An Autonomous Collaborative Forecasting System Implementation – The First Step towards Successful CPFR System

Chi-Fang Huang, Yun-Shiow Chen, and Yun-Kung Chung

Abstract**—**In the past decade, artificial neural networks (ANNs) have been regarded as an instrument for problem-solving and decision-making; indeed, they have already done with a substantial efficiency and effectiveness improvement in industries and businesses. In this paper, the Back-Propagation neural Networks (BPNs) will be modulated to demonstrate the performance of the collaborative forecasting (CF) function of a Collaborative Planning, Forecasting and Replenishment (CPFR®) system. CPFR functions the balance between the sufficient product supply and the necessary customer demand in a Supply and Demand Chain (SDC). Several classical standard BPN will be grouped, collaborated and exploited for the easy implementation of the proposed modular ANN framework based on the topology of a SDC. Each individual BPN is applied as a modular tool to perform the task of forecasting SKUs (Stock-Keeping Units) levels that are managed and supervised at a POS (point of sale), a wholesaler, and a manufacturer in an SDC. The proposed modular BPN-based CF system will be exemplified and experimentally verified using lots of datasets of the simulated SDC. The experimental results showed that a complex CF problem can be divided into a group of simpler sub-problems based on the single independent trading partners distributed over SDC, and its SKU forecasting accuracy was satisfied when the system forecasted values compared to the original simulated SDC data. The primary task of implementing an autonomous CF involves the study of supervised ANN learning methodology which aims at making "knowledgeable" decision for the best SKU sales plan and stocks management.

Keywords**—**CPFR, artificial neural networks, global logistics, supply and demand chain.

I. Introduction

IN SDC, the three parties, as shown in Fig. 1, vendors, manufacturers, and distributors (buyers), characterize interactive tasks to produce and deliver commodities, and meanwhile to satisfy consumer demands for the commodities; this interactive process with both (bottom-up) demand information and (top-down) supply material flows has inherent opportunities for the efficiency and effectiveness improvement of SDC. They can be discovered by integrating SDC flows, sharing flow information, communicating task interactions and supervising goods transactions, collaboratively. The collaborations result the above three parties into a single "virtual" unified enterprise in which its management information can be shared and then the loading of each SDC task can be levered.

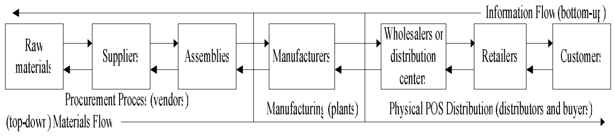


Fig. 1 The three parties of SDC

One way to achieve this SDC leverage process or to streamline the SDC pipeline (or network) shown in Fig. 1 is to implement a CPFR® system. (CPFR® is a Registered Trademark of the Voluntary Inter-industry Commerce Standards (VICS) Association, a retailer trade association, responsible for promoting common standards and business processes used to the continuous pursuit of improving both the customer satisfaction and the efficiency of business trade relationships, e.g., the speed and accuracy by which goods can be manufactured, distributed, and sold to consumers [1].) The overall goal of CPFR is the minimization of total cost or the maximization of total profit of the SDC operations. This goal can be accomplished by balancing the sufficiency of products supply and the necessity of customer demands in analyzing a CPFR process. Why and how?

CPFR, as its name implies, responds to the conduct of an enterprise's planning, forecasting and replenishment by drawing its supplying products with collaborative operations of all trading partners through the distributed channels in an SDC. This collaborative environment makes CPFR form a group of trading partners who consent to mutually perform and update the business conducts and operations, such as the joint sales and operational plans, the unified order and sales forecasting and the accurate time-based replenishment plans, etc., and also lets these partners be aware of their operations information one another so that SKU exceptions or changes occurred in the events, such as calendar displays, market demand, on-sales promotions or business policies, can be jointly and timely managed and corrected for keeping the SDC balance all the time; furthermore, the costly after-the-fact adjustments for the operation corrections can be avoided or reduced. To this end, a prediction mechanism for the advanced adjustments of the

Chi-Fang Huang is a graduate student with the Industrial Engineering Department, Yuan Ze University, Chung Li, Taiwan.
Yun-Shiow Chen, Ph.D., is a professor with the Industrial Engineering Department, Yuan Ze University, Chung Li, Taiwan (e-mail: ieyschen@saturn.yzu.edu.tw).
Yun-Kung Chung, Ph.D., is an associate professor with the Industrial Engineering Department, Yuan Ze University, Chung Li, Taiwan (e-mail: ieychung@saturn.yzu.edu.tw).

ISSN: 2517-9950
Vol:2, No:11, 2008

CPFR operations is necessary. Collaborative forecasting (CF) of CPFR exactly plays the key role for this necessity.

CF is the SKU sales prediction for distributed partner demands characterized by SKU inventory, promotion, season, cycle, trend, irregularity, and time period, which they also are the factors associated with the SDC balance. Once the factors can be monitored, CF can be modeled and help trading partners to mutually generate the most accurate forecasts so that the whole enterprise production conducts and material replenishments can be controllable and stable. Stable SKU positions lead to a controllable bullwhip effect [2]. Yet, since the SKU levels greatly count on time, and other marketing factors, a conventional forecasting method may not be good enough for implementing a practical CF system that meets the dynamic behavior of SDC.

Caro and Gallian [3] considered using demand learning with Bayesian theory and dynamic programming (DP) technique to investigate the dynamic SDC process, but the sophisticated mathematic development causes to the difficulty of practical application and implementation. Seifert [4] and Caridi, et al. [5] presented that those dynamic (time-based) SDC factors had better be modeled and operated with autonomy for the easy and correct on-line supervision of the SDC balance. This paper will present such an autonomous CF system, which is an extensible architecture based on ANNs that were implemented in authors' previous work [6], to perform the prediction for the real-time supervision of SKU inventory and sales status in the collaborative trading circumstance. The leaning ability of ANN conducts the autonomy of the CF.

ANN is a computational model constructed by mimicking the brain learning process to solve miscellaneous problems just like human do. A unique virtue of ANN is its ability to learn the relationship between input examples (feature vectors) and output information (solution) by means of repeatedly presenting examples to it. This relationship learning process is performed through the adaptation of the strengths (weights) connected between neurons of each layer within ANN. Mathematical optimization calculations and calculus derivatives for the weight adaptation cause ANN to learn (memorize) the examples presented. This learnability is one of ANN research area and will be introduced in latter section.

In this paper, the CF function of CPFR will be implemented with a group of modularized BPNs in which one BPN module accounts for the forecast and supervision of the SKU inventories at a POS or at any trading partner location. The number of the modular BPN needed for the CF fulfillment is equal to participating partners in the same CF system. The presented BPN-based autonomous CF system is an experimented one but with prospective substance. With regard to other two functions, collaborative planning (CP) and collaborative replenishment (CR), they will be further evolved with the current autonomous CF system to become a complete CPFR system in the future.

