

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Design of Demand Forecasting Expert System for Dynamic Supply Chains

Hanaa E. Sayed, Hossam A. Gabbar and Shigeji Miyazaki  
*Okayama University  
 Japan*

## 1. Introduction

When distributors and wholesalers seek help with issues relating to inventory management, they are usually concerned about an increasing level of out-of-stocks or over stocking. Out of stocks are leading to sales loss and customer service complaints. Over-stocks are resulting in slow inventory turnover and a buildup of dead inventory. In fact, out-of-stocks and over-stocks are actually a flip side of the same inventory management coin. Any effective initiative to resolve these issues must address core structural causes of these inventory management problems. Superior inventory management begins with timely, accurate, detailed demand forecasts.

Over last decade demand forecasting has played a prominent role in the corporations worldwide. Corporate executives have spent millions of dollars and invested thousands of man-hours trying to improve methods used & complicate it more. In each case little attention was paid to the integration between drivers, inputs and demand forecast (Harrison & Qizhong, 1993). In the face of all these advancements in hardware and software forecast error still remain high.

The inaccuracy in the forecast is due to previous researchers focused on statistical methods and their improvements only. There was no effort on the modeling of the problem and how to build an expert system to interact properly with the dynamic changes of the supply chain (Ajoy & Dobrivoje, 2005). The forecasting model is not treated as enterprise system has its specifications and constraints which are modeled and simulated.

In this research we propose a design of expert demand forecast system which is designed after deep understanding of demand cycle within dynamic supply chain and interaction between different parameters within the supply chain. It is utilizing Bayesian vector auto regression, restricted vector auto regression, and kernel fisher discriminant analysis (Scholkopf & Smola, 1998), (Scholkopf et al., 1999) with improved genetic algorithm to filter, analyze inputs and factors affecting demand along with demand history and then generate baseline and operational forecasts. This model proposes new mathematical and expert modeling methodology to generate forecasts. We used a practical case study from international FMCG (Fast Moving Consumer Goods) industry using over 1000 product types and results show that a significant forecast accuracy and other supply chain key performance indicators improvements over one year months rolling.

The proposed model is composed of the integration between statistical and intelligent methods with expert input to generate more accurate demand forecasts. The inputs to the

Source: Expert Systems, Book edited by: Petrică Vizureanu,  
 ISBN 978-953-307-032-2, pp. 238, January 2010, INTECH, Croatia, downloaded from SCIYO.COM

model are history demand time series, demand factors history series and the expert inputs. The process starts with calculating the effects of demand factors on the demand series, which is followed by eliminating the effects of these factors. The next step is to perform the forecasting of raw or baseline demand. Finally, perform the estimation of the best collection of factors and activities in future based on history and expert inputs or using the judgmental input to adjust the activities collection. The outcome of this process is final operational forecast.

## 2. The detailed framework

The framework of the proposed demand planning model consists of three sub models; the first sub model is called "Factors Classifying Model". Its inputs are history demand series and demand factors. The model outputs are cleaned history demand and Regular Factors Matrix (RFM) of possible factors on the demand series. The first model is responsible for the following consecutive steps: (1) evaluating real demand by eliminating the effects of the unavailability; (2) calculating the effects of demand factors on the cleaned sales using multiple regression models; (3) establishing the knowledge base which is updated after each run of the model; and (4) classifying input factors based on effect on demand.

The second sub model is called "Intelligent Time Series Model". Its inputs are the cleaned history demand series, RFM. It is responsible for calculating time series components (trend, seasonality, and cycles) of the real demand and calculating raw or baseline forecast which represents demand without the effects of demand factors. Baseline is calculated by combining selected statistical methods (Bovas & Johannes, 1983): simple exponential smoothing, winter's additive and winter's multiplicative methods, other techniques can be found in (Allan, 2005). The best combination is obtained by calculating the best set of weights for each method using genetic algorithm which minimizes MSE.

Finally, the last sub model is the "Intelligent Combining Model". Its inputs are the generated baseline forecast, RFM. And its output is the final forecast including forecast factors. The model compares factors using genetic algorithm, which minimizes the cost and increases the profit of forecast.

The final outcome of the model is the final demand forecast (Operational Forecast) and activities which maximize profit. Operational forecast is the summation of baseline forecast and activities forecast. The model can be further tuned using opinions of experienced people, which can change any of the activities followed by recalculation of the forecast model based on the new suggested parameters by experts. The proposed model is shown in Fig.1.

The following sections explain the construction of domain knowledge for demand factors which is used throughout the model. This is followed by the explanations of the first sub model (i.e. factors classifying model) in details. Then the second sub model (i.e. intelligent times series model) is explained along with the different statistical methods. And then the proposed genetic algorithm is explained using selected case study.

## 3. The domain knowledge & data structure

As explained in the introduction, the model inputs are divided into three categories: demand series, demand factors, and setup parameters. Demand series is the history demand time series which is considered as numbers exist in a time frame. Demand factors such as

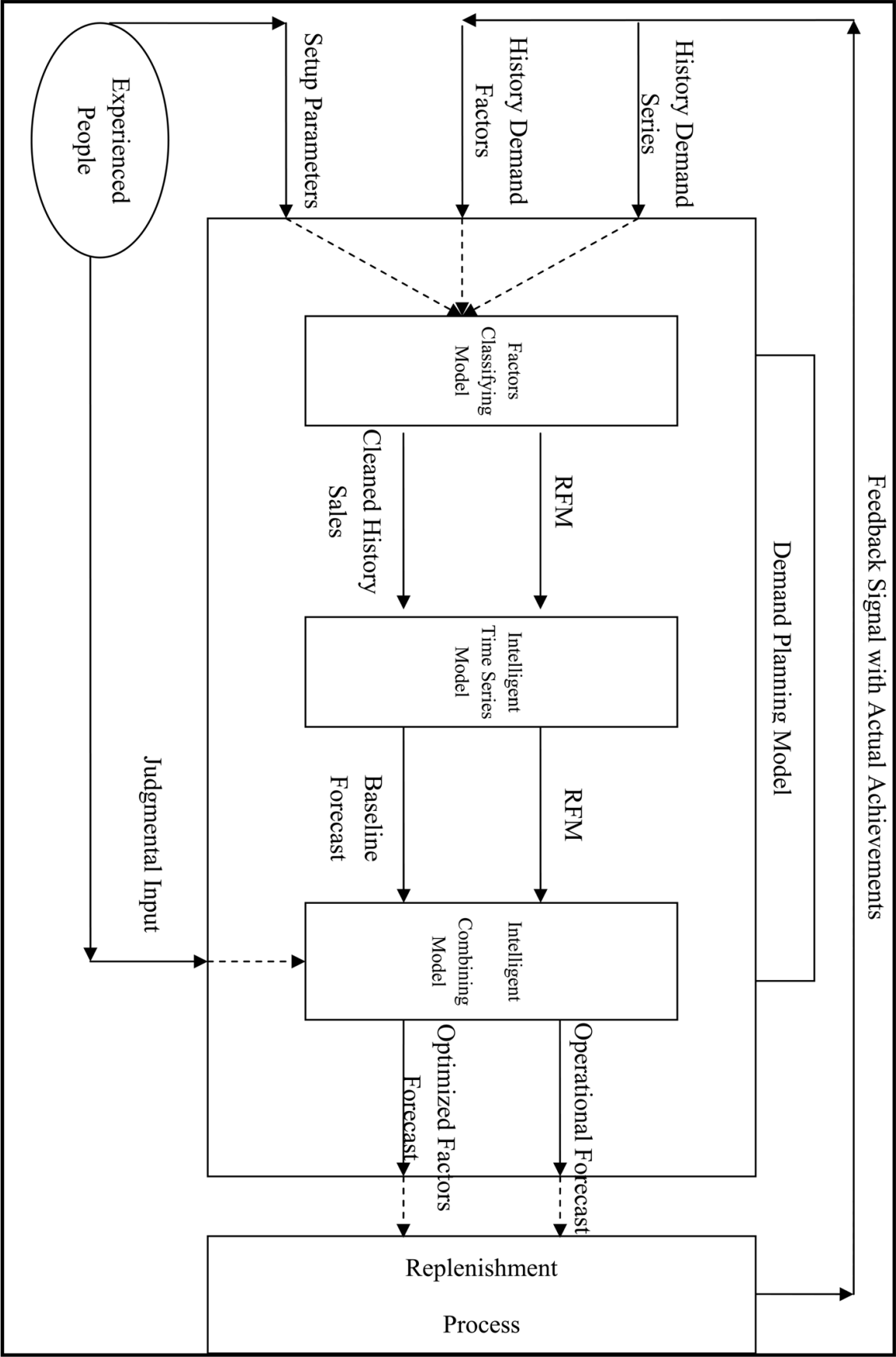


Fig. 1. Demand Forecast Model

activities or temperature. Set up parameters such as forecast horizon and history time buckets.

To ensure clear structuring of demand factors, it is needed to construct knowledge base, which is based on consumer products industry. It can be further tuned to fit another industry.

