoli, 2005). Detailed descriptions of learning algorithms can be found in Haykin (1994) and Demuth and Beale (2000).

A practical problem with NN is the selection of the correct complexity of the model, i.e., the correct number of hidden units or correct regularization parameters. The design of hidden layer is dependent on the selected learning algorithm. Supervised learning systems are generally more flexible in the design of hidden layers. A greater quantity of hidden layers enables a NN model to improve its closeness-of-fit, while a smaller quantity improves its smoothness or extrapolation capabilities. It was concluded that the number of hidden layers is heuristically set by determining the number of intermediate steps to translate the input variables into an output value (Choy et al., 2003). According to some literature studies, the number of hidden layer nodes can be up to (1) $2n + 1$ (where n is the number of nodes in the input layer), (2) 75% of the quantity of input nodes, or (3) 50% of the quantity of input and output nodes (Lenard, Alam, & Madey, 1995; Patuwo, Hu, & Hung, 1993; Piramuthu, Shaw, & Gentry, 1994).

A transfer function is needed to introduce the non-linearity characteristics into the network. The non-linear function will make the hidden units of multi-layer network more powerful that just plain perception. The used transfer function is a standard function for backpropagation, that is, the sigmoid transfer function. The sigmoid transfer function is chosen due to its ability to help the generalization of learning characteristics to yield models with improved accuracy (Choy et al., 2003).

The backpropagation training paradigm uses three controllable factors that affect the algorithm's rate of learning. They are the *learning rate coefficient*, *momentum* and the *exit conditions*. *Learning coefficient* governs the speed that the weights can be changed over time, reducing the possibility of any weight oscillation during the training cycle. *Momentum* parameter controls over how much iteration an error adjustment persists. There is no definitive rule regarding the momentum, in general it is set to 0.5 which is half of the maximum limit for training to reduce the damping effect. NNs use a number of different stopping rules to control the termination of the training process (Choy et al., 2003).

The residual entropy of the trained network is a measure of its generalization. It is monitored during training by means of the *mean square error* value. To generalize the NN architecture, a validation data set is applied to check the degree of generalization of the trained model and is evaluated whether the output is close enough for an input (Choy et al., 2003).

3.2. Neuro-fuzzy systems

System modeling based on conventional mathematical tools is not well suited for dealing with ill-defined and uncertain systems. By contrast, a fuzzy inference system (FIS) employing fuzzy if-then rules can model the qualitative aspects of human knowledge and reasoning process without employing precise quantitative analyses (Jang, 1993).

The numeric analysis approach to fuzzy systems was first presented by Takagi and Sugeno (1985) and then a lot of studies have been made (Figueiredo & Gomide, 1999; Jang, 1993; Nishina & Hagiwara, 1997; Takagi & Sugeno, 1985). Since the systems using the fuzzy set theory can express rules or knowledge as "if-then" form, they have advantages as they do not require mathematical analysis for modeling. However, they necessitate appropriate model construction and parameter selection. Elimination of unnecessary rules and selection of efficient input elements can contribute the performance improvement, reduction of calculation cost and analysis of the obtained rules that is one of the most important merits of fuzzy systems (Iyatomi & Hagiwara, 2004).

Combinations of neural networks and fuzzy systems (or *neuro-fuzzy systems* for short) have been recognized as a powerful alternative approach to develop fuzzy systems (Figueiredo & Gomide, 1999). In fuzzy systems, system input–output relationships are represented explicitly in the form of *if-then* rules whereas, in neural networks, the same relationships are not explicitly given, but are 'coded' in the network by its parameters. In contrast to knowledge-based techniques, no explicit knowledge is required in neural networks applications. On the other hand, neuro-fuzzy systems combine the semantic transparency of rule-based fuzzy systems with the learning capability of neural networks (Babuška & Verbruggen, 2003). Furthermore, neuro-fuzzy modeling has been recognized as a powerful tool that can facilitate the effective development of models by combining information from various sources, such as empirical models, heuristics and data. Hence, in most cases neuro-fuzzy models can be better used to explain solutions to users than completely black box models such as neural networks (Babuška & Verbruggen, 2003; Panchariya, Palit, Popovic, & Sharma, 2004).

In the recent past, a variety of neuro-fuzzy model architectures were proposed and described in studies belonging to Jang (1993), Nauck and Kruse (1999), Palit and Popovic (2000), Wang (1994). This study particularly focuses on the singleton type neuro-fuzzy model proposed by Wang (1994) in which an analytical expression is obtained for the output of the system versus the inputs and implemented by a neural network. The main feature of this model is that the number of input membership functions (fuzzy sets) is equal to the number of rules providing ease in implementation. The model was latter modified to Takagi–Sugeno (TS) type of neuro-fuzzy model by Palit and Babuška (2001) and was widely utilized in time-series predictions and system identifications.

Fuzzy modeling requires a method (1) for transforming human knowledge or experience into the rule base and database of a FIS and (2) for tuning the membership function to minimize the output error measure or maximize performance index (Jang, 1993). With this motivation, a novel architecture called *Adaptive Network-based Fuzzy Inference System* (ANFIS) was employed in this study.

3.3. Adaptive network-based fuzzy inference system

Adaptive network-based fuzzy inference system (ANFIS) can construct an input–output mapping based on both human knowledge in the form of fuzzy if-then rules with appropriate membership functions and stipulated input–output data pairs. It applies a neural network in determination of the shape of membership functions and rule extraction. ANFIS architecture uses a hybrid learning procedure in the framework of adaptive networks (Jang, 1993). This method plays a particularly important role in the induction of rules from observations within fuzzy logic (Matlab, 1996).

Suppose that the FIS under consideration has two inputs (x and y) and one output (z) assuming that the rule base contains two fuzzy *if-then* rules of Takagi and Sugeno's type (Takagi & Sugeno, 1983). The fuzzy model is based on a first order Sugeno polynomial that is generally composed of rules of the form:

Rule:1. If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$.
Rule:2. If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$.

where A_i and B_i are the fuzzy sets, f_i is the output set within the fuzzy region specified by the fuzzy rule p_i and q_i and r_i are the design parameters that are determined during the training process. Fig. 3 illustrates the fuzzy reasoning mechanism (Takagi & Sugeno, 1983).