Toward the end of an autonomous CF system implementation, the remainder of this paper is organized as follows. First, an introduction to CPFR is presented, then the BPN learning procedure is addressed; third, the reasons why using BPN to implement the autonomy of the CF are explained and then the relative literature is reviewed; fifth, the method used to implement the BPN-based CF system is developed; sixth, the simulated data and experimental results are presented; final, the conclusions and directions for further research are outlined.

II. Introduction to CPFR

This section offers a bird's-eye view of a CPFR system formed from individual buyer to vendors through the relative manufacturers. As aforementioned, CPFR is performed by the mutual cooperation that functions as the improvement of the accuracy of the marketing demand forecasts and the further maintenance of SKU inventory positions and replenishments. The reciprocity collaboration aims at the balance of the SDC leverage. To balance the SDC leverage, which is generated by the information flow pulled from customers and by the materials flow pushed to customers, CPFR, unlike other traditional logistics forecasting models, employees the idea of CF to fulfill relative SKU replenishment and vendor-managed inventory (VMI) policy, including SKU sales and inventories at POSs, SKU orders of the outlets to distributors, SKU orders of the distribution centers (wholesales) to the manufactures and SKU orders of the manufactures to the suppliers, by doing so to meet the requirements of eventual customers [7].

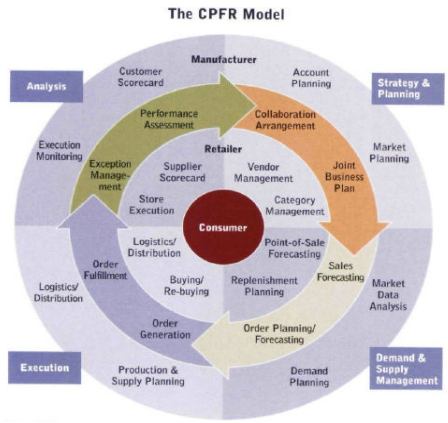


Fig. 2 The standard CPFR model

Traditionally, an individual trading partner takes charge of his own forecasting numbers that are analyzed independent of one another, thus there is no surprise to find that the trading partners do not have much in common. Call this inconsistent situation the bullwhip effect of which the result is that a minor discrepancy occurred at any stage of the business running cycle will balloon into a formidable misstep over time [2] [8]. Besides, the managerial reconciliation of time lags in traditional logistics forecasting methods involves a difficult communication problem. When demand is concentrated on a short selling season, the three parties of SDC will be impelled to take an inventory level far in advance of the peak selling period. This advanced inventory is with highly risky. By taking advantage of the on-line visible and sharable CPFR data, the bullwhip effect can be smoothed or even vanished [9].

The mechanism that performs the above bullwhipped inventory smoothing process in SDC is first to accomplish the most accurate CF for SKUs needed by trading partners, and then to set the efficient replenishment processes according to the information generated by the CF. Finally, it will be timely to ask suppliers for the acquisition of their products sufficient for matching the marketing necessity (requirements) the CF forecasted. In other words, a CPFR model cares elapsed time between consumption and production, and meanwhile reconciles supply availability with demand [10]. If how much inventory should be stockpiled can be properly planned and determined by this CPFR mechanism, the maximum sales while minimizing the risk of carryover can be obtained.

The standard CPFR model is shown in Fig. 2 that was copied from the website of the VICS Association [1]. The model is quaternary, including the four constitutes, strategy and planning, demand and supply management, execution and analysis. The core element is customer; the interactive missions are radiated from this core outside to business strategic planning based on a clockwise cycle. The following 9 steps used to perform the CPFR cycle was presented by Harrington [11]:

Step 1 - Develop Front-End Agreement
Step 2 - Create Joint Business Plan
Step 3 - Create Sales Forecast
Step 4 - Identify Exceptions for Sales Forecast
Step 5 - Resolve/Collaborate on Exception Items
Step 6 - Create Order Forecast
Step 7 - Identify Exceptions for Order Forecast
Step 8 - Resolve/Collaborate on Exception Items
Step 9 - Order Generation.

These general 9 steps can be iteratively performed and revised according to the factual status of executing the CPFR process.

Eventually, it should be mentioned that there is rarely an intact or optimal CPFR that stands for the best leverage balance of responsiveness to the forecasted customer needs and the productive SKU supplies. Although there is no exact CPFR method, some methods are distinctly better than others, and some are so competitive that when one weighs pros and cons [9]. For these reasons, CPFR is both frustrating and highly rewarding. More detailed CPFR discussion can be read in [4] [7] [12] and [13].

III. Introduction to Artificial Neural Networks

As the description in Section 1, ANN learning is a type of learning from examples. The examples input to an ANN will be learned or memorized, and then transformed to become the connected strength in terms of weight values; after the transformation, the weighted memorization will be utilized to recall, infer or generalize the solution to the new problem that was unlearned or unseen before. Two major sorts of typical ANN learning algorithm are supervised and unsupervised. The following introduction to ANN learning algorithms was summarized from [14] [15] and [16].

A supervised ANN learning is performed by an optimization technique to minimize the differences between the paired relationships of the input examples to the corresponding desired target vectors. Once this “difference-minimization” learning completed appropriately, the learned ANN is able to distinguish or generalize different input feature vectors unseen or seen before. In contrast, an unsupervised ANN learning algorithm does not consider target output vectors as a supervisor used to conduct the minimization of differences, but generally takes an internal weight adaptive mechanism that is able to iteratively “self-minimize” the differences between the output information and the connection weights for discovering emergent output characteristics. Fig. 3 shows a general ANN structure, which is called Back-Propagation Network (BPN). If the desired target vector $D^P(t)$ is considered, the ANN is supervised; if not, it becomes an unsupervised ANN. In this paper, the supervised BPN is used as a module to construct the proposed modular CF architecture based on the topology of a SDC network. In what follows, BPN is introduced.