The following are the advantages behind the proposed knowledge base:

- 1. Explicitly define suitable structure of possible demand factors;
- 2. Enable the reuse of domain knowledge;
- 3. Clearly explain the different assumptions and intentions of model parameters;
- 4. Facilitate the analysis of domain knowledge.

3.2 “Demand Factors” classes

The classes of the demand factors domain knowledge are divided based on how they will be handled and analyzed in the model. Detailed explanations about the different demand factors with examples can be found in (Flores & Olson, 1992). Demand factors are divided into controllable and uncontrollable as shown in Fig.2.

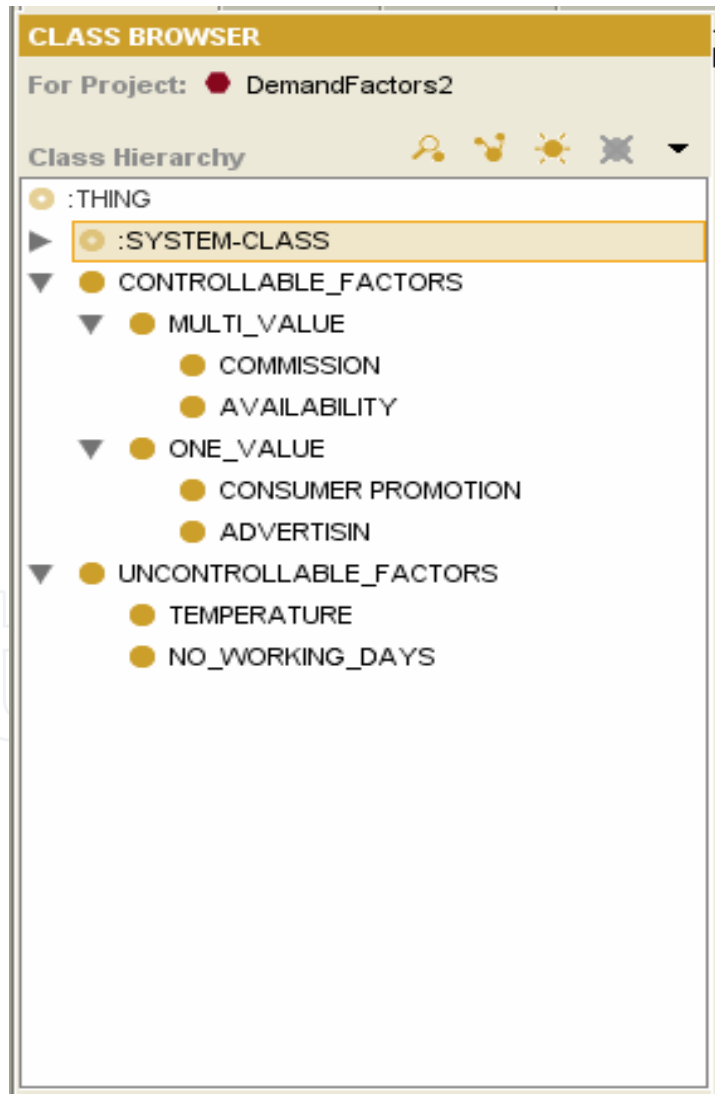


Fig. 2. Knowledge Base Classes for the Demand Factors

3.2 Controllable factors

Controllable factors are those that expert can control their timing and amount. During the forecast process, the proposed model calculates their effect percentages on demand series using multiple regression methods then forecast their time and amount which maximizes the profit.

Controllable factors are divided into multi value and one value. Multiple values mean its values are ranged from multiple numbers like percentage of discount, e.g. 2%, 4%, or any other value. One value means it occurs or not, it can be explained by ON / OFF, where ON is mapped to 1 and OFF is mapped to 0.

Multi value division is divided into:

1. Commission (CM): which indicates percentage of price discount given for certain period, and it can be maximum 100% and minimum 0%
2. Availability (AV): which indicated percentage of product availability, its maximum limit is 100% and its minimum limit is 0%.

One value division is divided into:

1. Consumer Promotion (CP): which are the promotions done to the consumers, like “chance to win” so its value is done or not done.
2. Advertising (AD): Like TV or radio advertising, or street shows, also its value is done or not done.

3.3 Uncontrollable factors

Uncontrollable factors means we can’t control their time nor amount. In order to predict them in future, the proposed model forecasts them using time series analysis method like linear regression.

Uncontrollable factors are divided into:

Temperature (T): This represents the temperature degree for each period, so it can be varied under long range. The temperature is represented qualitatively based on ranges of the degrees as shown in Table 1.

No\_Of\_Working\_Days (WD): Number of working days per period. It gives an indication of the strength of each period in terms of number of days they can produce and sell the product. WD is represented into two values small, and Normal as shown in Table 2.

Range	Quantitative
T < 10	VL (Very Low)
10 <= T < 20	L (Low)
20 <= T < 30	M (Moderate)
30 <= T < 35	H (High)
35 <= T	VH (Very High)

Table 1. Temperature Degree

Range	Quantitative
WD < 26	Small
WD > = 26	Normal

Table 2. Distribution of Number of Working Days

4. Factors classifying model

Factors Classifying model is the first sub model in the solution framework. The inputs to the sub model are the history demand series which is coming from sales history, the demand factors, products prices, and the setup parameters. The outcomes of that model are RFM and cleaned history demand series (Raw History Demand).

First module is using domain knowledge rule. Inputs are categorized and inserted into the predefined factors parameters which are used throughout the model. This division is useful where it facilitates the utilization of rules and constraints. It is easy to add additional factors in future where system can adapt itself automatically, without changing the model structure.

The second module, real demand calculation, is used to eliminate the effects of availability factor from regular series and promotion series to generate real history regular demand and real history promotion series.

Final module is activity analysis & cleaning. Its inputs are real history demand series and real history factors series. The outcome of the factors classifying model is the cleaned regular and promotion series, and the regular factors matrix and the promotion factors matrix. The term cleaned means that it represents the real demand without any effect of the demand factors. Multiple regression method is used to calculate the effects of demand factors on demand series. The framework of the cause & effect model is shown in Fig.3.

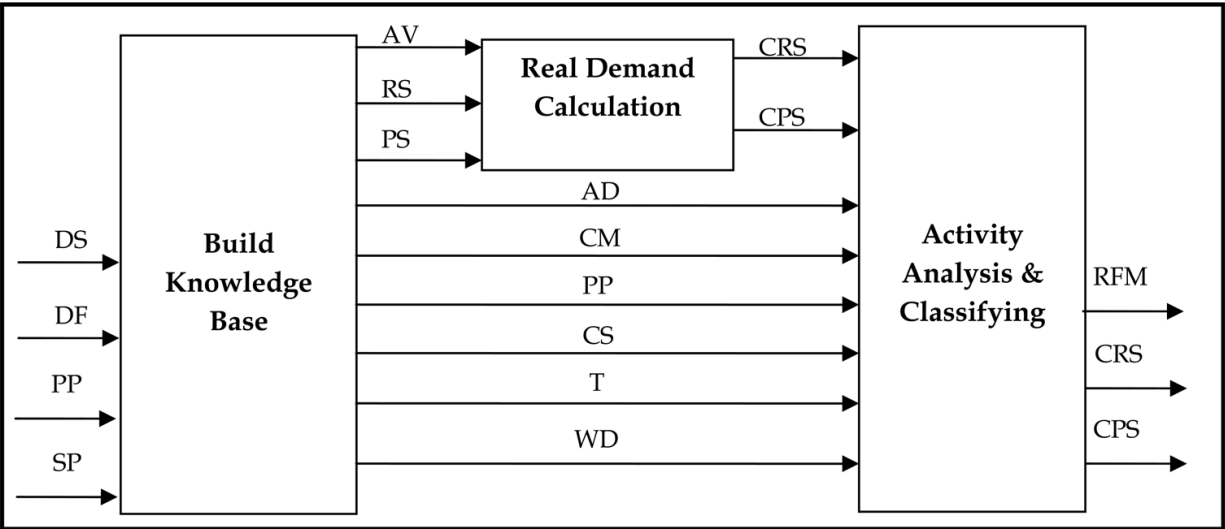


Fig. 3. Factors Classifying Model

4.1 Build knowledge base

Objectives:

This module is used to prepare and maintain the required data sets to perform the following tasks:

1. Calculate the data Series;
2. Calculate the activities costs;
3. Use the rules identified.

Inputs:

DS: represents the time series of the volumes of the demand and the time is the period buckets such as weeks or months.