ANFIS has a five layer feed forward neural network. The node functions in the same layer are of the same function family as described below (Jang, 1993):

Layer 1. Each node i in this layer has a node function as

$$O_i^l = \mu_{A_i}(x), \quad i = 1, 2. \tag{3}$$

Here, x is the input to node i, and A_i is the linguistic label (*small, large, etc.*) associated with this node function. In other words, O_i^l is the membership function of A_i and it specifies the degree to which the given x satisfies the quantifier A_i. Usually $\mu_{A_i}(x)$ is chosen as bell-shaped with a maximum value of 1 and a minimum value of 0, where a generalized bell function can be expressed as:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}}, \tag{4}$$

where $\{a_i, b_i, c_i\}$ is the membership function parameter set.

As the values of these parameters change, bell-shaped functions also alter accordingly, exhibiting various forms of membership functions on linguistic label A_i. Parameters in this layer are referred to as *premise parameters*.

Layer 2. Every node in this layer multiplies the incoming signals and sends the product out

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2. \tag{5}$$

Each node output represents the firing strength of a rule.

Layer 3. Every node in this layer calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths:

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \tag{6}$$

For the sake of simplicity, the outputs of this layer are called *normalized firing strengths.*

Layer 4. Every node in this layer has a node function as

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i), \quad i = 1, 2, \tag{7}$$

where $\bar{w}_i$ is the output of Layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. The parameters in this layer are referred to as *consequent parameters.*

Please cite this article in press as: Efendigil, T., et al. A decision support system for demand forecasting with artificial neural ... *Expert Systems with Applications* (2008), doi:10.1016/j.eswa.2008.08.058

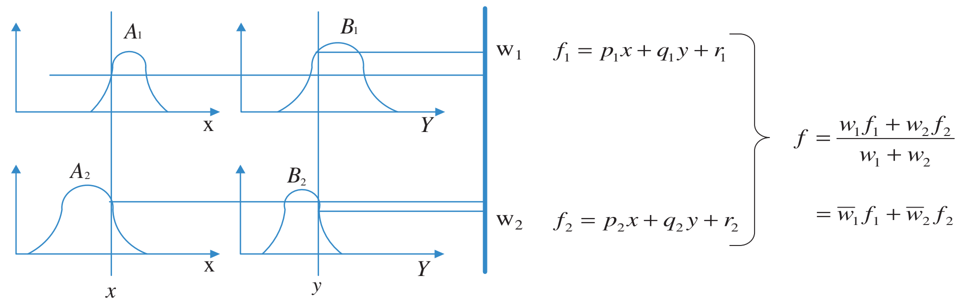


Fig. 3. Fuzzy reasoning mechanism.

Layer 5. The single node in this layer computes the overall output as the summation of all incoming signals:

$$O_i^5 = \text{overall output} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \tag{8}$$

The last layer has a single fixed node and its outputs have crisp characteristics.

In ANFIS structure, the premise and consequent parameters should be noted as important factors for the learning algorithm in which each parameter is utilized to calculate the output data of the training data. The premise part of a rule defines a subspace, while the consequent part specifies the output within this fuzzy subspace (Jang, 1993).

It is observed that given the values of premise parameters, the overall output can be expressed as linear combinations of the consequent parameters. More precisely, the output of the ANFIS model can be written as in Jang (1993):

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2. \tag{9}$$

Substituting Eq. (6) into Eq. (9) yields Eq. (10):

$$f = \bar{w}_1 f_1 + \bar{w}_2 f_2. \tag{10}$$

Substituting the fuzzy *if-then* rules into Eq. (10), the equation becomes:

$$f = \bar{w}_1(p_1x + q_1y + r_1) + \bar{w}_2(p_2x + q_2y + r_2). \tag{11}$$

.

After rearrangement, the output f can be expressed as in Eq. (12)

$$f = (\bar{w}_1x)p_1 + (\bar{w}_1y)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2x)p_2 + (\bar{w}_2y)q_2 + (\bar{w}_2)r_2. \tag{12}$$

The hybrid algorithm used in ANFIS structure consists of the *least squares method* and the *backpropagation gradient descent method* for training FIS membership function parameters to emulate a given training data (Matlab, 1996). The hybrid algorithm is composed of a forward pass and a backward pass. In the forward pass of the hybrid learning algorithm, the least squares method is used to optimize the consequent parameters with the premise parameters fixed. After the optimal consequent parameters are found, the backward pass starts immediately. In the backward pass of the algorithm, the gradient descent method is used to adjust optimally the premise parameters corresponding to the fuzzy sets in the input domain. The output of the ANFIS is calculated by employing the consequent parameters found in the forward pass. The output error is used to adapt the premise parameters by means of a standard backpropagation algorithm. In the following section, a numerical example is presented to explain the methodology.