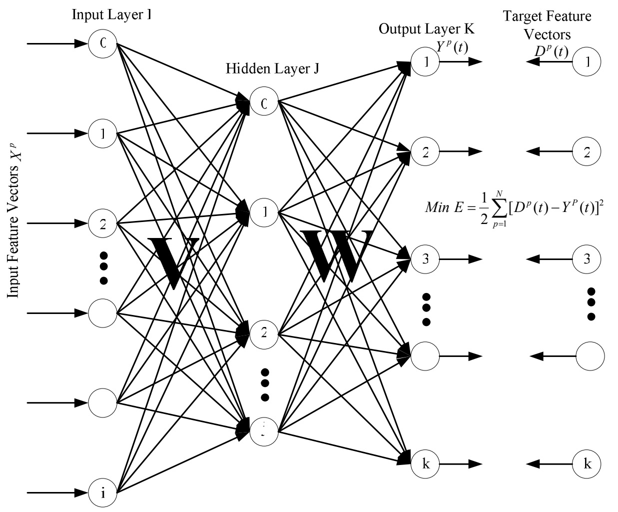


Fig. 3 A general BPN structure

A standard BPN is a fully connected network made up of input, hidden and output layers, as shown in Fig. 3. It is a typical three-layered BPN. Mathematically, the matrix input to hidden layer J from input layer I is formulated as:

$$NETJ = V \times X \tag{1}$$

$$a_j(t) = \frac{1}{1+\exp\left(-net_j(t)\right)},\ j = 1,2,\ldots,n_j;\ t = \text{time step.} \tag{2}$$

$$NETK = W \times A_{\text{hidden layer}} \tag{3}$$

$$a_k(t) = \frac{1}{1+\exp\left(-net_k(t)\right)},\ k = 1,2,\ldots,n_k;\ t = \text{time step.} \tag{4}$$

$$Y = A_{\text{output layer}} \tag{5}$$

where

$X \in R^{n_i \times N}$: input vector matrix of the BPN

$V \in R^{n_j \times n_i}$: weight matrix between the input layer I and the hidden layer J

$NETJ \in R^{n_j \times N}$: vector matrix input to sigmoid activation function of the hidden layer J

$A_{\text{hidden layer}} \in R^{n_j \times N}$: activated vector matrix output from the hidden layer J

$W \in R^{n_k \times n_j}$: weight matrix between the hidden layer J and the output layer K

$NETK \in R^{n_k \times N}$: vector matrix input to sigmoid activation function of the output layer K

$A_{\text{output layer}} \in R^{n_k \times N}$: activated vector matrix output from the output layer K

$Y = A_{\text{output layer}} \in R^{n_k \times N}$: output vector matrix of the BPN

N : the number of input vectors, p = 1, 2, .., N.

n_I , n_J , n_K : the number of neurons of input, hidden and output layers, respectively;

$a_j(t)$: sigmoid activated output of the jth hidden neuron,

$a_k(t)$: sigmoid activated output of the k^{th} hidden neuron,

$y_k(t)$: output of the k^{th} output neuron,

$y_k(t) = a_k(t)$, t = time step.

In detail, the matrices X, NET, A and Y are the respective vector matrixes as follows:

$$X^p(t) = \left[x_1^p(t), x_2^p(t), \ldots, x_{n_i}^p(t)\right]^T \tag{6}$$

$$X(t) = \left[X^1(t), X^2(t), \ldots, X^N(t)\right] \tag{7}$$

$$NETJ(t) = \left[net_1(t), net_2(t), \ldots, net_{n_j}(t)\right]^T \tag{8}$$

$$A_j(t) = \left[a_1(t), a_2(t), \ldots, a_{n_j}(t)\right]^T \tag{9}$$

$$NETK(t) = \left[net_1(t), net_2(t), \ldots, net_{n_k}(t)\right]^T \tag{10}$$

$$\begin{aligned} Y(t) &= \left[y_1(t), y_2(t), \ldots, y_{n_k}(t)\right]^T \\ &= A_k(t) = \left[a_1(t), a_2(t), \ldots, a_{n_k}(t)\right]^T \end{aligned} \tag{11}$$

where

$x_i^p(t)$: the i^{th} feature of the vector p input to the input layer,

$net_j(t)$: the j^{th} activation neuron of the hidden layer at time t,

$net_k(t)$: the k^{th} activation neuron of the hidden layer at time t.

The BPN training rule is to minimize the energy function $E(V,W)$ defined by the difference between the output Y^p and its target D^p as follows:

$$\min_{V,W}\left\{E(V,W) = \frac{1}{2}\sum_{p=1}^{N}\left[D^p(t) - Y^p(t)\right]^T\left[D^p(t) - Y^p(t)\right]\right\} \tag{12}$$

subject to (1) - (5),

where $D^p(t) = \left[d_1^p(t), \;\ldots, d_{n_k}^p(t)\right]^T$ and $Y^p(t) = \left[y_1^p(t), \ldots, y_{n_k}^p(t)\right]^T$

The training set of the BPN contains the pairs in the form of (x_i^p, d_k^p), where X^p and D^p can have the same or different dimensional number of the vector space. Traditionally, an optimal gradient steepest descent algorithm has been used to minimize the energy function $E(V,W)$, i.e.,

$$\frac{\partial E(V,W)}{\partial v_{ji}(t)} = \frac{\partial E(V,W)}{\partial a_j(t)} \frac{\partial a_j(t)}{\partial net_j(t)} \frac{\partial net_j(t)}{\partial v_{ji}(t)} \tag{13}$$

$$\frac{\partial E(V,W)}{\partial w_{kj}(t)} = \frac{\partial E(V,W)}{\partial d_k^p(t)} \frac{\partial d_k^p(t)}{\partial net_k(t)} \frac{\partial net_k(t)}{\partial w_{kj}(t)} \tag{14}$$

For obtaining optimal weights V and W, the following adapted weight rules are used.

$$v_{ji}(t+1) = v_{ji}(t) - \eta \frac{\partial E(V,W)}{\partial v_{ji}(t)} \tag{15}$$

$$w_{kj}(t+1) = w_{kj}(t) - \eta \frac{\partial E(V,W)}{\partial w_{kj}(t)} \tag{16}$$

where η is the learning rate, a term that has an important effect on the convergence time. If η is too small, the learning time may be too long to get a minimal solution of the energy function (12); a large η , on the other hand, may cause to convergence oscillations that may lead the energy minimization process to reach at a sub-optimal or infeasible point. The range of an adequate η value, generally, is set between 0.2 and 0.3, or the value can be adjusted according to the convergence progress situation during the learning process. Another additional factor called monument rate μ, as in (22) and (23) latter, is set a smaller value than η , normally from 0.05 to 0.15, and functions to "perturb" the current arriving point for "jumping" to another better solution point in the learning process.

Although the iterative gradients derive the learning vectors to form the memorization of ANN, this memorization could not be globalized in the weight space since the character of the gradient convergence itself; nevertheless, a global memorization perhaps also generalizes an unsatisfied solution due to the fact that the learned (global or local) weights are not the true solution to an original problem but are a kind of metaphor used to get the approximated solution to the problem. A global metaphor may not always approximate a global solution. Thus, how to find an appropriate metaphor for the best solution acquisition still is an opened question in the ANN research field. Establishing the relation between the metaphor and the current generalizing solutions during the learning process may be the one of directions to find the metaphor used for generalizing the best approximated solution [17].