DF: the history demand factors and activities and it consist of the following series:

P: series of the consumer promotions types and cost per month  
CM: represents series of the price reduction percentage per month  
AD: represents the series of the advertising types and cost per month  
T: the temperature degrees per month series  
WD: represents the series of number of working days per month  
AV: availability % parameter  
PP: represents the series of the product price history per month, this series is used to calculate the cost of the activities

Outputs:  
RS: the time series of the regular demand (actual sales without the consumer promotions)  
PS: the times series of the promotional demand.  
AD: time series of one value either ON or OFF whether there was ad or not.  
CM: the percentage of the price reduction.  
CS: the series of the cost for all activities

```
For All History i, i is the monthly periods {  
    IF (Pi > 0) Then { PSi = DSi  
        RSi = 0  
    } else { PSi = 0  
        RSi = DSi  
    } END IF  
    CS_CPi = PSi * P_PCi  
    CS_CMi = ( RSi + PSi ) * PPi * CMi  
    CS_ADi = AD_CCi  
    IF (Ti < 10) Then Ti = "VL"  
    Else IF (10<= Ti < 20) Then Ti = "L"  
    Else IF (20<= Ti < 30) Then Ti = "M"  
    Else IF (30<= Ti < 35) Then Ti = "H"  
    Else Ti = "VH"  
    END IF  
    IF (WDi < 26) Then WDi = "Small"  
        Else WDi = "Normal"  
    END IF }
```

Fig. 4. Knowledge Base Algorithm



T: temperature series is a divided into ranges in order to enable analyzing them and forecasting them.

WD: number of working days, which affects the number of visits to the consumer, and hence the sales.

Constraints:

$$RS \cap PS = \Phi$$

$$AD = 1 \text{ or } 0$$

#### 4.2 Real demand calculation

Objective:

1. Obtain the real demand by removing the effect of the availability parameter on the sales:

$$\text{Loss} = 0, \text{ if availability} = 100\%$$

$$= [RS * (100 - \text{availability})] / \text{availability}, \text{ if availability} < 100\%$$

2. Removing the effect of promotion volumes on regular sales (For all  $RS = 0$ ,  $RS = \text{average}$  (3 previous  $PS$ ))

Constraints:

$$\text{Availability} \leq 100\%$$

```

For All History {
    IF (AVi = 100) Then LSi = 0
    Else LSi = [(RSi * (100 - AVi)) / AVi] + [(PSi * (100 - AVi)) / AVi]
END IF

IF (RSi = 0) Then
    CPSi = PSi - LSi
Else
    CRSi = RSi - LSi
END IF
}

For All History {
    IF (CRSi = 0) Then
        CRSi = Average (CRSi-2, CRSi-1, CRSi+1, CRSi+2)
    END IF
}

```

Fig. 5. Real Demand Calculation Algorithm

4.3 Activity analysis & classifying

Objective:

- 1. Calculate the effect of the CP & AD & CM & CPS on the regular series using multiple regression statistical method for each uncontrollable pair (T, WD) to construct RFM;
- 2. Clean the regular series by removing the effect of the AD, CM and recalculate RS at PS > 0;
- 3. Classify the input factors based on effect using KFDA and improved genetic algorithm.

T	WD	CP (%)	CM (%)	AD (%)
VL	S	20	2	12
VL	N	23	3	14
L	S	27	5	18
L	N	35	7	19
M	S	40	8	25
M	N	46	10	27
H	S	52	11	29
H	N	60	12	31
VH	S	84	15	34
VH	N	95	17	36

Table 3. RFM Example

In the previous example, for raw one it indicates that when temperature is very low and #of working days is small then the effect of consumer promotion on the regular sales is 20%, the effect of 1% commission is 2%, and the effect of the advertising is 12%.

```
For Each Row in RFM {
    -- To calculate the CP%, AD%, CM% for the RFMi
    Multiple_Regression (RFMi)
}
-- Removing the effect of the AD & CM from the sales
FOR All History {
    ADi% = GET_AD_RFM (Ti, WDi)
    CMi% = GET_CM_RFM (Ti, WDi)
    CRSi = CRSi - ADi% * CRSi - CMi% * CRSi
}
FOR All History {
    IF (CPSi > 0) Then
        CRSi = Average( CRSi-2, CRSi-1, CRSi+1, CRSi+2)
    END IF
}
G_KFDA (RFM);
```

Fig. 6. Activity Analysis & Classifying Algorithm

5. Intelligent time series model

After cleaned regular sales are calculated, using previous module, the history sales will be analyzed using statistical and computational methods. At this model three types of methods are used: SES, additive winter's method and multiplicative winter's method (Armstrong, 1998). Then the genetic algorithm is used to calculate the best weights between the three methods to give the least error and generate baseline forecast using optimum weights.

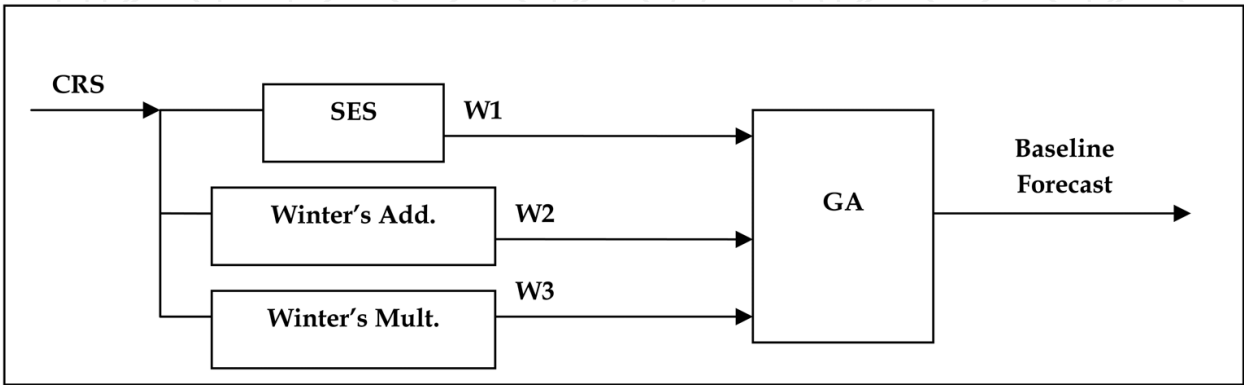


Fig. 7. Intelligent Time Series Model

5.1 Simple exponential smoothing

SES is used to forecast non seasonal Time series. The assumption is that the means moves slowly over time. Heuristically, in such a case it would be reasonable to give more weight to the most recent observations and less to the observations in the distant past.

- A = CRS
- Y = Forecast
- K = number of history periods
- $Y_n = \alpha A_n + (1 - \alpha) Y_{n-1}$
- $\alpha$  is the smoothing constant and it is usually chosen between 0.05 and 0.30
- Initial value of  $Y_0$ :
- Through repeated equation of the SES, it can be shown that:

$$Y_n = \alpha [A_n + (1 - \alpha) A_{n-1} + (1 - \alpha)^2 A_{n-2} + \dots + (1 - \alpha)^{n-1} A_1] + (1 - \alpha)^n Y_0$$

Thus the influence of  $Y_0$  on  $Y_n$  is negligible, provided that n is moderately large and  $(1 - \alpha)$  is smaller than 1. We take the simple average of the available history data  $(A_1, A_2, A_3, \dots, A_n)$  as the initial estimate of Y.

Choosing the smoothing constant  $\alpha$

The best value is between 0.05 and 0.30 so by simulating the result by calculating the MSE for each  $\alpha$ . and estimating  $\alpha$  which gives least MSE is the optimum one.

5.2 Winters' methods

Winters (1960) considers linear trend model with seasonal indicators. The seasonal and the trend components can be either additive or multiplicative. More details about the methods equations are introduced in chapter 2.

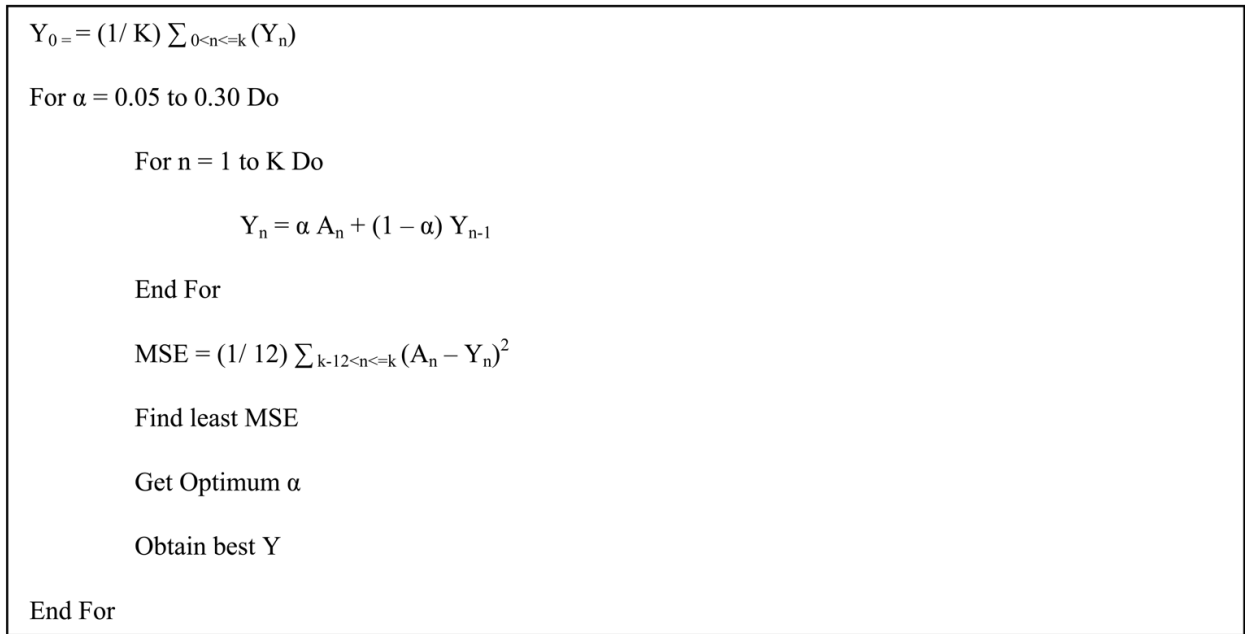


Fig. 8. SES Algorithm

To make initialization, it is needed one complete cycle of data, i.e.  $s$  values. Then set

$$L_s = \frac{1}{s}(Y_1 + Y_2 + \dots + Y_s)$$

To initialize trend, it is used  $s + k$  time periods

$$b_s = \frac{1}{k} \left[ \frac{Y_{s+1} - Y_1}{s} + \frac{Y_{s+2} - Y_2}{s} + \dots + \frac{Y - Y}{s} \right]$$

As long as the series is long enough, so it is used here  $k=s$  so that two completed cycles are used.