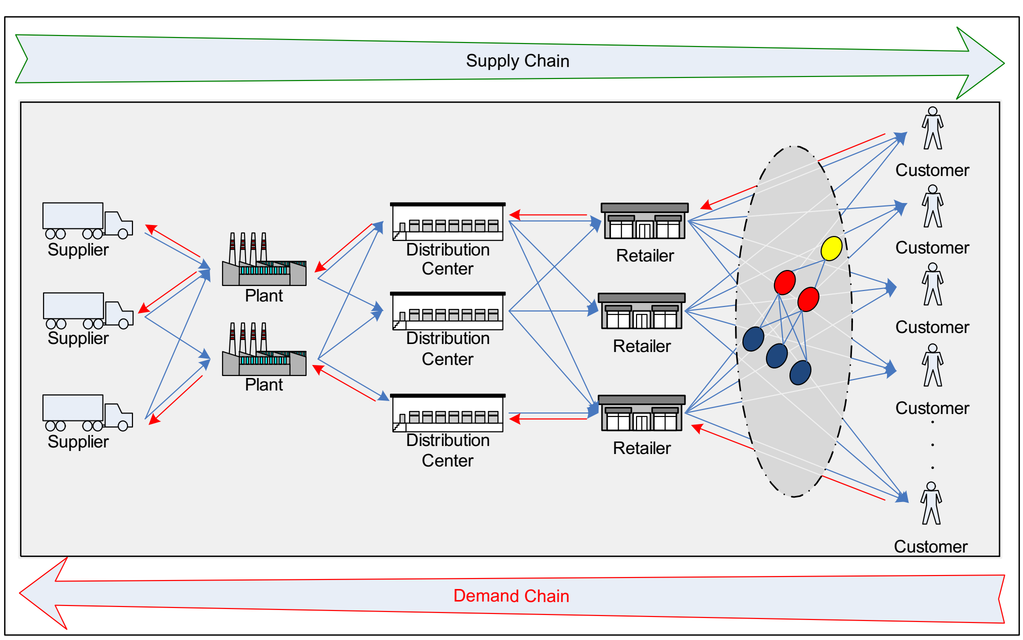


Fig. 4. A general supply-demand chain structure.

4. Problem Statement

A typical SC includes different levels of enterprises as shown in Fig. 4. The first level organization is retailer or market where products are presented and sold to customers; the second level organization is a distribution center or warehouse where products are delivered from plants to retailer; the third level organization is a plant where products are produced according to determined pro-

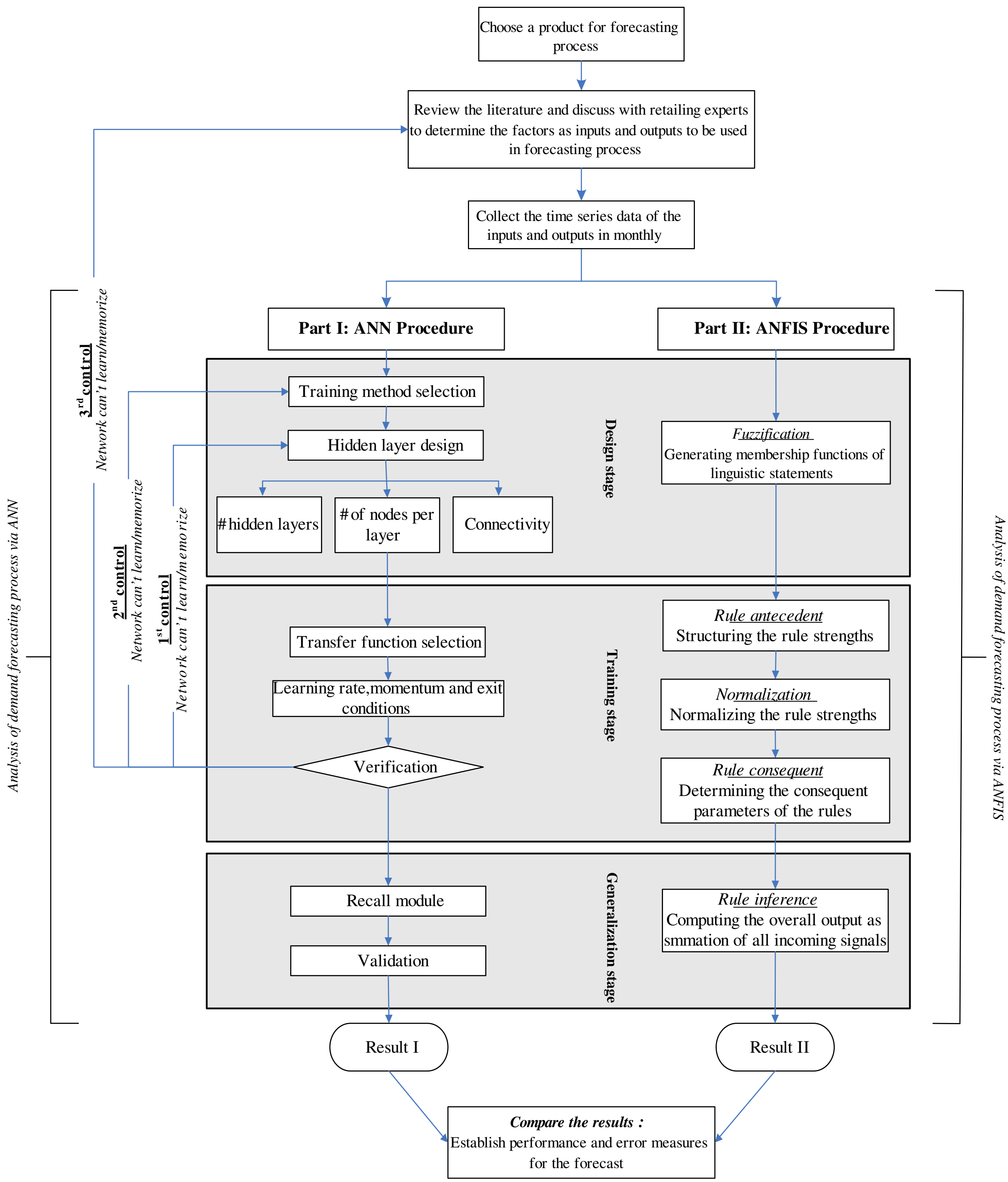


Fig. 5. The structural design of the proposed comparative analysis.