The determination of the initial weights is another investigating problem in the ANN learning process. Different initialized weights perhaps cause to different solutions. In general, however, random starting weights are initiated the gradient search to find the optimal point in the weight space; besides, a different number of hidden layers also determine a different weight space. One hidden layer with moderate neurons may be enough for generalizing an approximated solution to a general problem. On the other hand, ANN may also suffer from the drawbacks of the slow learning and the over fitting, if its network size is too large, particularly, when it is used for a large volume of high dimensional training data, such as the CPFR data that will be simulated in the later section. The drawbacks may be conquered by the popular cross-validation method, which will be used in the proposed autonomous system implementation and introduced in the Experiments Section latter.

Furthermore, there an activation function, i.e., (2) and (4), called sigmoid function, is hided in each of hidden and output neurons and functions to fire the matrix product values, i.e., *NETJ* and *NETK*, for determining the output values from the layers. Normally, an activation function should be a continuous differentiable function so as to deal with the continuous weight values. In what follows, the standard BPN learning steps are written.

Step 1. (a) Set the initial values of all weights to small real numbers.

(b) Input the values of learning rate η, momentum rate μ, and error tolerance ε.

Step 2. Present learning matrix X to the input layer and specify the desired target output matrix D. The paired vector (X, D) is used to describe the learning examples.

Step 3. Calculate

$$net_j = \sum_{i=0}^{n_i} v_{ji} x_i^p, \quad a_j = 1/[1+\exp(-net_j)], \tag{17}$$

$$j = 0, 1, 2, \ldots, n_j; p = 1, 2, \ldots, \mathrm{N},$$

and

$$net_k = \sum_{j=0}^{n_j} w_{kj} a_j, \quad a_k = 1/[1+\exp(-net_k)], \tag{18}$$

$$k = 1, 2, \ldots, n_k.$$

The calculation involves the inputs from the bias neurons indicated by the subscript 0. The activation value of a bias neuron is 1.

Step 4. Set $y_k = a_k$

Calculate $E(V,W) = \sum_{k=1}^{n_k} (d_k - y_k) y_k (1 - y_k)$ (19)

(a) If $E(V,W)$ is less than ε, then stop;

(b) otherwise, continue the learning, i.e., calculate

$$E_{\text{hidden layer}} = \sum_{j=0}^{n_j} a_j (1 - a_j) \sum_{k=1}^{n_k} w_{jk} E(V,W) \tag{20}$$

Step 5. Calculate the weight adjustments

$$\Delta wkj = \eta E_k a_j, \quad \Delta w_{ji} = \eta E_j x_i^p \tag{21}$$

Step 6. Repeat steps 2-5 for each learning pair (x_i^p, d_k^p).

Step 7. Calculate new weights

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t) + \mu \Delta w_{kj}(t-1) \tag{22}$$

and $w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) + \mu \Delta w_{ji}(t-1)$ (23)

where t is a time step.

Step 8. Repeat Step 7 until the network finish learning all (x_i^p, d_k^p).

Go back to Step 3..

IV. Why BPN for CPFR

During the past two decades, conventional forecasting techniques on account of the statistic and analytic capabilities have become increasingly sophisticate. The general development supposes that greater sophistication or combined forecasting causes to increase forecast accuracy; while this could be true, considerable research has pointed out that simpler is sometime better! A more complex statistic or analytic technique does not mean to always provide significantly better results, particularly where the situations involve the resource information constraints in terms of both experienced quantitative information and empirical qualitative expertise [14] [18] [19] [20].

Another argument on conventional statistic forecasting method is that it could analyze a statistic result about what predicted value is being viewed prior to building the application. Although the statistic method can be developed and run to learn how to predict a future value based on known features of each forecasting factor, there are several problems with it, which are why progress in the field of forecasting was so slow prior to the advent of applied ANNs. One of the problems is that the statistic distribution of data should be determined. Incorrect distribution may result to a wrong statistic forecasting solution. Another problem is that a conventional statistic forecasting is hard to deal with the situations of outliers, missed or incomplete data, and thus a particular analytic procedure is needed to treat them. This additional therapy must cause a more complex statistic solution procedure. Instead, the above problems of conventional statistic forecasting can be simply, readily and simultaneously solved by BPN owing to its own inherent capabilities of fault tolerance, semantic structure and generalization [15] [19] [20]. These three virtues are explained below.

Because of an abundance of input neurons, the input faults can be limitedly tolerated by BPN. If the input data is imprecise, missed or incomplete, BPN can also output an approximate or exact solution, without knowing the priori statistic distribution but depending on the level of the data fault. The higher fault level, the more incorrect solution is. An acceptable tolerance of the data fault is asked for the correct solution. "GIGO (garbage in garbage out)" could occur in the BPN processing. Semantic discrimination is another BPN ability of handling symbolic data. The semantics in terms of rules and facts can be mapped or made onto the structure of the network [19] [20]. After learning the semantic rules and facts, the trained BPN can produce the new rules and facts and infer the solution. Other symbolic data can be binary, integer or discrete data.

Generalization is the solution process based on the trained BPN. The trained BPN is able to interpolate or extrapolate its learned data to produce the solution to new input data. The generalization may be regarded as the inference contrivance in the symbolic condition. It also can produce an approximate solution with a plausible data fault. This approximated solution can also be thought of as an answer to the problem of recovering nonlinear characteristics of physical phenomena from the unseen or imprecise measurement data [15] [16].

The most significant born ability of BPN is to deal with binary, continuous, discrete, range number and symbolic data at the same time. This is the major reason why BPN has a more suitable acceptability than conventional statistic analysis methods. For example, BPN can be presented with data of continuous valued function and meanwhile be updated after each learning data. Few if any conventional statistic forecasting methods are able to meet both of the data presentation and updating. Also, the spatial and temporal predictions can be simultaneously performed by taking advantage of the presentation of distinct data types to BPN. Again, there no conventional statistic forecasting approach has this kind of the "versatile" data representational ability [20].

In the proposed autonomous CF system with learnability for distributed sales and SKU stock data, BPN is taken as the learning method for the system implementation not because of its capability of dealing with different data types, but simply because its generalized solution is generally better than conventional statistic forecasting solutions. Several empirical studies have pointed out that there are certain problem domains where a BPN provides superior predictive accuracy to commonly used statistic formulation methods [14] [18] [20].

Moreover, the interest of using a BPN as a modular contrivance respectively to forecast the sales and stocks at various trading partners is caused by the facts that a truly fast modeling tool reflecting physical situations that frequently occur in CF is required, and an online solution to the CF problem is often desired. In a practical CF situation, a prior knowledge of a possibly nonlinear is poor and no well-grounded assumptions about the collected statistic data can be formulated; therefore, a kind of generalization tool like BPN is required.