Initial seasonal indices can be taken as

$$S_k = \frac{Y_k}{L_s} \quad k = 1, 2, \dots, s$$

The parameters  $\gamma$ ,  $\beta$ , and  $\alpha$  must lie in the interval  $[0, 1]$ , and they are selected to optimize the MSE.

5.3 Proposed genetic algorithm

Objective:

The objective of GA is to calculate the optimum weights  $W = (W1, W2, W3)$  which minimize the MSE in the training set to get the best forecast.

- W1 is the weight of the SES method
- W2 is the weight of the winter's additive method
- W3 is the weight of the winter's multiplicative method

**GA elements:**

Fitness function = min (MSE)

$$\text{MSE}(Y) = (1/12) \sum_{k-12 < n \leq k} (A_n - Y_n)^2$$

$$Y_{\text{final}} = W_1 Y_1 + W_2 Y_2 + W_3 Y_3$$

**Constraints:**

$$0 \leq W < 1$$

$$\sum W'_s = 1$$

**Representation:**

Our variables are the weights of the different statistical methods.

First, we need to encode decision variables into binary strings. The length of the string depends on the required precision. In this case the domain of the variable  $w_j$  is  $[a_j, b_j]$  and the required precision is two places after the decimal point.

The required bits, denoted by  $m_j$ , for the variable is calculated as follows:

$$2^{m_j-1} < (b_j - a_j) * 10^2 \leq 2^{m_j}$$

The mapping from a binary string to a real number for variable  $w$  is completed as follows:

$$W_j = a_j + \text{decimal}(\text{substring}) * (b_j - a_j) / (2^{m_j} - 1)$$

Where the decimal (substring) represents the decimal value of substring<sub>j</sub> for decimal variable  $w_j$ .

To calculate the number of digits needed to represent the weight:

$$(1 - 0) * 100 = 100$$

$$2^{m_j-1} < 100 \leq 2^{m_j}$$

$$(M_j - 1) \log 2 < \log 100 \leq M_j \log 2$$

$$M_j - 1 < 6.6 \leq M_j$$

$$M_j < 7.6$$

$$M_j \geq 6.6$$

$$M_j = 7$$

Then number of digits needed for each weight is 7 digits. So the total length of the chromosome is  $3 * 7 = 21$  digits

Example of calculating the Weights:

$$W: 0011010 \rightarrow \text{its decimal number} = 26$$

$$W = 0 + 26 * 1 / (2^7 - 1) = 0.20$$

**GA Procedure:**

- Initialization of the population

The initial population is selected randomly. The population is consisting of 10 solutions as follows:

$V1 = [001001101101010110111]$   
 $V2 = [110010100011000001100]$   
 $V3 = [000010010011010101110]$   
 $V4 = [110010000100100001000]$   
 $V5 = [000000111110100000100]$   
 $V6 = [011111100110010100110]$   
 $V7 = [010001010011100001111]$   
 $V8 = [000000011101000001010]$   
 $V9 = [001010101010011000001]$   
 $V10 = [11110000000000000111]$

The corresponding decimals:

$V1 = [w1, w2, w3] = [0.15, 0.42, 0.43]$   
 $V2 = [w1, w2, w3] = [0.80, 0.10, 0.10]$   
 $V3 = [w1, w2, w3] = [0.03, 0.61, 0.36]$   
 $V4 = [w1, w2, w3] = [0.79, 0.14, 0.07]$   
 $V5 = [w1, w2, w3] = [0.01, 0.96, 0.03]$   
 $V6 = [w1, w2, w3] = [0.50, 0.20, 0.30]$   
 $V7 = [w1, w2, w3] = [0.27, 0.61, 0.12]$   
 $V8 = [w1, w2, w3] = [0.00, 0.91, 0.09]$   
 $V9 = [w1, w2, w3] = [0.17, 0.32, 0.51]$   
 $V10 = [w1, w2, w3] = [0.94, 0.00, 0.06]$

#### Evaluation:

The process of evaluation the fitness of a chromosome consists of the following three steps

#### Evaluation Procedure:

- Step 1.** Convert the chromosome's genotype to its phenotype. This means converting binary string into relative real values, which is happened above.
- Step 2.** Evaluate the objective function  $f(w^k)$ .
- Step 3.** Convert the value of objective function into fitness. For the minimization problem, the fitness is simple equal to the value of objective function eval  $(v_k) = f(w^k)$ ,  $k=1, 2, \dots, \text{pop\_size}$

$$\text{eval}(v_k) = \text{MSE}(Y)$$

$$Y = w1 * Y1 + w2 * Y2 + w3 * Y3$$

It is clear that chromosome  $V_7$  is the best result and that chromosome  $V_{10}$  is the weakest result.

#### Selection:

Its target to choose which solutions will remain in the new population and which solutions will be changed. By doing the following steps:

1. Get the least MSE from the original methods which is MSE
2. Arrange the solutions ascending based on their evaluation function
3. Compare the eval function for each solution by the  $\text{MSE}_{\text{best}}$ . The first one which is less than  $\text{MSE}_{\text{best}}$  will be taken to the new population. And the rest of the solutions will be changed.

Actual	SES	ADD	Multi
11.58	29.86	12.57	10.48
29.19	29.86	18.59	9.44
26.14	29.86	29.97	23.45
29.07	29.86	16.73	17.15
37.63	29.86	40.70	36.50
33.82	29.86	13.65	25.00
65.61	29.86	70.12	61.11
14.40	29.86	14.23	25.49
24.14	29.86	25.34	22.01
21.26	29.86	21.17	15.79
	24.00	67.04	81.78
	24.00	16.84	22.59
	24.00	19.10	15.19
	24.00	41.95	23.28
	24.00	39.55	22.64
	24.00	45.41	21.95
	24.00	48.45	26.02
	24.00	43.67	26.61
	24.00	63.10	42.92
	24.00	8.67	16.86
	24.00	15.02	19.22
	24.00	11.08	21.21

				weights			V1	
				W1	W2	W3	Forecast	Error
11.58	29.86	12.57	10.48	0.15	0.42	0.43	14.3	7.1
29.19	29.86	18.59	9.44	0.80	0.10	0.10	16.3	165.7
26.14	29.86	29.97	23.45	0.03	0.61	0.36	27.1	1.0
29.07	29.86	16.73	17.15	0.79	0.14	0.07	18.9	103.9
37.63	29.86	40.70	36.50	0.01	0.96	0.03	37.3	0.1
33.82	29.86	13.65	25.00	0.50	0.20	0.30	21.0	164.5
65.61	29.86	70.12	61.11	0.27	0.61	0.12	60.2	29.3
14.40	29.86	14.23	25.49	0.00	0.91	0.09	21.4	49.6
24.14	29.86	25.34	22.01	0.17	0.32	0.51	24.6	0.2
21.26	29.86	21.17	15.79	0.94	0.00	0.06	20.1	1.3
MSE	204	72	80				MSE	52.3

V2		V3		V4		V5		V6	
Forecast	Error	Forecast	Error	Forecast	Error	Forecast	Error	Forecast	Error
26.2	213.6	12.4	0.6	26.0	209.0	12.6	1.1	20.6	81.2
26.7	6.2	15.6	183.8	26.8	5.6	18.4	116.6	21.5	59.4
29.2	9.5	27.6	2.1	29.4	10.8	29.8	13.1	28.0	3.3
27.3	3.2	17.3	138.6	27.1	3.9	16.8	149.3	23.4	31.9
31.6	36.2	38.8	1.5	31.9	33.2	40.5	8.1	34.0	13.0
27.8	36.7	18.3	241.6	27.2	43.5	14.1	387.4	25.2	74.9
37.0	817.8	65.6	0.0	37.8	774.3	69.5	15.3	47.3	335.7
27.9	181.1	18.8	19.3	27.3	167.3	14.7	0.1	25.4	121.4
28.6	20.1	24.3	0.0	28.7	20.5	25.3	1.3	26.6	6.1
27.6	40.0	19.5	3.1	27.6	40.6	21.1	0.0	23.9	7.0
MSE	136.5		59.1		130.9		69.2		73.4