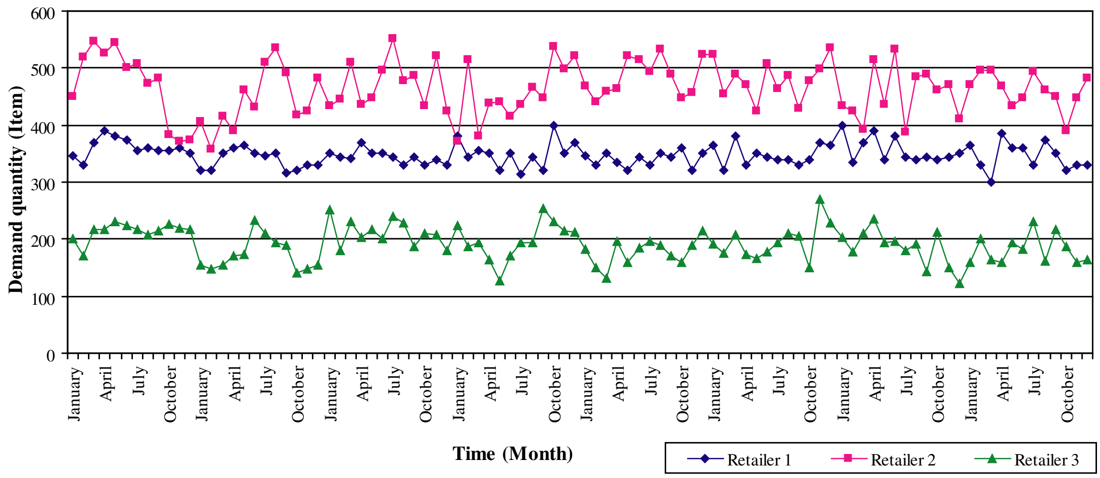


Fig. 6. Retailers and demand quantities for 96 monthly periods.

duction and inventory scheduling; the fourth level organization is a supplier where raw materials are transported to plants to be produced. Although SCs have many different aspects, we focused on customer side related to demand forecasting. Fig. 4 shows the supply and demand chain structure between the supply chain entities. The examined side of numerical example is also pointed out in Fig. 4.

This study presents a comparative analysis for the first level of SC which is modeled by artificial intelligence approaches including both ANN and ANFIS techniques in a fuzzy environment. A bipartite structural design is built for the proposed comparative analysis (see Fig. 5). In *Part I* (adapted from Choy et al. (2003)), MATLAB's Neural Network Toolbox (Demuth & Beale, 2000) was used for designing and training the *neural network*, and in *Part II*, the MATLAB's Fuzzy Logic Toolbox (Matlab, 1996) was used for defining and solving the *ANFIS* structure.

A demand forecasting problem of a real SC structure is analyzed and performed to illustrate the application of the proposed approach. A company serving in durable consumer goods industry in Istanbul was investigated to control the supply-demand relationship between retailers and customers. The distribution channel of this company is formed by three large retail firms. This study was performed for the cash register product which classified as consumer electronics by the company. Initially, two assumptions have been made as follows: Customer demand is valid for only one type product and each retailer has its own inventory control policy (periodical review system). The factors affecting the demand are determined in the light of the papers of Aburto and Weber (2007), Kuo and Xue (1998), Kuo (2001) and Kuo et al. (2002) according to experts from the retailers. The demand forecasting system for this study is built by the factors in the following:

- Unit sales price (input): Unit sales price is a competitive factor affecting the customer behaviors, especially for independent retailers. It is processed as quantitative information.
- Product quality (input): This factor includes the evaluation about product quality according to customers via a 1-9 scale. It is processed as qualitative information.
- Customer satisfaction level (input): This factor shows the sales and post-sales behaviors of retailers to the customers. It is processed as qualitative information.
- Effect of promotions, holidays and special days (input): This factor means the percent increase of sales related to promotions or special days (such as feasts, new year's day, etc.). It is processed as qualitative information.
- Demand quantity (output): Demand quantity is the quantity from customers to retailers. It is processed as quantitative information.

The data were obtained from each retailer via the questionnaire consisting of determined factors from literature and retailing experts mentioned above and the demand quantities from the customers. Twenty-four months data were gotten, however, these 24 months data were supposed to be inadequate for neural forecasting process. Therefore, Monte Carlo simulation was used to generate data considering the characteristics of samples. BestFit 4.5 software package was used for generating the data, while MS Excel was used for simulating and processing the data. The data for 96 monthly periods were generated for the proposed neural forecasting mechanism. Fig. 6 clearly shows that there is no periodically pattern such as trend or seasonality.

One of the most commonly used approaches for neural systems validation is called *data splitting*, which usually has three sets: *train*, *test*, and *check*. This method randomly splits the data into a *training data set*, a *testing data set*, and a *checking data set*. The training data set is used to construct the network, whereas the testing data set is employed to determine any model over fitting during the training, while the checking data set is used to check the generalization capability of the neural systems at each epoch. In this study, 96 generated samples were utilized for the comparative analysis. The proposed methodology was trained with 77 samples corresponding to 75% of the data set and tested with 19 samples

Table 1
Neural network model definition

Architecture	• Multi layer perceptron neural networks • Input neurons: 4 • Hidden Layers 1 or 2 • Hidden neurons: 7 or 12 or 25 • Output neuron: 1 • Learning algorithm: Levenberg–Marquardt optimization • Transfer functions: Sigmoid transfer function in hidden and output layers • Learning rates: 0.5 or 0.05 or 0.8 • Momentum rate: 0.7 • All input data normalized between 0 and 1
Computation/ termination	• Training: Backpropagation rule. • Training termination: Stop training when reaches a specified number of epochs or 0.01 error tolerance

Table 2
The possibilities of network architectures for *Part I* in the proposed methodology