V. Literature Review

There are literally hundreds of papers addressing methods and performance for forecasting, estimation or prediction using ANNs; however, it is still few publications in discussing the architecture and implementation of the autonomous CF with ANNs in CPFR. Caridi, et al., [5] [21] separately presented the same two-agent system for the automation and optimization of the negotiation process in CPFR. One agent called Advanced Model was in charge of solving the SKU exception situations based on the trading partners' predetermination of stock threshold values; another agent named Learning Model functioned as the re-determination of the current SKU stock thresholds according to the past SKU sales records. Their papers focused only on the SKU exceptions handling occurred in the CF process, and the threshold was not autonomously or intelligently re-determined but was time-consumptively recalculated and reworked with all of updated historic SKU data.

Gaur [22] addressed a two-stage supply chain model consisted of an ARMA demand model for a retailer serves and a manufacturer's fulfillment of the retailer's orders. The ARMA structure of the retailer demand process determined how the value of sharing demand information in SDC was. Three different conditions were assumed to conduct whether the manufacturer was necessary for acquiring the SKU sales information from its retailers and when there is value to sharing demand information, inferring demand information, or treating the order process as an independent noninvertible ARMA time-series for the manufacturer. The classic statistic ARMA time series, however, is with complex mathematics and without non-autonomous mechanism for the implementation of a practical CPFR system.

Danese [7] classified and organized CPFR structures into seven sorts based on certain contingency factors like CPFR goals and developments, characteristics of SKUs, market segments and SDC structure. The findings of this paper, such as the influence of SDC's relational structure on the number of potential CPFR partners that were affected with the CPFR development stage, and others, were proclaimed to offer an original contribution to the discussions on CPFR from both academic and managerial perspectives.

Caro [3] considered using demand learning with Bayesian theory and dynamic programming (DP) technique with a finite horizon multi-armed bandit model with several plays per stage to investigate the dynamic behavior of the SKU assortments of fast-fashion retailers. How such retailers to improve their SKU assortment over time for the maximization of overall profits during a given selling season was the subjective of the paper. The profound statistic parameters and DP methods accounting for implementation delays, switching costs, and demand substitution effects caused the mathematic formulation of the dynamic product assortment to be complicated.

Pramatari [23] studied the impact that their proposed SKU stock replenishment and ordering practice has on shelf availability. The practice was performed by information sharing and daily collaboration based on the concept of safety stock level determination implemented on the WWW-based SDC platform so as to increase order accuracy and to improve shelf availability. The safety stock simplicity and internet platform implementation make the system work efficient, but the calculation is not with learnability of the updated SKU stock data and cost.

The idea of Aviv [24] is one of the motivations to develop the proposed autonomous CF system proposed in this paper. He introduced the role of CF in SDC and proposed an autoregressive demand model used to reflect the reality in SDC environments and meanwhile utilized a scorecard to capture inventory considerations, make production smoothing, and update adherence-to-plans for the improvement of the SDC environment. An adaptive production planning process for the manufacturer established an appropriate balance between the various metrics of CF concern. The experimental results showed that a successful CF system can bring better partnerships information, improve decision supportability, correct SDC process for trading partners and then the SDC agility can be improved and maintained.

Chandra [8] provides another motivation to conceive the proposed CF architecture. A good CF technology can reduce the bullwhip effect with variability of inventory replenishment orders and then increase SDC efficiency. He also proposed an

International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:2, No:11, 2008

autoregressive time series applied to obtain multiple step-ahead demand forecasts for the serially correlated external demand consideration. The simulation results showed that the autoregressive multi-step forecasting model leaded to higher inventory performance for the downstream SKUs measured by the average inventory size at the fixed service level, compared to other forecasting methods considered. The SDC performance could be much higher, if the inventory status generated from MRP (materials requirement planning) was combined with the multiple step forecasting technique.

VI. THE LOGIC OF BPN-BASED CF SYSTEM

An accurate collaborative sales and SKU inventory forecast is a key for implementing a successful CPFR system. This section presents the development of the proposed autonomous CF system. The proposed architecture is composed of modules in the form of BPN, and such a modular BPN can be operated as a CF platform run at a trading partner location connected to other locations. At present, the proposed CF system has been experimented and developed by MATLAB neural networks tool box. At this moment, although it is not a web developed one, conceivably and expectably the idea and logic of programming modules and their uniform interfaces providing a consistent and platform-independent baseline CF mechanism are still suitable for developing a CPFR system on WWW. In the future, the modular BPN allows for the flexible and convenient extension to new autonomous collaborative SDC tasks.

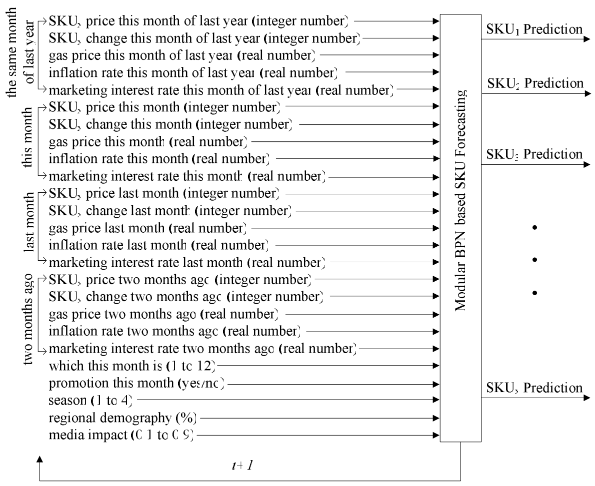


Fig. 4 CF features input to an autonomous modular BPN-based SKU predictor

Fig. 4 illustrates an example of an autonomous modular BPN used to forecast the future situation of several SKU sales transacted by a trading partner. The key factors affecting the SDC balance, such as historical sale trend, tracing SKU inventory position and variability, promotion and seasonality, goods transaction and certain economic factors can be taken into account since the modular BPN can deal with the mixed data with different types. The conventional statistic CPFR process, however, is difficult to answer all SDC factors with diverse measure units and assortment of all SKUs. The factor values of the factors vary from partner to partner.

Fig. 5 shows architecture of the proposed autonomous CF system, which introduces a modular, hierarchical framework consisted of a local level at the bottom and a global level at the top. The similar framework can be found in [25] and [26]. The modularity of the system introduces parallelism naturally. Modularized BPN predictors can run at the same time, and their outcomes are aggregated to become the data input to the manufacturer's BPN prediction module for getting and sharing the sales information in the global SDC. Fig. 6 shows a numeric example of the aggregated data from local trading partners to their manufacturing supplier for its global SDC prediction. The data values were scaled between 0.1 and 0.9 for meeting the requirement of activation task during the BPN training and solution processing.