V7		V8		V9		V10	
Forecast	Error	Forecast	Error	Forecast	Error	Forecast	Error
17.0	28.9	12.4	0.7	14.4	7.7	28.8	296.3
20.5	75.0	17.8	129.8	15.8	180.0	28.7	0.2
29.2	9.2	29.4	10.7	26.6	0.2	29.5	11.3
20.3	77.0	16.8	151.3	19.1	99.0	29.2	0.0
37.3	0.1	40.3	7.3	36.8	0.8	30.2	54.8
19.3	209.8	14.6	368.0	22.1	136.3	29.6	17.8
58.3	53.8	69.3	13.9	58.9	45.7	31.6	1157.7
19.7	28.5	15.2	0.6	22.6	66.8	29.6	231.6
26.2	4.1	25.0	0.8	24.4	0.1	29.4	28.0
22.9	2.6	20.7	0.3	19.9	2.0	29.1	61.2
MSE	48.9		68.3		53.9		185.9

Implementing on our example:

- Eval (V1) = 52
- Eval (V2) = 136
- Eval (V3) = 59
- Eval (V4) = 131
- Eval (V5) = 69
- Eval (V6) = 73
- Eval (V7) = 49
- Eval (V8) = 68
- Eval (V9) = 54
- Eval (V10) = 186

Arranging the solutions based on the evaluation function:

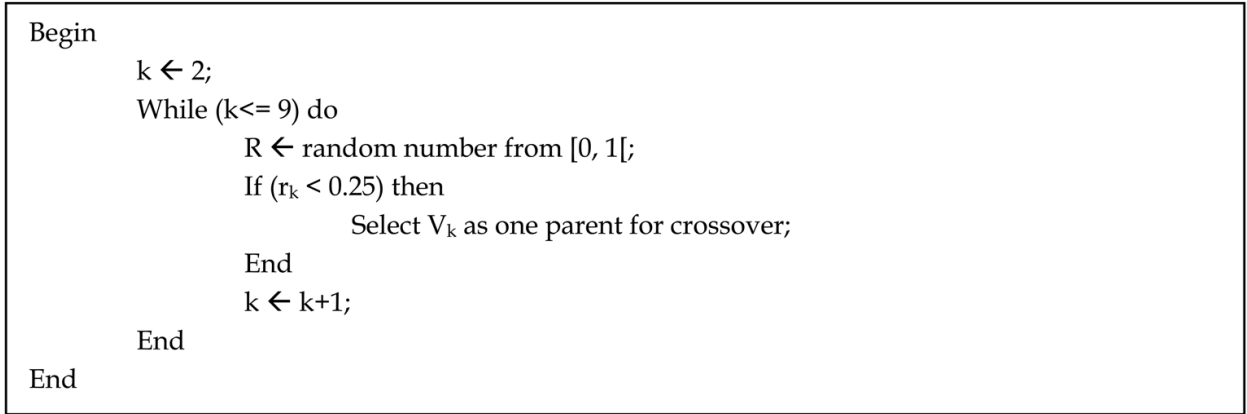
V7	49
V1	52
V9	54
V3	59
V8	68
V5	69
V6	73
V4	131
V2	136
V10	186

Table 5. Ordered Solution

The  $MSE_{best} = MSE(V7) = 72$   
Then V7 will stay in the new population and the rest of the solutions will be changed.

Crossover:

Crossover used here is one-cut-point method, which randomly selects one cut-point and exchanges the right parts of two parents to generate offspring.  
The probability of crossover is set as  $p_c = 0.25$ , so we expect that, on average, 25% of chromosomes undergo crossover. Crossover is performed in the following way:





Assume that the sequence of random numbers is:

0.266 0.288 0.295 0.163 0.567 0.0859 0.392 0.770 0.548 0.337

This means that the chromosome v4 and v6 were selected for crossover. Then we generate random number pos from the range [1, 20] (because 21 is the total length of a chromosome) as cutting point or in other words, the position of the crossover point. Assume that the generated number pos equals 3, the two chromosomes are cut after the first bit, and offspring are generated by exchange the right parts of them as follow:

V4 = [000000111110100000100]  
V6 = [010001010011100001111]  
      ||  
      V  
V4' = [010000111110100000100]  
V6' = [000001010011100001111]

Mutation:

Mutation alters one or more genes with a probability equal to the mutation rate. The probability of mutation is set as  $p_m = 0.01$ , so we expect that, on average, 1% of total bit of population would undergo mutation. There are  $m * pop\_size = 21 * 10 = 210$  bits in the whole population; we expect 2.1 mutations per generation. Every bit has an equal chance to be mutated. Thus we need to generate a sequence of random numbers  $r_k$  ( $k=1,2,...,210$ ) from the range [0,1]. Assume that the following genes will go through mutation:

Bit_pos	Chrom_num	bit_no	random_num
50	2	8	0.00987
62	2	20	0.00311
110	5	5	0.00128

After Mutation:

V<sub>2</sub> = [110010100011000001100]  
V<sub>2</sub> = [110010110011000001110]  
V<sub>5</sub> = [000000111110100000100]  
V<sub>5</sub> = [000010111110100000100]  
After the mutation & the crossover, the new generation is:  
V<sub>1</sub> = [001001101101010110111]  
V<sub>2</sub> = [110010110011000001110]  
V<sub>3</sub> = [000010010011010101110]  
V<sub>4</sub> = [010000111110100000100]  
V<sub>5</sub> = [000010111110100000100]  
V<sub>6</sub> = [000001010011100001111]  
V<sub>7</sub> = [010001010011100001111]  
V<sub>8</sub> = [000000011101000001010]

$V_9 = [001010101010011000001]$   
 $V_{10} = [111100000000000000111]$

We continue the iterations until the termination condition happened. The termination condition is that when doing 10 consecutive iterations without generating any new solution which is giving better evaluation function than the best previous one. Then we stop and the best solution with the optimum weights will be the last better evaluation function solution we obtained.

6. Intelligent combining model

The objective from this module is to calculate the optimum choices of the activities in the future periods which maximize the profit. So using this genetic algorithm module we can choose the best time we can do the activity and the best timing.  
The first step is to forecast the uncontrollable factors for the required periods. T and WD are forecasted using the intelligent time series model at Fig.10. Then to calculate what is the best combination of the controllable factors (AD, CP, and CM) timings using the genetic algorithm and the values are calculated using RFM from the values of forecasted T & WD corresponding to the time point.

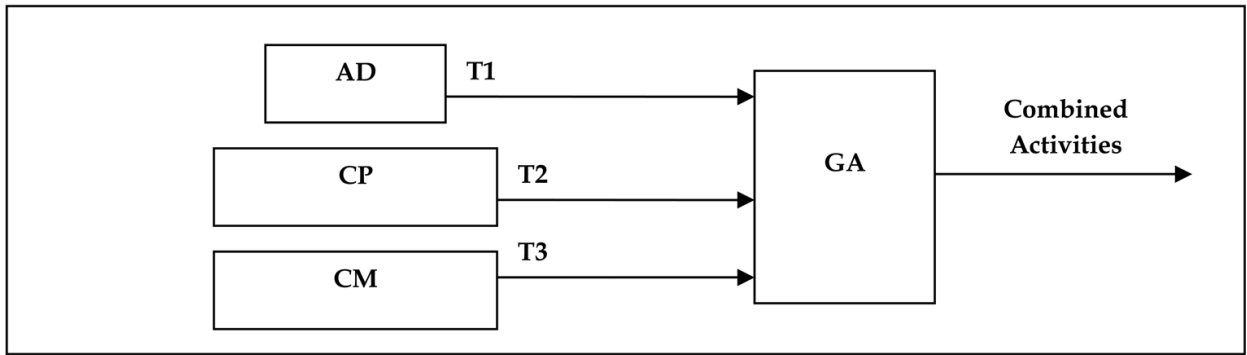


Fig. 10. Intelligent Combining Model

GA Elements:

Fitness function = Max (Profit)  
Profit = Sales - Cost

Constraints:

Cost <= Limited\_Cost

Assumptions:

# Periods = 12 periods

Representation:

- There are three variables:
- The timing of the CP for the coming 12 periods
  - The timing of the CM for the coming 12 periods
  - The timing of the AD for the coming 12 periods

To encode the above variables into binary digits so each variable will be consisting of 12 genes or bits representing the 12 periods. And if the bit is 0 that means there is no activity at this period, if the bit is 1 that means there is an activity at this period.

Total number of genes at each chromosome will equal  $12 * 3 = 36$ .

#### GA Procedure:

Initialization of the population:

The initial population is selected randomly. The population is consisting of 10 solutions.