Learning rate	# of hidden layer (HL)	# of hidden neurons		Activation function (AF)			Forecasted demand quantities			Training MSE (%)	Validation MAPE (%)
		In *HL* 1	In *HL* 2	*HL* 1 AF	*HL* 2 AF	Output layer AF	Retailer 1	Retailer 2	Retailer 3		
							Actual demand quantities				
							345	531	218		
0.5	1	7		tansig		logsig	302	389	190	0.006573	26.244
0.5	1	12		tansig		logsig	325	415	215	0.005724	24.704
0.5	***1***	***25***		***tansig***		***logsig***	***330***	***461***	***206***	***0.003945***	***18.754***
0.5	1	7		logsig		tansig	306	298	213	0.006708	29.221
0.5	1	12		logsig		tansig	314	358	226	0.005412	25.292
0.5	1	25		logsig		tansig	356	427	211	0.004667	24.417
0.5	1	7		logsig		logsig	355	405	197	0.006319	22.537
0.5	1	12		logsig		logsig	350	380	179	0.005008	25.274
0.5	1	25		logsig		logsig	326	416	194	0.004167	19.312
0.5	1	7		tansig		tansig	308	403	203	0.006805	20.571
0.5	1	12		tansig		tansig	346	383	187	0.006241	25.765
0.5	1	25		tansig		tansig	261	230	249	0.005589	34.410
0.05	1	7		tansig		logsig	315	357	198	0.007691	24.195
0.05	1	12		tansig		logsig	322	399	201	0.007075	21.237
0.05	1	25		tansig		logsig	319	401	227	0.005447	24.754
0.05	1	7		logsig		tansig	339	261	182	0.007150	30.266
0.05	1	12		logsig		tansig	289	352	175	0.004929	30.385
0.05	1	25		logsig		tansig	N/A	N/A	N/A	N/A	N/A
0.05	1	7		logsig		logsig	360	355	198	0.005496	22.061
0.05	1	12		logsig		logsig	344	411	241	0.005840	18.829
0.05	1	25		logsig		logsig	313	395	153	0.006465	24.448
0.05	1	7		tansig		tansig	330	338	226	0.005108	19.152
0.05	1	12		tansig		tansig	290	334	186	0.005505	27.850
0.05	1	25		tansig		tansig	164	368	280	0.006320	38.752
0.8	1	7		tansig		logsig	308	401	187	0.006158	21.584
0.8	1	12		tansig		logsig	345	362	189	0.005572	22.665
0.8	1	25		tansig		logsig	312	362	187	0.004018	27.179
0.8	1	7		logsig		tansig	356	377	223	0.008308	19.655
0.8	1	12		logsig		tansig	333	329	247	0.006461	24.436
0.8	1	25		logsig		tansig	251	437	221	0.005041	27.505
0.8	1	7		logsig		logsig	306	399	137	0.005082	25.044
0.8	1	12		logsig		logsig	302	380	148	0.004667	36.792
0.8	1	25		logsig		logsig	341	423	195	0.005306	23.508
0.8	1	7		tansig		tansig	353	412	210	0.007819	27.249
0.8	1	12		tansig		tansig	327	265	245	0.006449	33.743
0.8	1	25		tansig		tansig	368	244	287	0.007201	32.590
0.5	2	7	3	tansig	tansig	logsig	327	419	163	0.006428	27.528
0.5	2	3	7	logsig	logsig	logsig	316	363	167	0.007389	29.606
0.05	2	7	3	logsig	logsig	logsig	322	359	203	0.007707	50.401
0.8	2	7	3	logsig	logsig	logsig	324	381	162	0.008192	23.469

(approximately 25% of the data set) that were selected in random from the data set.

The employed training errors are the *mean squared error* (*MSE*) of the training data set at each epoch and the *mean absolute percentage error* (*MAPE*) of the checking data set at each time. If Y_t is the actual observation for time period t and F_t is the forecast for the same period, then *MSE* and *MAPE* are defined as in Eqs. (13) and (14) (Makridakis, Wheelwright, & Hyndman, 1998);

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(Y_t - F_t)^2, \tag{13}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\left(\frac{Y_t - F_t}{Y_t}\right) * 100\right|. \tag{14}$$

5. Results and validation

As seen in Fig. 5, after collecting monthly time-series data, *Part I* was applied to have *Result I*. In *Part I*, the network with 4 input neurons and 1 output neuron was trained for 1000 epochs. The rule of "*stop it after a specified number of epochs*" or "*stop it when error tolerance value reaches 0.01*" was applied for the training process.

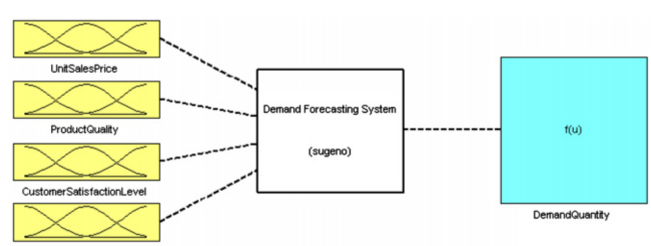


Fig. 7. Demand forecasting system for ANFIS structure.

The model was trained with "*Levenberg–Marquardt optimization*" learning algorithm. The network was simulated for various numbers of hidden layers and hidden neurons in order to minimize the error between actual and forecasted values. Besides, various learning rate values were utilized whereas a fixed value of momentum coefficient was used as 0.7. To train the different network architectures, both *hyperbolic tangent sigmoid* and *logarithmic sigmoid transfer functions* were applied. Identifying a set of parameter combinations leading to optimum neural network performance for