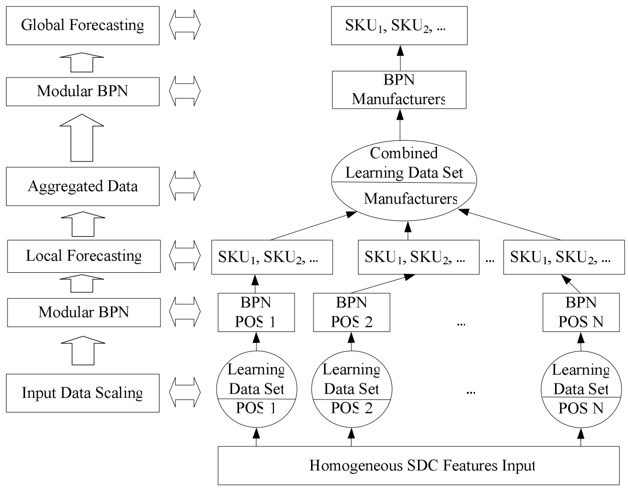


Fig. 5 Structure of the proposed autonomous modular BPN-based CF system

The proposed logic of performing CF tasks is according to the combination of local BPN predictors. The local BPN prediction module is trained by the sales and SKU data of the corresponding trading partner and is adaptively updated online, based on the previous and current data. The logic is hierarchical and allows complex learning problems to be solved by dividing the problems into a set of sub-problems. It can be modified or extended for matching the structure of the applied SDC problem.

VII. DATA GENERATION AND EXPERIMENTAL RESULTS

Based on the computer implementation and experiment experiences of authors' previous works [6] and [17], the proposed autonomous CF system was experimentally evaluated using simulated data sets in this section. The experiments of forecasting the SKU situations for a future target period are depicted as follows.

Given past experienced target SKU records as the target values of a modular BPN for a trading partner, they were paired with the corresponding input vectors that were formed by marketing factors shown in Fig. 4 to generate the learning set of the paired vectors. The marketing factors considered includes the following 25 forecast features: SKU price, SKU change, gas price, inflation rate, marketing interest rate and current month index for the same month of both last year and current year and for the previous two successive months of current month, in addition to promotion, season, regional demography and media impact; therefore, the number of input neurons in a module BPN is 25. The output neurons are 5 if 5 assorted SKU sale trends are considered for a trading partner (see Fig. 4). 5 time series data respective to these 5 SKUs were randomly generated based on the statistic normal distribution. Suppose 4 trading partners (3 local partners and one manufacturer) were in the simulated SDC, the total of random time series generated was 20 in which the 5 SKU series of data for the manufacturer were the combined ones from other three local partners (see Fig. 6). Besides, a window of successive 10 values for each series data was sampled with the scaled bounds of data magnitudes for the generation of diverse series that were needed to represent varying degrees of the forecast factors' situations; therefore, the simulated 20 SKU sales time series data were randomly tuned by a perturbation operation with three sigma variance ($\pm 3\sigma$). This perturbation made each input vector a number of operating levels as well as the corresponding target data can represent the full range of the simulated system behavior.

POS 1

X_{1}	SKU_{1}	SKU_{12}	SKU_{1}	Y_{1}
0 1	0 9	0 5	0 7	0 3
0 2	0 6	0 5	0 6	0 2
0 3	0 6	0 5	0 7	0 1
0 4	0 5	0 3	0 9	0 2
0 5	0 7	0 5	0 9	0 4

Wholesaler 2

X_{21}	SKU_{21}	SKU_{22}	Y_{21}
0 4	0 8	0 6	0 2
0 4	0 6	0 7	0 4
0 3	0 6	0 5	0 5
0 5	0 9	1 6	0 4
0 6	0 9	0 7	0 4

Manufacturer

X_{1}	SKU_{1}	SKU_{12}	SKU_{1}	Y_{1}	X_{21}	SKU_{21}	SKU_{22}	Y_{21}
0 1	0 9	0 5	0 7	0 3	0 4	0 8	0 6	0 2
0 2	0 6	0 5	0 6	0 2	0 4	0 6	0 7	0 4
0 3	0 6	0 5	0 7	0 1	0 3	0 6	0 5	0 5
0 4	0 5	0 3	0 9	0 2	0 5	0 9	1 6	0 4
0 5	0 7	0 5	0 9	0 4	0 6	0 9	0 7	0 4

Fig. 6 The aggregated data from trading partners to their supplier

After a number of the perturbation operations, a wide variety of the 20 sets of the simulated historical SKU sale trends with the 25 CF factors was generated and used as the training sets of the proposed modular BPN-based CF system. The generation of all of the 20 time series data were performed with MS Execl and sent to the proposed system made by MATLAB utilization tool box. An example of generating a SKU historical sale tread with the 25 CF factors for a modular BPN is described below.

Given a time series $x_1^p(t), x_2^p(t), \ldots, x_{n_i}^p(t)$ as an input vector $X^p(t)$ and the associated desired target $d^p(t)$, where p is an index of sampling window, and $n_i = 25$ = the number of CF factors, the value of target $d^p(t)$ was paired with its relative input vector $X^p(t)$ for the modular BPN training. A modular BPN was trained by off-line and on-line phases.

In the off-line phase, the weights W and V of a module BPN were adapted by minimizing the energy function (12). As $D(t)$ was known, simulated data vector were consecutively fed into the 25 input neurons and intermediate activated information was propagated forwards to the output layer. Comparing $Y(t)$ to $D(t)$, all connection weights were updated by (22) to (23) of the BPN training algorithm depicted in the former section.

In the on-line phase, the observed output data vector $Y(t)$ with respect to $X(t)$ were obtained after each one time lag, namely, the output data $y^p(t+1)$ was turned back to be the vector re-input to the module shown in Fig. 4. When the module got the fed-back output data vector, it produced a corresponding prediction, and the desired vector corresponding to this predicted value was the newest re-sending data vector. However, the weights were kept unchanged. The same procedure repeated for the following fed-back data vectors. For instance, the coming data vector $X(t)$ was the desired value of $Y(t+1)$. Since then, the weights of the module could be updated on-line by the same BPN training algorithm. The autonomy of the proposed CF system was performed by this on-line training and forecasting.

Yet, it should be known that the BPN learning is often faced with the problem of deciding which learning vectors should be memorized for the future generalization of solution. Learning too many data may result in mixed and confused memorization (improper weights), which is called over-fitting result, and long learning time, which is called over-learning process. These two abnormal conditions may cause oversensitivity to noise, which means that a genuine solution may be incorrectly generalized. The conventional method used to overcome these two abnormalities, and will be used in the following experiment, is called cross-validation [27]. This validation procedure is performed by perturbing (described in the former section) the data in learning set for the prevention of learning similar feature vectors, and by partitioning the learning set into several (S) subsets in which some of them are for learning and the remaining subsets for testing. This data division process is repeated ("crossed") S times by changing the test segments, and the performance is measured by the validation error between the BPN predicted and original data in test subsets.