Example of initial population:

```
V1 = [1011010111010100100101010010101010]
V2 = [001010101010101010001111110010101000]
V3 = [111010001100101001011111000100001010]
V4 = [010101010000111010101000011100101010]
V5 = [101010101010000001010101110010101011]
V6 = [010101010110101010010000111001010111]
V7 = [110101011011001010001110101000101011]
V8 = [111010010101001010010101001000010111]
V9 = [110110010100010101010100011100010110]
V10 = [010000101001010101010100010001100011]
```

After calculating their real values:

V1 → CP will happen in future in periods: 1, 3, 4, 6, 8, 9, 10, and 12

CM will happen in future in periods: 2, 5, 8, 10, and 12

AD will happen in future in periods: 3, 5, 7, 9, and 11

And so on

Evaluation:

Objective function = Max (Sales – cost)

$CP_i \rightarrow$  If at point  $i$  there is CP then it is 1, else it is 0

$AD_i \rightarrow$  If at point  $i$  there is CP then it is 1, else it is 0

$CM_i \rightarrow$  If at point  $i$  there is CP then it is 1, else it is 0

$CP\_Volume_i = CP_i * Y_i * GET\_CP\_RFM(T_i, WD_i)$

$AD\_Volume_i = AD_i * Y_i * GET\_AD\_RFM(T_i, WD_i)$

$CM\_Volume_i = CM_i * Y_i * GET\_CM\_RFM(T_i, WD_i)$

$Sales = \sum_{i=1to12} PP * [Y_i + (CP\_Volume_i + AD\_Volume_i + CM\_Volume_i)]$

$CP\_Cost_i = Average(P\_PC_i) * CP\_Volume_i$

$AD\_Cost_i = AD_i * Average(CS\_AD_i)$

$CM\_Cost_i = Average(PP_i) * CM\_Volume_i$

$Cost = \sum_{i=1to12} (CP\_Cost_i + AD\_Cost_i + CM\_Cost_i)$

For maximization problem, the fitness is simply equal to the value of the objective function

$eval(v_k) = f(x^k), k=1, 2, \dots, pop\_size.$

In this case we have three  $x$ 's:  $x_1$  represents the CP timing through the future 12 periods.  $x_2$  is represents the CM timing through the future 12 periods and  $x_3$  represents the AD timing through the future 12 periods.

CP\_Effect, AD\_Effect and CM\_Effect are calculated through getting the value of the % from the RFM table.

### Selection:

In most practices, a roulette wheel approach is adopted as the selection procedure; it belongs to the fitness-proportional selection and can select a new population with respect to the probability distribution based on fitness values. We constructed the roulette wheel as follows:

Calculate the fitness value  $\text{eval}(v_k)$  for each chromosome  $v$ :

$$\text{Eval}(v_k) = f(x^k) \quad k=1, 2, \dots, \text{pop\_size}$$

Calculate the total fitness for the population:

$$F = \sum_{k=1 \text{ to } \text{pop-size}} \text{eval}(v_k)$$

Calculate selection probability  $p$  for each chromosome  $v_k$ :

$$p_k = \text{eval}(v_k) / F, \quad k=1, 2, \dots, \text{pop\_size}$$

Calculate cumulative probability for each chromosome  $v_k$ :

$$q_k = \sum_{j=1 \text{ to } k} p_j, \quad k=1, 2, \dots, \text{pop\_size}$$

This selection process begins by spinning the roulette wheel  $\text{pop\_size}$  times; each time, a single chromosome is selected for a new population in the following way:

### Selection Procedure:

**Step 1.** Generate a random number  $r$  from the range  $[0,1]$ .

**Step 2.** If  $r \leq q_1$ , then select the first chromosome  $v_k$ ; otherwise, select the  $k$ th chromosome  $v_k$  ( $2 \leq k \leq \text{pop\_size}$ ) such that  $q_{k-1} < r \leq q_k$ .

Crossover & mutation will be implemented as explained in GA model. A detailed case study will be explained in the next chapter.

## 7. Results

In this session, there is a comparison between traditional statistical methods and the proposed model results. The comparison is based on the forecasting accuracy measure which is MSE. By running it on different types of products time series patterns and showing the difference between the two scenarios, it is shown that for all different types of times series the proposed model is giving better results than any traditional statistical method. Other examples are shown also in (Fred & Scott, 1992).

The model is run for 6 consequent months. And after each month the results are compared between the traditional statistical methods and the proposed model. And the following graphs are showing how improvements have done. A comparison between the models MSE is also shown. That proofs that the proposed model is improving forecasting accuracy for different types of times series. We will find too that each statistical method is giving better forecasting accuracy for some types of time series. That shows that complicating or using a standalone statistical method is not the best case to improve the forecasting accuracy.