Table 3
ANFIS results related to various alternative structures

Input function		Output function					
Function type		Constant			Linear		
		Training error	Test error	Difference	Training error	Test error	Difference
Retailer 1							
2 Membership function	triMF	2.181	5.178	2.997	0.767	24.672	23.906
	trapMF	2.945	5.316	2.371	0.873	19.641	18.768
	gbellMF	2.436	5.093	2.657	0.676	11.734	11.059
	gaussMF	2.270	***4.781***	2.612	0.669	13.279	12.610
3 Membership function	triMF	1.348	58.532	57.184	0.424	59.556	59.132
	trapMF	2.043	80.684	78.641	0.423	83.397	82.965
	gbellMF	1.067	27.484	26.406	0.285	767.765	767.480
	gaussMF	1.165	17.606	16.441	0.286	1014.847	1014.562
4 Membership function	triMF	0.552	79.865	79.314	0.261	105.987	105.726
	trapMF	1.293	104.607	103.314	0.074	103.963	103.889
	gbellMF	0.328	67.236	66.908	0.232	421.256	421.024
	gaussMF	0.399	70.395	69.995	0.260	427.111	426.852
Retailer 2							
2 Membership function	triMF	0.452	***0.570***	0.118	0.315	1.810	1.495
	trapMF	11.963	21.8472	9.884	0.303	8.3549	8.051
	gbellMF	7.450	12.3113	4.862	0.248	24.2232	23.975
	gaussMF	4.269	7.2142	2.945	0.253	13.8125	13.559
3 Membership function	triMF	0.297	7.314	7.017	0.199	19.1171	18.918
	trapMF	3.361	124.3983	121.038	0.212	99.1831	98.971
	gbellMF	0.912	39.011	38.099	0.195	328.1107	327.916
	gaussMF	0.474	22.9938	22.520	0.165	526.1986	526.033
4 Membership function	triMF	0.264	204.1142	203.850	0.157	204.6106	204.454
	trapMF	4.181	235.3258	231.145	0.105	241.5942	241.489
	gbellMF	0.175	179.7655	179.591	0.312	1781.8066	1781.494
	gaussMF	0.188	157.9085	157.721	0.150	779.6213	779.471
Retailer 3							
2 Membership function	triMF	1.920	3.034	1.114	1.565	194.086	192.521
	trapMF	2.342	3.677	1.336	1.241	12.128	10.888
	gbellMF	2.138	3.171	1.032	1.332	26.015	24.683
	gaussMF	2.039	***3.001***	0.962	1.334	192.009	190.675
3 Membership function	triMF	1.486	28.819	27.33	1.042	28.712	27.671
	trapMF	1.925	32.839	30.913	0.971	41.618	40.67
	gbellMF	1.676	22.958	21.283	1.394	237.633	236.239
	gaussMF	1.643	32.935	31.292	0.986	119.561	118.574
4 Membership function	triMF	1.529	59.488	57.959	0.952	58.583	57.631
	trapMF	1.732	72.727	70.996	1.018	103.423	102.405
	gbellMF	1.136	59.586	58.450	0.968	217.085	216.117
	gaussMF	1.033	46.241	45.208	0.982	102.129	101.147

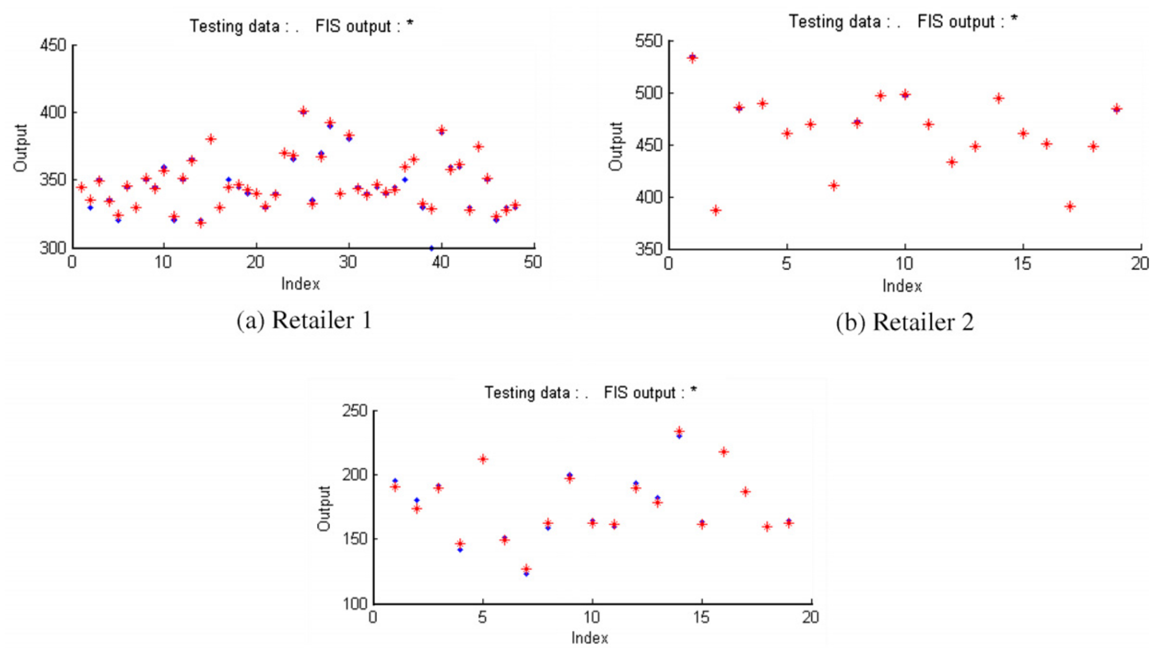
this problem was done using the "change one factor at a time" method of experimentation, or by trial and error. The different combinations of this experimental study of forecasting demand values are summarized in Table 1.

In this study, the designed network model was run ten times for each different network architecture. The average of *MSE*s was calculated for the training processes. After testing the oscillation with the test data, a checking data set was prepared to validate and check the generalization the network's capability according to the difference of the actual and forecasted data. Using a fresh and unseen data set, to generalize a network model is an important principle. Therefore, a new training process with a new data set (can be named as checking data set) was run ten times employing *Levenbert–Marquardt* algorithm. Basically, the better model has the lower error in this study. Hence, depending on the results, as shown in Table 2, a *4-25-1* network architecture with 0.5 learning rate has the lower errors regarding to the values of *MSE* and *MAPE*. Here the obtained values as 330 (for retailer 1), 461 (for retailer 2) and 206 (for retailer 3) are the forecasted values named as *Result I* in the proposed methodology.

In *Part II*, ANFIS structure with four input neurons and one output neuron with 0.01 error tolerance for each retailer is illustrated. Fig. 7 depicts the demand forecasting system according to Sugeno approach. The structure using "product" function for linking the rules together and "weighted average" for defuzzification was trained for 30 epochs.

The neuro-fuzzy model was run ten times for each model parameter with varying numbers and types of input–output membership functions (MFs) considering the over fitting of the model with constructed 16 rules. *Triangular-shaped-built-in MF* (triMF), *trapezoidal-shaped-built-in MF* (trapMF), *generalized bell-shaped built-in MF* (gbellMF) and *gaussian curve built-in MF* (gaussMF) were utilized as the MF types with the numbers of 2-3-4 MFs for input functions. Output functions were evaluated according to the characteristics of being *constant* or *linear*. Table 3 shows the results of the simulation study to find the best definition of the constructed ANFIS structure.

First-order Sugeno type systems were found to be more appropriate for this model. In addition, the all best-responding models to this neuro-fuzzy system have the two membership function, but different function types as: *gaussMF* with *4.781* test error for retailer 1, *triMF* with *0.570* test error for retailer 2, and *gaussMF* with *3.001* test error for retailer 3. The testing data set and the fuzzy inference output fitness are given for three retailers in Fig. 8.



(a) Retailer 1 (b) Retailer 2 (c) Retailer 3

Fig. 8. The testing data and fuzzy inference output fitness for three retailers.

Table 4
Control data set and average MAPE values for Retailer 1-2-3

	Actual value	Forecasted value	Average *MAPE* (%)
Retailer 1	345	341	4.88
Retailer 2	531	531	7.05
Retailer 3	218	218	2.41

Neuro-fuzzy systems also require a validation process to validate and generalize the constructed structure. Therefore a validation approach was performed for *Part II*, like in *Part I*. A fresh and unseen control data set was used to measure the generalization capability of the model. Table 4 shows the actual and forecasted values with the average value of *MAPE* for the control data set related to the difference between forecasted and actual data for each control data.

According to the proposed methodology (Fig. 5), *Result* I and *Result II* should be compared and evaluated to have a closest forecasting value to an actual value. When Tables 2 and 4 are compared, it is significantly seen that ANFIS structure gives closer forecasted values than the actual values.

6. Conclusions

This study has developed a comparative forecasting mechanism based on ANN and ANFIS techniques to manage the demand forecasting issue under fuzziness. The mechanism aims to utilize both neural networks and fuzzy modeling to compare the reasoning processes for demand estimation. Therefore, a bipartite methodology which applies the ANN and ANFIS approaches is presented to obtain more accurate forecasts. To demonstrate the effectiveness of the proposed methodology, demand forecasting issue was investigated on a SC of a white goods company as a real-world case study. Evaluation results indicate that ANFIS method performs more effectively than ANN structure in estimation of the more reliable forecasts for our case. To our knowledge, this is the first study that achieves to bring a new perspective to academic literature in making forecasting with artificial intelligence methods. With this motivation, the proposed methodology can be considered as a successful decision support tool in forecasting customer demands. Future research will perform various ANN types and aforementioned neuro-fuzzy systems to make a similar approach.

References

Aburto, L., & Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing, 7*, 136–144.
Al-Saba, T., & El-Amin, I. (1999). Artificial neural networks as applied to long-term demand forecasting. *Artificial Intelligence in Engineering, 13*, 189–197.
Ayata, T., Çam, E., & Yıldız, O. (2007). Adaptive neuro-fuzzy inference systems (ANFIS) application to investigate potential use of natural ventilation in new building designs in Turkey. *Energy Conversion and Management, 48*, 1472–1479.
Babuška, R., & Verbruggen, H. (2003). Neuro-fuzzy methods for linear system identification. *Annual Reviews in Control, 27*, 73–85.
Beccali, M., Cellura, M., Lo Brano, V., & Marvuglia, A. (2004). Forecasting daily urban electric load profiles using artificial neural networks. *Energy Conversion and Management, 45*, 2879–2900.
Cavalieri, S., Maccarrone, P., & Pinto, R. (2004). Parametric vs. neural network models for the estimation of production costs: A case study in the automotive industry. *International Journal of Production Economics, 91*, 165–177.
Chen, F., Drezner, Z., Ryan, J. K., & Simchi-Levi, D. (2000). Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Management Science, 46*(3), 436–443.
Chen, F., Ryan, J. K., & Simchi-Levi, D. (2000). The impact of exponential smoothing forecasts on the bullwhip effect. *Naval Research Logistics, 47*(4), 269–286.
Chiu, M., & Lin, G. (2004). Collaborative supply chain planning using the artificial neural network approach. *Journal of Manufacturing Technology Management, 15*(8), 787–796.
Chopra, S., & Meindl, P. (2001). *Supply chain management: Strategy, planning and operation*. NJ: Prentice-Hall.
Choy, K. L., Lee, W. B., & Lo, V. (2003). Design of an intelligent supplier relationship management system: A hybrid case based neural network approach. *Expert Systems with Applications, 24*, 225–237.
Demuth, H., & Beale, M. (2000). *Neural network toolbox user's guide*. Natick: The Math Works Inc.
Dhar, V., & Stein, R. (1997). *Seven methods for transforming corporate data into business intelligence*. NJ: Prentice-Hall.
Du, T. C. T., & Wolfe, P. M. (1997). Implementation of fuzzy logic systems and neural networks in industry. *Computers in Industry, 32*, 261–272.
Efendigil, T., Önüt, S., & Kongar, E. (2008). A holistic approach for selecting a third-party reverse logistics provider in the presence of vaguness. *Computers and Industrial Engineering, 54*(2), 269–287.

Escoda, I., Ortega, A., Sanz, A., & Herms, A. (1997). Demand forecast by neuro-fuzzy techniques. In *Proceedings of the sixth IEEE international conference on fuzzy systems* (pp. 1381–1386).

Figueiredo, M., & Gomide, F. (1999). Design of fuzzy systems using neurofuzzy networks. *IEEE Transactions on Neural Networks, 10*(4), 815–827.

Gaafar, L. K., & Choueiki, M. H. (2000). A neural network model for solving the lot-sizing problem. *Omega, 28*, 175–184.

Garetti, M., & Taisch, M. (1999). Neural networks in product ion planning and control. *Production Planning and Control, 10*(4), 324–339.

Haykin, S. (1994). *Neural networks a comprehensive foundation*. New York: MacMillan.

Hobbs, B. F., Helman, U., Jitprapaikulsarn, S., Konda, S., & Maratukulam, D. (1998). Artificial neural networks for short-term energy forecasting: Accuracy and economic value. *Neurocomputing, 23*, 71–84.

Iyatomi, H., & Hagiwara, M. (2004). Adaptive fuzzy inference neural network. *Pattern Recognition, 37*(10), 2049–2057.