To do the experiment of the proposed autonomous CF system, the system was trained by 1700 data vectors and tested by the remaindering 300 data vectors, which were extracted from the 4 simulated trading partners. Table I gives the partial training data simulated by the Execl software. It contains the time, order code and SKU type and its current quantity, and other factors are not shown because of the limited size of the paper. One data vector representing a SKU transaction situation was generated at the end of a random time. In this experiment, the training of the proposed autonomous CF system was performed by its four modular BPNs, simultaneously.

International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:2, No:11, 2008

The aim of the system is to predict the l^{th} sample ahead, $y^{p}(t+l)$, where l = 1 means for single-step ahead forecasting and l = 10 for 10-step ahead forecasting. The experimented scheme is based on the following five steps:

(1) Each of the 20 time series was decomposed into training and testing subsets for the cross-validation of the system.

(2) Each of 5 SKU records simulated for a trading partner was trained by the partner's modular BPN.

(3) The different training results of the "crossed subsets" of a SKU were validated respectively with its "test subsets" for determining the best trained BPN module; meanwhile, the autonomous CF system was established as the best 4 trained BPNs was had been determined completely.

(4) Using the completed autonomous CF system to perform the single-step ahead forecasting for each SKU of the 4 trading partners. The different forecasting outcomes of the BPN modules were regarded as the targeted SKU situations that were the next on-line training samples for the new simulated SKU sale trend prediction.

(5) Forecasting l-step ahead of one or more SKU stock levels, if necessary.

TABLE I
THE PARTIAL TRAINING DATA GENERATED BY THE EXCEL SOFTWARE

Time	Order Code	SKU	Quantity	Time	Order Code	SKU	Quantity
0	001	C	245	80.0	051	A	155
0.4	002	B	202	82.2	052	B	204
1.1	003	C	246	82.9	053	A	160
1.3	004	C	231	84.6	054	A	141
3.5	005	B	204	86.6	055	C	239
4.4	006	C	258	87.9	056	A	152
6.6	007	B	213	88.7	057	A	144
7.2	008	B	191	90.9	058	B	201
8.7	009	A	150	91.9	059	A	136
10.	010	B	214	93.9	060	A	150
12.4	011	C	240	95.0	061	C	225
13.7	012	A	139	96.0	062	C	256
16.0	013	C	229	97.0	063	B	207
17.7	014	C	229	100.0	064	A	151
20.1	015	B	214	103.0	065	C	259
22.1	016	C	262	104.6	066	C	264

Fig. 7 shows the four training curves of the same kind of SKU for the four simulated trading partners. It can be seen that all of the 4 convergence speeds were slow and the curves were "flatten" since the sizes of the training vectors were very large. Fig. 8 illustrates the prediction accuracy in terms of the differences between the original and predicted values of the two SKUs considered for the simulated manufacturer. It can be seen that the forecast accuracy of testing subsets of the two SKUs seemed not like to that of the training subsets which met the expected results. The reasons for the larger forecasting errors for the testing subsets may include (1) the improper number of hidden neurons and (2) the lake of better BPN learning algorithm. Nevertheless, these two BPN weaknesses will be overcome using an improvement BPN learning algorithm [17]. More comparisons and discussions among the simulated manufacturer and the three simulated local partners had been done but are not shown here because the limited length of the presenting paper. Detailed work about the experiments of proposed autonomous modular BPN-based CF system can be found in [28].

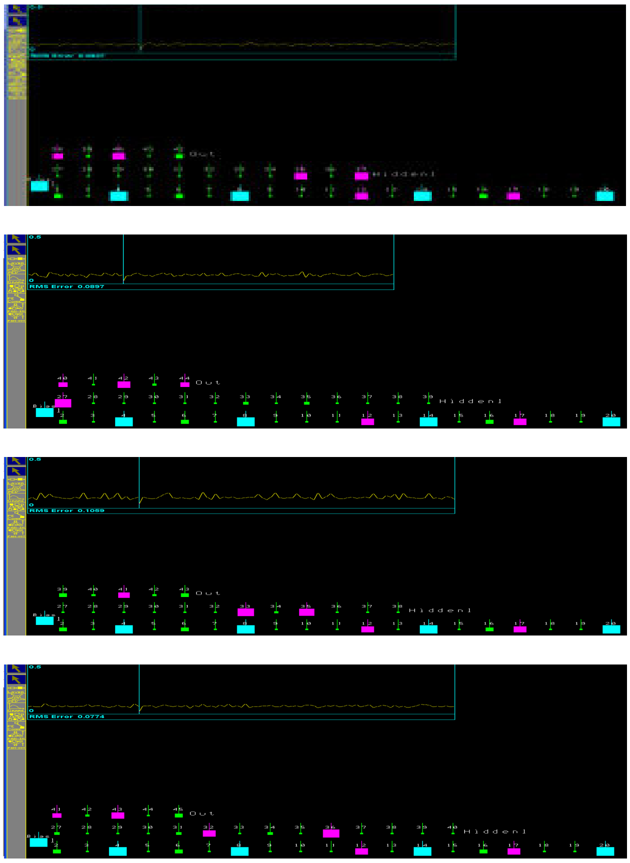
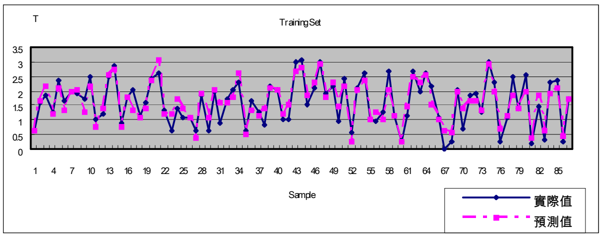


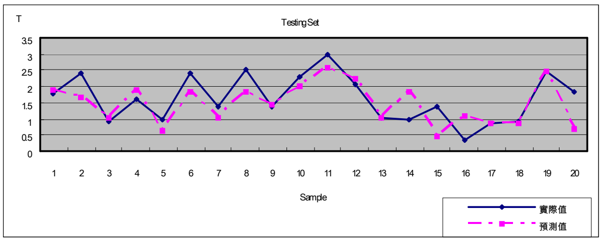
Fig. 7 The training curves of the simulated four trading partners

VIII. CONCLUSION

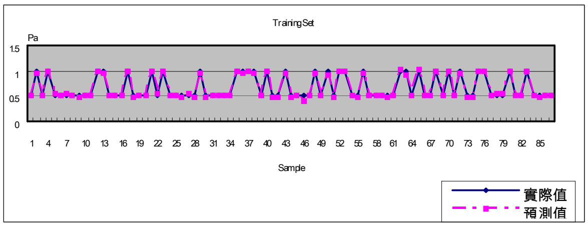
It is well-known that through CPFR, an enterprise is able to set up its SDC objective of product sales and productions so as to direct its global logistics system, particularly to guide how to reduce the product inventory level and to smooth the bullwhip effects. The proposed autonomous modular BPN-based CF system functions to achieve this objective by mans of assimilating marketing and production information. This paper positioned CF in the overall process of meeting SDC information requirements. It may not be as challenging as the integration of an intact CPFR system, but it truly is the primary foundation of recognizing CPFR. Besides, applied to an incomplete SKU input feature vector, the proposed autonomous CF system can perform the reconstruction of the missing feature(s) and the correction of outliers, which they are not presented in this paper but can be found in [29]. Also, applied to input feature vectors of lagged time series data, the proposed system can well perform the SKU forecasting for the upcoming management of the SKU inventory. The experimented 4-partner SDC can be expendable and applicable to more multistage SDC because the SKU order and sales process at each successive tier remains the nature of modularity process. The results are not startling but may provide a useful idea in implementing an autonomous CPFR system. Some further improvement forwards to an intact CPFR system is currently under study and future work will focus on applicability and refinement of the updated ANN methodology.



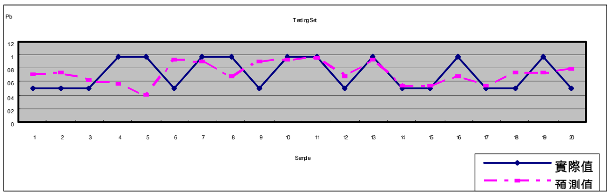
(a-1) Error of training set of SKU A



(a-2) Error of testing set of SKU A



(b-1) Error of training set of SKU B



(b-2) Error of testing set of SKU B

Fig. 8 Prediction accuracy of training and testing sets for two SKUs of the simulated manufacturer

REFERENCES

[1] VICS Association, Collaborative Planning Forecasting and Replenishment Voluntary Guidelines, 2002, Available: www.vics.org.
[2] Croson, R. and K. Donohue, "Behavioral Causes of the Bullwhip Effect and the Observed Value of Inventory Information," Management Science, vol.52, no.3, pp.323-336, 2006.
[3] Caro, F. and J. Gallien, "Dynamic Assortment with Demand Learning for Seasonal Consumer Goods," Management Science, vol.53, no.2, pp.276-292, 2007.
[4] Seifert, D., Collaborative Planning, Forecasting and Replenishment: How to Create a Supply Chain Advantage, AMACOM, American Management Association, 2003.
[5] Caridi, M., R. Cigolini and D. De Marco, "Linking autonomous agents to CPFR to improve SCM," Journal of Enterprise Information Management, vol.19, no.5, pp.465-482, 2006.
[6] Cheng, Yun-Hui, Hai-Wei Liao and Yun-Shiow Chen, "Implementation of a Back-Propagation Neural Network for Demand Forecasting in a Supply Chain - A Practical Case Study," in 2006 IEEE International Conference on Service Operations, Logistics and Informatics (SOLI '06), pp.1036-1041.
[7] Danese, P. "Designing CPFR collaborations: insights from seven case studies," International Journal of Operations and Production Management, vol.27, no.2, pp.81-204, 2007.
[8] Chandra, C. and J. Grabis, "Application of multi-steps forecasting for restraining the bullwhip effect and improving inventory performance under autoregressive demand," European Journal of Operational Research, vol.166, no.2, pp.337-350, 2005.
[9] Vereecke, A. and S. Muylle, "Performance improvement through supply chain collaboration in Europe," International Journal of Operations and Production Management, Vol.26, No.11, pp.1176-1198, 2006.
[10] Gurbuz, M. C., K. Moinzadeh and Y-P Zhou, "Coordinated Replenishment Strategies in Inventory/Distribution Systems," Management Science, vol.53, no.2, pp.293-307, 2007.
[11] Harrington, L. H., "9 steps to success with CPFR," Transportation and Distribution, pp.50-52, April, 2003.
[12] Chang, Ti-H, H-P Fu, W-I Lee, Y Lin and H-C Hsueh, "A study of an augmented CPFR model for the 3C retail industry," Supply Chain Management: An International Journal, vol.12, no.3, pp.200-209, 2007.
[13] Fliedner, G, "CPFR: an emerging supply chain tool," Industrial Management and Data Systems, vol.103, no.1/2, pp.14-21, 2003.
[14] Kotsialos, A., M. Papageorgiou, A. Poulimenos, "Long-term sales forecasting using holt-winters and neural network methods," Journal of Forecasting, vol.24, no. 5, pp.353-368, 2005.
[15] Bishop, Christopher M., Neural Networks for Pattern Recognition, Oxford University Press, 2004.
[16] Haykin, S., Neural Networks: A Comprehensive Foundation, 2nd Ed., Macmillan College Publishing, New York, 2001.
[17] Jeng-Bin Li and Yun-Kung Chung, "A Novel Back-propagation Neural Network Training Algorithm Designed by an Ant Colony Optimization," Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES, pp.1-5.
[18] Tawfiq, A. S. and E. A. Ibrahim, "Artificial neural networks as applied to long-term demand forecasting," Artificial Intelligence in Engineering, vol.13, no.2, pp.189-197, 1999.
[19] Sima, J., "Neural Expert System," Neural Networks, vol.8, no.2, pp.261-271, 1995.
[20] Fu, Li-Min, Neural Networks in Computer Intelligence, McGraw-Hill, 1995.
[21] Caridi, M., R. Cigolini and D. De Marco, "Improving supply-chain collaboration by linking intelligent agents to CPFR," International Journal of Production Research, vol.43, no.20, pp.4191-4218, 2005.
[22] Gaur, V., Giloni, A. and Seshadri, S., "Information sharing in a supply chain under ARMA demand", Management Science, vol.51, no.6, pp.961-969, 2005.
[23] Pramatari, K. P. T, "The impact of collaborative store ordering on shelf availability," Supply Chain Management: An International Journal, vol.13, no.1, pp.49-61, 2008.
[24] Aviv, Y., "On the benefits of collaborative forecasting partnerships between retailers and manufacturers," Management Science, vol.53, no.5, pp.777-794, 2007.
[25] Hrycej, T. Modular learning in Neural Networks: A Modularized Approach to Classification, Wiley, New York, 1992.
[26] Kehagias, A. and V. Petridis, "Predictive Modular Neural Networks for Time Series Classification," Neural Networks, vol.10, no.1, 1997, pp.31-49.
[27] Saito, K. and R. Nakano, "Discovery of relevant weights by minimizing cross-validation error," Proc. PAKDD 2000, LNAI 1805, pp. 372–375.
[28] Chi-Fang Huang, "Design of an integrated system of artificial neural networks and grey theory for collaborative prediction," MS thesis, Dept. of Industrial Engineering, Yuan Ze University, Taiwan, 2006.