No trend no seasonality

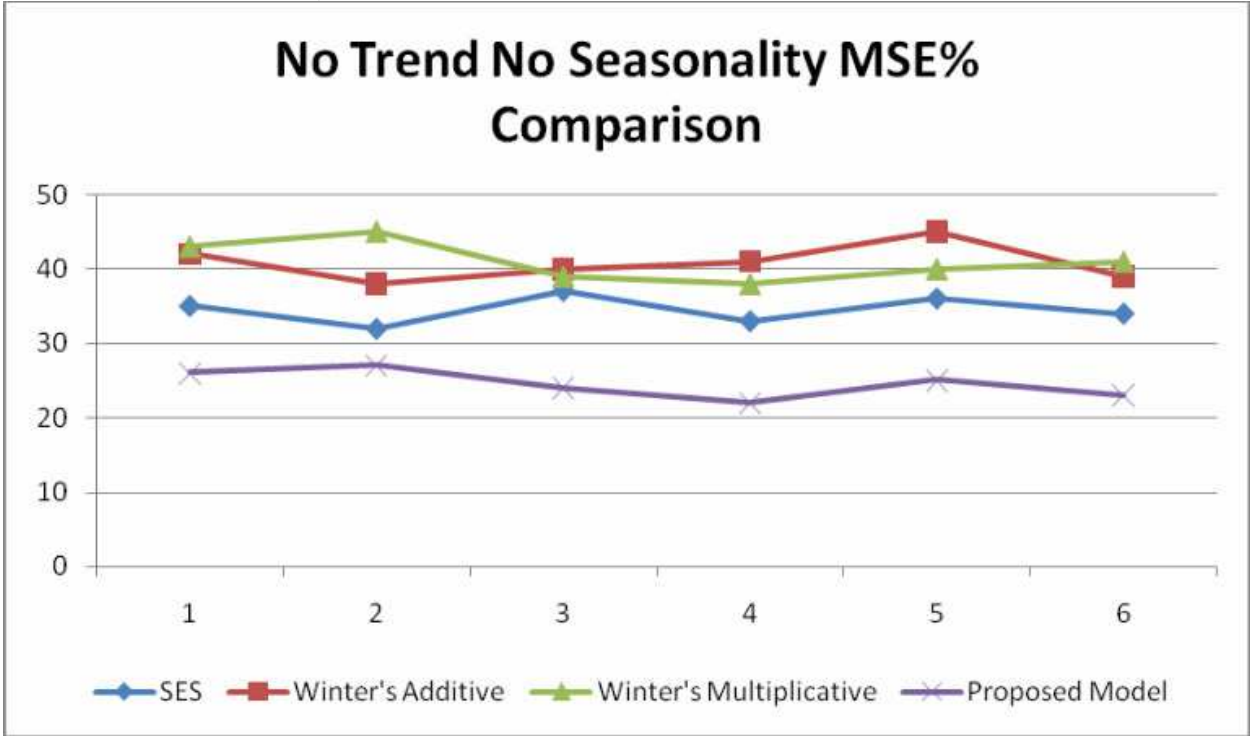


Fig. 11. Comparison of MSE% for No Trend No Seasonality Time Series

No trend linear seasonality

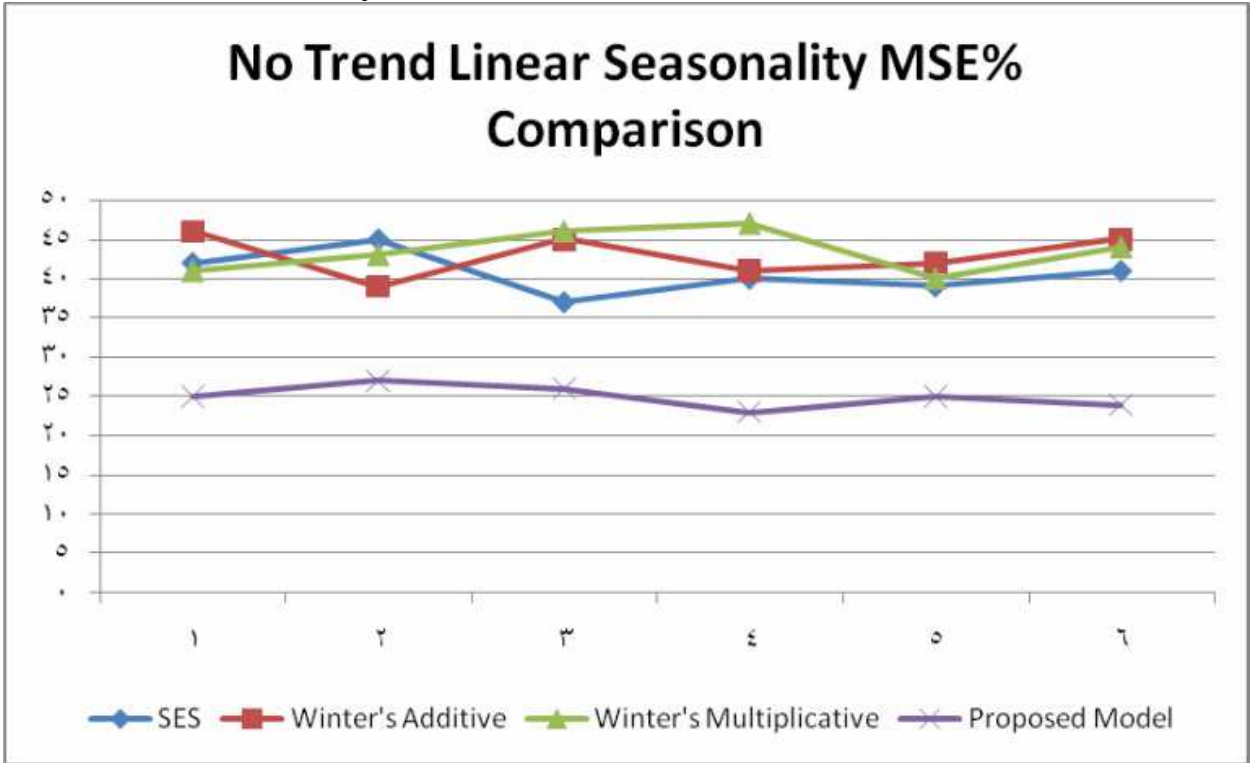


Fig. 12. Comparison of MSE% for No Trend Linear Seasonality Time Series

Linear trend linear seasonality

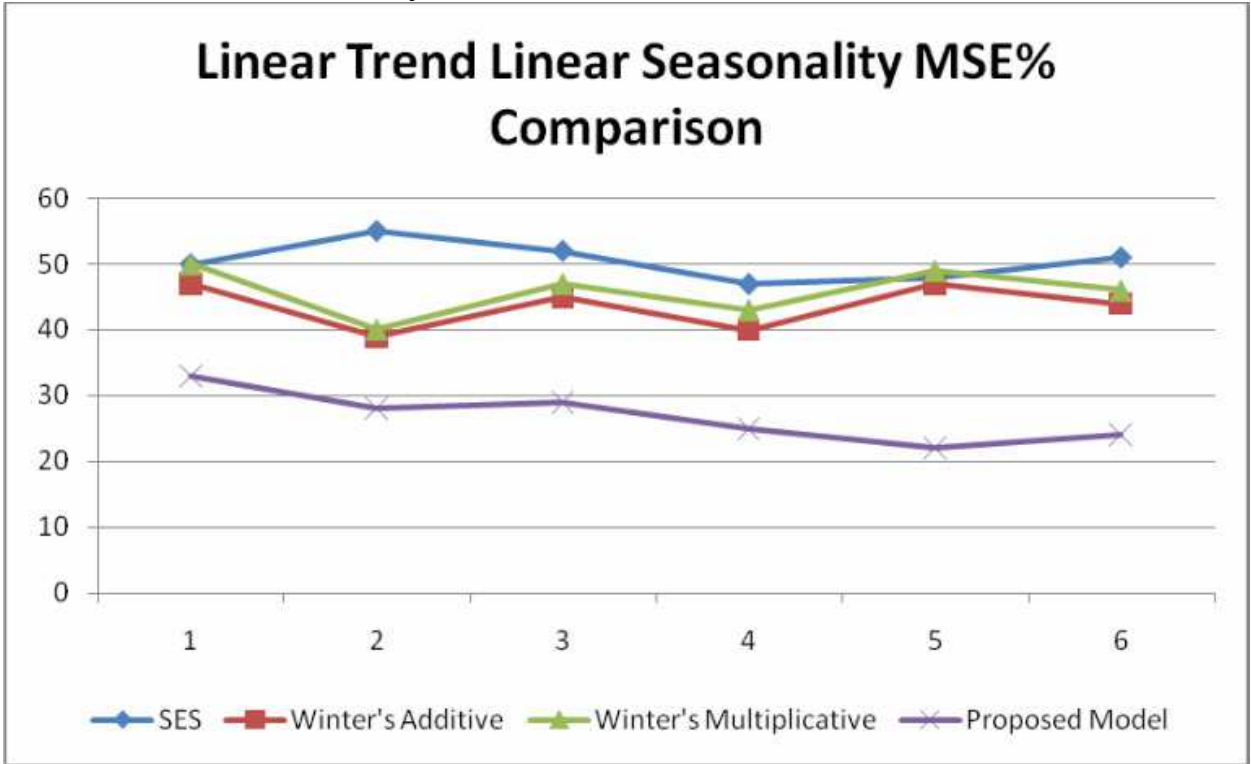


Fig. 13. Comparison of MSE% for Linear Trend Linear Seasonality Time Series

Linear trend exponential seasonality

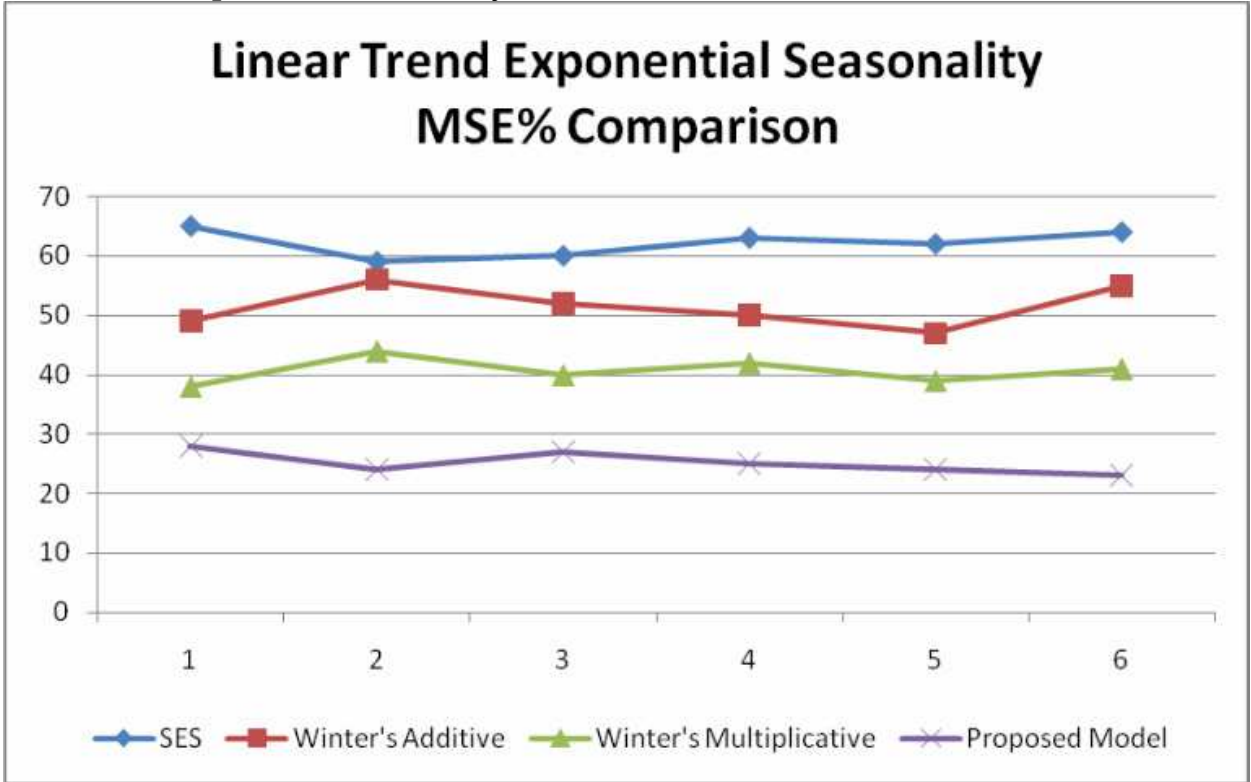


Fig. 14. Comparison of MSE% for Linear Trend Exponential Seasonality Time Series



Exponential trend linear seasonality

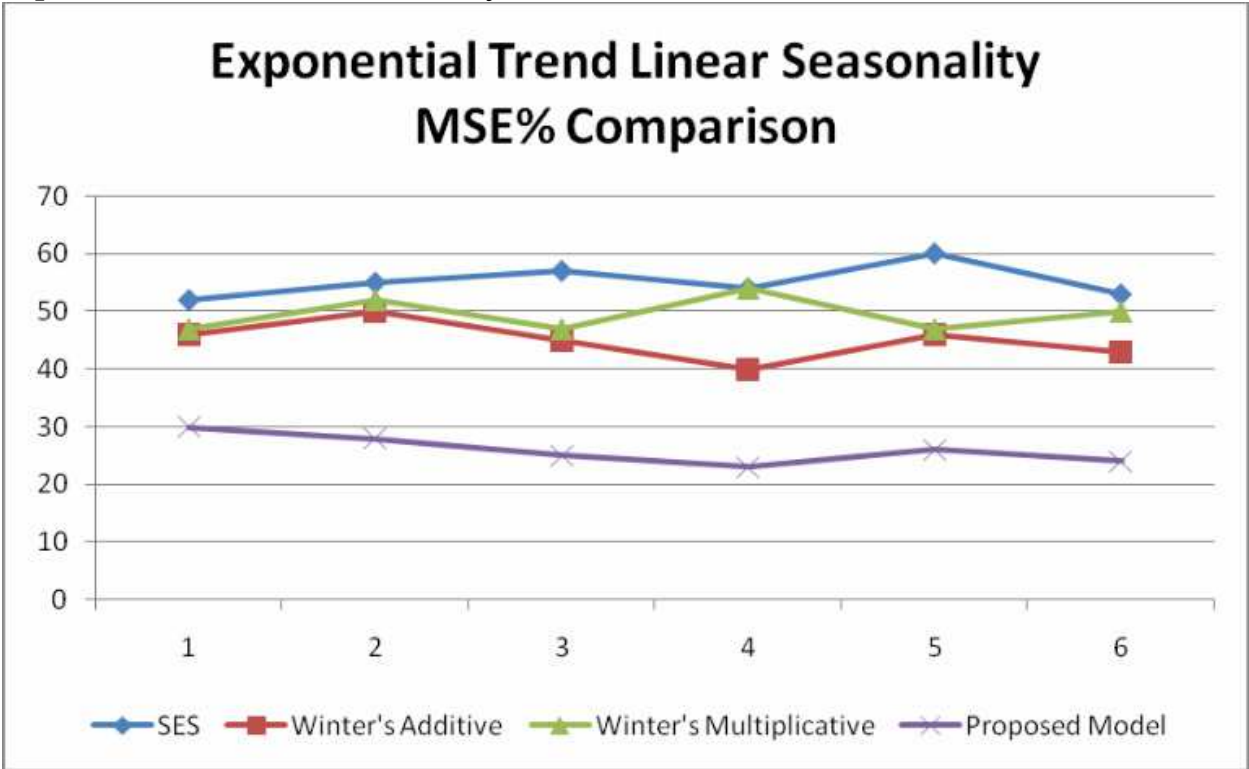


Fig. 15. Comparison of MSE% for Exponential Trend Linear Seasonality Time Series

Exponential trend exponential seasonality

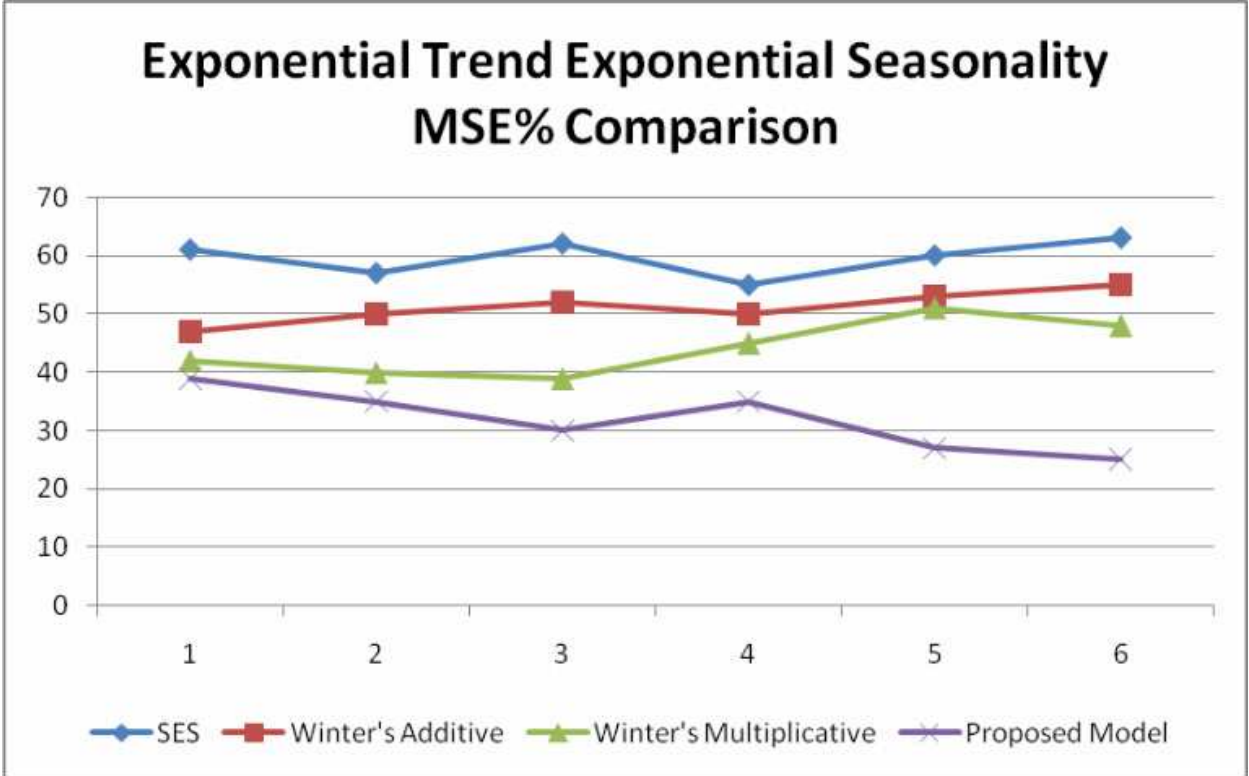


Fig. 16. Comparison of MSE% for Exponential Trend Exponential Seasonality Time Series

## 8. Conclusions & future work

Forecasting using single statistical method showed limitations in providing accurate forecasting. The use of a combined intelligent model is needed for providing accurate forecasting specially for complex environments which have different demand factors. As described in this chapter, the use of a new genetic algorithm is proposed to combine statistical methods and for combining activities with the baseline forecast, which suggests forecasts that are more accurate. First, the genetic algorithm searches for the best combination of weights between the methods, which minimizes MSE error. Then, genetic algorithm searches between the activities timing to choose the best timing, which increases the profit. Forecast methods are chosen so that they can cover different types of times series. Comparison of the obtained results shows better accuracy than that obtained using traditional methods. Other combinations of forecast methods can be included in the proposed solution for better forecast accuracy. Further improvement to the forecast model is obtainable by changing the crossover, mutation in the genetic algorithm, or by changing the initial population. As future work, it is recommended to try to add some neural techniques to the proposed model as it showed improving the forecasting capabilities.

## 9. References

- Ajoy K. Palit and Dobrivoje Popovic, 2005, *Computational Intelligence in Time Series Forecasting theory and engineering applications*, Springer.
- Allan Timmermann, 2005, *Forecast Combinations*, UCSD.
- B. Scholkopf, A.J. Smola, 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299-1319. [Online]. Available from: <http://www.mitpressjournals.org/doi/abs/10.1162/089976698300017467>. DOI: 10.1162/089976698300017467.