Jang, J. S. R. (1993). ANFIS: Adaptive network based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics, 23*(3), 665–684.

Koskivaara, E. (2004). neural networks in analytical review procedures. *Managerial Auditing Journal, 19*(2), 191–223.

Kuo, R. J. (1998). A decision support system for the stock market through integration of fuzzy neural networks and fuzzy Delphi. *Applied Artificial Intelligence, 12*, 501–520.

Kuo, R. J., & Xue, K. C. (1998). An intelligent sales forecasting system through integration of artificial neural network and fuzzy neural network. *Computers in Industry, 37*, 1–15.

Kuo, R. J. (2001). A sales forecasting system based on fuzzy neural network with initial weights generated by genetic algorithm. *European Journal of Operational Research, 129*, 496–517.

Kuo, R. J., Wu, P., & Wang, C. P. (2002). An intelligent sales forecasting system through integration of artificial neural networks and fuzzy neural networks with fuzzy weight elimination. *Neural Networks, 15*, 909–925.

Law, R., & Au, N. (1999). A neural network model to forecast Japanese demand for travel to Hong Kong. *Tourism Management, 20*, 89–97.

Law, R. (2000). Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. *Tourism Management, 21*, 331–340.

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). The bullwhip effect in supply chains. *Sloan Management Review, 38*, 93–102.

Leernders, M. R., Fearon, H. E., Flynn, A. E., & Johnson, P. F. (2002). *Purchasing and supply management* (12th ed.). New York: McGraw-Hill.

Lenard, M. J., Alam, P., & Madey, G. R. (1995). The application of neural networks and a qualitative response model to the auditor's going concern uncertainty decision. *Decision Sciences, 26*(2), 209–227.

Leung, H. C. (1995). Neural networks in supply chain management. In *Proceedings of the '95 engineering management conference* (pp. 347–352).

Luxhøj, J. T., Riis, J. O., & Stensballe, B. (1996). A hybrid econometric-neural network modeling approach for sales forecasting. *International Journal of Production Economics, 43*, 175–192.

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting: Methods and applications* (3rd ed.). New York: John Wiley & Sons.

Manevitz, L., Bitar, A., & Givoli, D. (2005). Neural network time series forecasting of finite-element mesh adaptation. *Neurocomputing, 63*, 447–463.

Matlab (1996). Fuzzy logic toolbox user's guide. Natick: The Math Works Inc.

Metaxiotis, K., & Psarras, J. (2003). Neural networks in production scheduling: Intelligent solutions and future promises. *Applied Artificial Intelligence, 17*, 361–373.

Nauck, D., & Kruse, R. (1999). Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Systems, 101*, 261–271.

Nishina, T., & Hagiwara, M. (1997). Fuzzy inference neural network. *Neurocomputing, 14*, 223–239.

Palit, A. K., Popovic, D. (2000). Nonlinear combination of forecasts using artificial neural network, fuzzy logic and neuro-fuzzy approaches. In *The ninth IEEE international conference on fuzzy systems* (pp. 566–571).

Palit, A. K., Babuška, R. (2001). Efficient training algorithm for Takagi–Sugeno type neuro-fuzzy network. In *Proceedings of FUZZY-IEEE* (pp. 1367–1371).

Panchariya, P. C., Palit, A. K., Popovic, D., & Sharma, A. L. (2004). Nonlinear system identification using Takagi–Sugeno type neuro-fuzzy model. In *Second international IEEE conference on intelligent systems* (pp. 76–81).

Patuwo, E., Hu, M. Y., & Hung, M. S. (1993). Classification using neural networks. *Decision Sciences, 26*(6), 749–779.

Piramuthu, S., Shaw, M., & Gentry, J. (1994). A classification approach using multi-layered neural networks. *Decision Support Systems, 11*(5), 509–525.

Sabuncuoglu, I. (1998). Scheduling with neural networks: A review of the literature and new research directions. *Production Planning and Control, 9*(1), 2–12.

Shervais, S., Shannon, T. T., & Lendaris, G. G. (2003). Intelligent supply chain management using adaptive critic learning. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 33*(2), 235–244.

Sözen, A., Arcaklioğlu, E., & Özkaymak, M. (2005). Turkey's net energy consumption. *Applied Energy, 81*(2), 209–221.

Takagi, T., & Sugeno, M. (1983). Derivation of fuzzy control rules from human operator's control actions. *Proceedings of the IFAC symposium on fuzzy information, knowledge representation and decision analysis*, 55–60.

Takagi, T., & Sugeno, M. (1985). Structure identification of systems and its application to modeling and control. *IEEE Transactions on Systems Man and Cybernetics, 15*, 116–132.

Vellido, A., Lisboa, P. J. G., & Vaughan, J. (1999). Neural networks in business: A survey of applications (1992–1998). *Expert Systems with Applications, 17*, 51–70.

Wang, L. X. (1994). *Adaptive fuzzy systems and control: Design and stability analysis*. Englewood Cliffs, NJ: Prentice Hall.

Wei, S., Zhang, J., & Li, Z. (1997). A supplier-selecting system using a neural network 1997. In *IEEE international conference on intelligent processing systems* (pp. 468–471).

Wong, B. K., Bodnovich, T. A., & Selvi, Y. (1997). Neural network applications in business: A review and analysis of the literature (1988–1995). *Decision Support Systems, 19*, 301–320.

Wong, B. K., & Selvi, Y. (1998). Neural network applications in finance: A review and analysis of literature (1990–1996). *Information and Management, 34*, 129–139.

Wong, B. K., Lai, S. V., & Lam, J. (2000). A bibliography of neural network business applications research: 1994–1998. *Computers & Operations Research, 27*, 1045–1076.

Zhao, X., Xie, J., & Leung, J. (2002). The impact of forecasting model selection on the value of information sharing in a supply chain. *European Journal of Operational Research, 142*, 321–344.

Please cite this article in press as: Efendigil, T., et al. A decision support system for demand forecasting with artificial neural ... *Expert Systems with Applications* (2008), doi:10.1016/j.eswa.2008.08.058