- B. Scholkopf, S. Mika, C.J.C. Burges, P. Knisch, K.R. Muller, G. Ratsch. 1999. Input space versus feature space in kernel based methods. *IEEE Transactions on Neural Networks*, 10:1000-1017. DOI: 10.1109/72.788641.
- Bovas Abraham, Johannes Ledolter, 1983, *Statistical Methods for Forecasting*, Wiley.
- Flores, B.E., Olson, B.L. & Wolfe, C., 1992, Judgmental adjustment of forecasts: a comparison of methods. *International Journal of Forecasting*, 7, 421-433.
- Fred Collopy, J. Scott Armstrong, 1992, *Rule-Based Forecasting: Development and Validation of an Expert Systems Approach to Combining Time Series Extrapolations*, *Management Science*, 38 (10), 1394-1414.
- H.C.Harrison, Gong Qizhong, 1993, "An intelligent business forecasting system", *ACM Annual Computer Science Conference*, Pages: 229 - 236.
- J. Scott Armstrong, 1998, "Integration of Statistical Methods and Judgment for Time Series Forecasting: Principles from Empirical Research", University of



Pennsylvania, Philadelphia, PA Fred Collopy Case Western Reserve University,  
Cleveland, OH.

IntechOpen

IntechOpen



## **Expert Systems**

Edited by Petrica Vizureanu

ISBN 978-953-307-032-2

Hard cover, 238 pages

**Publisher** InTech

**Published online** 01, January, 2010

**Published in print edition** January, 2010

Expert systems represent a branch of artificial intelligence aiming to take the experience of human specialists and transfer it to a computer system. The knowledge is stored in the computer, which by an execution system (inference engine) is reasoning and derives specific conclusions for the problem. The purpose of expert systems is to help and support user's reasoning but not by replacing human judgement. In fact, expert systems offer to the inexperienced user a solution when human experts are not available. This book has 18 chapters and explains that the expert systems are products of artificial intelligence, branch of computer science that seeks to develop intelligent programs. What is remarkable for expert systems is the applicability area and solving of different issues in many fields of architecture, archeology, commerce, trade, education, medicine to engineering systems, production of goods and control/diagnosis problems in many industrial branches.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hanaa E. Sayed, Hossam A. Gabbar and Shigeji Miyazaki (2010). Design of Demand Forecasting Expert System for Dynamic Supply Chains, Expert Systems, Petrica Vizureanu (Ed.), ISBN: 978-953-307-032-2, InTech, Available from: <http://www.intechopen.com/books/expert-systems/design-of-demand-forecasting-expert-system-for-dynamic-supply-chains>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen