International Journal of Computer Applications (0975 – 8887)
Volume 38– No.10, January 2012

A Particle Swarm Intelligence based Fuzzy Time Series Forecasting Model

Jilani T. A., Burney S.M.A., Amjad U.
Department of Computer Science,
University of Karachi,
Karachi-Pakistan.

Tanveer A. Siddiqui
Department of Mathematics
University of Karachi
Karachi-Pakistan.

ABSTRACT

In this paper, we have presented a new particle swarm optimization based multivariate fuzzy time series forecasting method. This method assumes five-factors with one main factor of interest. History of past three years is used for making new forecasts. This new method is applied in forecasting total number of car accidents in Belgium using four secondary factors. We also make comparison of our proposed method with existing methods of fuzzy time series forecasting. Experimentally, it is proved that our proposed method perform better than many existing fuzzy time series forecasting methods. The interest of this paper centers in applying swarm intelligence approaches in forecasting related problems. The dataset is taken from National Institute of Statistics, Belgium.

General Terms

Nature inspired computing, Time series forecasting

Keywords

Average forecasting error rate (AFER), Fuzziness of fuzzy sets, Fuzzy If-Then rule, particle swarm optimization (PSO).

1. INTRODUCTION

During last few decades, various approaches have been developed for time series forecasting. Among them ARMA models and Box-Jenkins model building approaches are highly famous. In recent years, many researchers used fuzzy time series to handle prediction problems. Song and Chissom [1] presented the concept of fuzzy time series based on the concepts of fuzzy set theory to forecast the historical enrollments of the University of Alabama. Huarng [2] presented the definition of two kinds of intervals in the universe of discourse to forecast the TAIFEX. Chen [3] presented a method for forecasting based on high-order fuzzy time series. Lee [4] presented a method for temperature prediction based on two-factor high-order fuzzy time series. Melike [5] proposed forecasting method using first order fuzzy time series for forecasting enrollments in University of Alabama. Lee [4] Presented handling of forecasting problems using two-factor high order fuzzy time series for TAIFEX and daily temperature in Taipei, Taiwan. In ([6], [7], [8], [9]), Jilani and Burney presented a number of new fuzzy metrics for multivariate fuzzy time series forecasting for car road casualties data, Taipei temperature data and enrollments data problem of University of Alabama.

The rest of this paper is organized as follows. In section 2, brief review of fuzzy time series is given. In section 3 and 4, brief review on nature inspired computing and particle swarm optimization is given. Our proposed PSO based FTS method is given in section 5. Experimental results are performed in section 6. The conclusions are discussed in section 7.

In this paper, we present a new modified method to predict total number of annual car road accidents based on the m-factors high-order fuzzy time series. This method provides a general framework for forecasting that can be increased by increasing the stochastic fuzzy dependence ([10], [11]). For simplicity of computation, we have used triangular membership function. The proposed method constructs m-factor high-order fuzzy logical relationships based on the historical data to increase the forecasting accuracy rate. Our proposed forecasting method for fuzzy time series gives better results as compared to Lee, Wang, Chen [4], Chen [3], and Jilani and Burney ([6], [7], [8], [9]). Table 1 shows yearly car accidents mortalities from 1974 – 2004 for Belgium.

2. FUZZY TIME SERIES

Most of the real life problems are born with uncertainty, therefore, each observation of a fuzzy time series is assumed to be a fuzzy variable along with associated membership function. Based on fuzzy relation, and fuzzy inference rules, efficient modeling and forecasting of fuzzy time series is possible, ([12], [13]). This field of fuzzy time series analysis is not very mature due to the time and space complexities, thus we can extend this concept for many antecedents and single consequent. For example, in designing two-factor kth-order fuzzy time series model with X be the primary and Y be second fact. We assume that there are k-antecedent $\left((X_1,Y_1),(X_2,Y_2),\ldots,(X_k,Y_k)\right)$ and one consequent X_{k+1}.

$$\textit{If } \left(X_1{=}x_1,Y_1{=}y_1\right),\left(X_2{=}x_2,Y_2{=}y_2\right),\ldots,\left(X_k{=}x_k,Y_k{=}y_k\right)\rightarrow\left(X_{k+1}{=}x_{k+1}\right) \tag{1}$$

In the similar way, we can define m-factor i=1,2,...,m and kth order fuzzy time series as

$$\begin{aligned}&\textbf{If } \left(X_{11}{=}x_{11},X_{12}{=}x_{12},\ldots,X_{1k}{=}x_{1k}\right),\\&\left(X_{21}{=}x_{21},X_{22}{=}x_{22},\ldots,X_{2k}{=}x_{2k}\right),\ldots,\\&\left(X_{m1}{=}x_{m1},X_{m2}{=}x_{m2},\ldots,X_{mk}{=}x_{mk}\right)\\&\textbf{then } \left(X_{m+1,k+1}{=}x_{m+1k+1}\right)\\&\textit{for } \; i=1,2,\ldots,m \;,\; j=1,2,\ldots,k\end{aligned} \tag{2}$$

Formally, the proposed method is given in section 5.

3. NATURE INSPIRED COMPUTING

Nature Inspired Computing (NIC) is a type of computing technique that gets ideas by understanding how nature deals in various situations to solve complex problems ([14]). Research on NIC has opened new branches such as evolutionary computation, artificial immune systems, swarm intelligence, and so on. The improvement in the computer hardware and massive increase in the computational power have made the emergence of nature inspired computing possible. That's why NIC techniques are implemented in many fields like biology, physics, engineering, economy, management, and so on. Many different types of models like swarms, colonies, or other natural metaphors have been developed which can help us to model extremely complex and dynamic systems.

Table 1. Yearly Car Accidents Mortalities and Victims from 1974 to 2004

Year	Killed (X)	Mortally wounded (Y1)	Died 30 days (Y2)	Severely wounded (Y3)	Light casualties (Y4)
2004	953	141	1,094	5,949	41,627
2003	1,035	101	1,136	6,898	42,445
2002	1,145	118	1,263	6,834	39,522
2001	1,288	90	1,378	7,319	38,747
2000	1,253	103	1,356	7,990	39,719
1999	1,173	126	1,299	8,461	41,841
1998	1,224	121	1,345	8,784	41,038
1997	1,150	105	1,255	9,229	39,594
1996	1,122	115	1,237	9,123	38,390
1995	1,228	109	1,337	10,267	39,140
1994	1,415	149	1,564	11,160	40,294
1993	1,346	171	1,517	11,680	41,736
1992	1,380	173	1,553	12,113	41,772
1991	1,471	209	1,680	12,965	43,578
1990	1,574	190	1,764	13,864	46,818
1989	1,488	312	1,800	14,515	46,667
1988	1,432	339	1,771	14,029	45,956
1987	1,390	380	1,770	13,809	44,090
1986	1,456	330	1,786	13,764	42,965
1985	1,308	352	1,660	13,287	39,879
1984	1,369	363	1,732	14,471	42,456
1983	1,479	412	1,891	14,864	42,023
1982	1,464	406	1,870	14,601	40,936
1981	1,564	454	2,018	15,091	41,915
1980	1,616	557	2,173	15,915	42,670
1979	1,572	544	2,116	15,750	42,346
1978	1,644	728	2,372	16,645	44,797
1977	1,597	701	2,298	15,830	44,995
1976	1,536	728	2,264	16,057	44,227
1975	1,460	701	2,161	15,792	42,423
1974	1,574	819	2,393	16,506	44,640

NIC helps us understand the underlying mechanism of a real-world complex system by formulating computing models and testing hypotheses through controlled experimentation ([15], [16]). The end product of such computing experiments is a deep understanding or a new discovery of the real working mechanism of the modeled system. NIC also enables us to embody autonomous (e.g., life-like) behavior in solving computing problems. With detailed knowledge of the underlying mechanism, abstracted autonomous behavior can be used as a model for a general-purpose problem-solving strategy or method. Self-organization is the essential process of its working mechanism. Through the local interactions, an NIC system self-aggregates and amplifies the outcome of entity behavior.

4. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality ([17]). PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search space according to simple mathematical formulae over the particles position and it is also guided toward the best known position in the search space, which are updated as better positions are found by other particles.

PSO could be explained well in an imagined scenario: a group of birds are flying in an area to look for food, and there's only one piece of food in this area. Each bird in the group doesn't know the exact location of the food, but they are aware of the distance between the food and themselves. In this way, the easiest way to find the food is to follow the one who is closest to the food. When it comes to the algorithm of PSO, it starts with the initialization of a population of random particles, each of which is associated with a position and a velocity. The velocities are adjusted according to the historical behavior of each particle and its neighbors while they fly through the search space. The positions are updated according the current position and the velocities at the next step ([18]). Therefore, the particles have a tendency to fly towards the better and better search area over the search process course. Initially, some particle is identified as the best particle in a neighborhood of particles, based on its fitness. All the particles are then accelerated in the direction of this particle, but also in direction of their own best solutions that they have discovered previously.

In other words we can say that the core concept of PSO lies in accelerating each particle toward its pbest which was achieved so far by that particle, and the gbest which is the best value obtained so far by any particle in the neighborhood of that particle, with a random weighted acceleration at each time step. A particle's velocity and position are updated as follows.

$$v_i^{(t+1)} = w^{(t)} \times v_i^{(t)} + c_1 \times r_1 \times (pbest - x_i^{(t)}) + c_2 \times r_2 \times (gbest - x_i^{(t)}) \tag{3}$$

$$X^{(t+1)} = X^{(t)} + v^{(t+1)} \tag{4}$$

Where X and V are position and velocity of particle respectively. w is inertia weight, c1 and c2 are positive constants, called acceleration coefficients which control the influence of pbest and gbest on the search process, P is the number of particles in the swarm, r1 and r2 are random values in range [0, 1].

The Basic Algorithm for PSO is as under:

1. **Create a 'population' of particles uniformly distributed over X.**
2. **Initialize every particle.**
3. **Calculate fitness value for each particle.**
4. **If current fitness value of a particle is better than the personal best (pbest) of particle set this as pbest.**
5. **Choose the particle with the best fitness value of all particle as the global best (gbest).**
6. **Update particles velocities according to equation 3.**

7. **Move particles to their new positions according to equation 4.**
8. **Go to step 2 until stopping criteria are satisfied.**

PSO has some drawbacks, such as when the search space is high its convergence speed becomes very slow near global optimum. Another drawback is its nature to a fast and premature convergence in mid optimum points.

5. NEW METHOD FOR FORECASTING USING FUZZY TIME SERIES AND PARTICLE SWARM OPTIMIZATION

Let $\mathrm{Y}(\mathrm{t}),(t=...,0,1,2,...)$ be the universe of discourse and $Y(t)\subseteq R$. Assume that $f_i(t),\ i=1,2,...$ is defined in the universe of discourse $\mathrm{Y}(\mathrm{t})$ and $F(t)$ is a collection of $\mathrm{f}(\mathrm{t_i}),(i=...,0,1,2,...)$, then $F(t)$ is called a fuzzy time series of $\mathrm{Y}(\mathrm{t}),\ i=1,2,......$ Using fuzzy relation, we define $F(t)=F(t-1)\circ R(t,t-1)$, where $R(t,t-1)$ is a fuzzy relation and "$\circ$" is the max–min composition operator, then $F(t)$ is caused by $F(t-1)$ where $F(t)$ and $F(t-1)$ are fuzzy sets. For forecasting purpose, we can define relationship among present and future state of a time series with the help of fuzzy sets. Assume the fuzzified data of the ith and $(i+1)th$ day are A_j and A_k, respectively, where $A_j, A_k \in U$, then $A_j \to A_k$ represented the fuzzy logical relationship between A_j and A_k.

Let $F(t)$ be a fuzzy time series. If $F(t)$ is caused by $F(t-1), F(t-2), \dots, F(t-n)$, then the fuzzy logical relationship is represented by

$$F(t-n),...,F(t-2),F(t-1)\to F(t) \tag{5}$$

is called the one-factor nth order fuzzy time series forecasting model. Let $F(t)$ be a fuzzy time series. If $F(t)$ is caused by $(F_1(t-1),F_2(t-1)),(F_1(t-2),F_2(t-2)),\ldots,(F_1(t-n),F_2(t-n))$, then this fuzzy logical relationship is represented by

$$(F_1(t-n),F_2(t-n)),...,(F_1(t-2),F_2(t-2)),(F_1(t-1),F_2(t-1))\to F(t) \tag{6}$$

is called the two-factors nth order fuzzy time series forecasting model, where $F_1(t)$ and $F_2(t)$ are called the main factor and the secondary factor fuzzy time series' respectively. In the similar way, we can define m-factor nth order fuzzy logical relationship as

$$\begin{aligned}&(F_1(t-n),F_2(t-n),...,F_m(t-n)),...,\\&(F_1(t-2),F_2(t-2),...,F_m(t-2)),\\&(F_1(t-1),F_2(t-1),...,F_m(t-1))\to F(t)\end{aligned} \tag{7}$$

Here $F_1(t)$ is called the main factor and $F_2(t),F_3(t),...,F_m(t)$ are called secondary factor fuzzy time series'. Here we can implement any of the fuzzy membership function to define the fuzzy time series in above equations. Comparative study by using different membership functions is also possible. We have used triangular membership function due to low computational cost. Using fuzzy composition rules, we establish a fuzzy inference system for fuzzy time series forecasting with higher accuracy. The accuracy of forecast can be improved by considering higher number of factors and higher dependence on history.

Now we present an extended method for handling forecasting problems based on m-factors high-order fuzzy time series. The proposed method is now presented as follows.

Step 1) Initialize the PSO parameters (weight factor, acceleration factor and maximum iteration) with the predefined values and generate initial random velocity and position of all particles. Divide the particle into two parts. Each part consist of n-1 'X' and m-1 'Y', where the X means the main-factor's each interval and the Y means the second-factor's each interval. In the sequence, the format of each particle is represented as follows: where $x_i^{(1)}\le x_i^{(2)}\le\ldots\le x_i^{(n-1)}$ and $y_i^{(1)}\ge y_i^{(2)}\ge\ldots\ge y_i^{(m-1)}$, for i=1,2,…,c; c is a swarm size.

Step 2) Take particles one by one and using values of first part define the universe of discourse U of the main factor $U=[D_{\min}-D_1,D_{\max}-D_2]$, where $D_{\min}$ and $D_{\max}$ are the minimum and the maximum values of the main factor of the known historical data, respectively, and D1, D2 are two proper positive real numbers to divide the universe of discourse into n equal length intervals $u_1,u_2,...,u_l$. Similarly by using values of second part of particle define the universes of discourse $\mathbf{V}_i,\ i=1,2,...,m-1$ of the secondary-factors $\mathbf{V}_i=[(\mathbf{E_i})_{\min}-\mathbf{E}_{i1},(\mathbf{E_i})_{\max}-\mathbf{E}_{i2}]$, where $(\mathbf{E_i})_{\min}=[(E_1)_{\min},(E_2)_{\min},...,(E_m)_{\min}]$ and $(\mathbf{E_i})_{\max}=[(E_1)_{\max},(E_2)_{\max},...,(E_m)_{\max}]$ are the minimum and maximum values of the secondary-factors of the known historical data, respectively, and $\mathbf{E_{i1}}$, $\mathbf{E_{i2}}$ are vectors of proper positive numbers to divide each of the universe of discourse $\mathbf{V}_i,\ i=1,2,...,m-1$ into equal length intervals termed as $\mathbf{v}_{1,l},\mathbf{v}_{2,l},...,\mathbf{v}_{m-1,l},\ l=1,2,...,p$, where $\mathbf{v}_{1,l}=[\mathbf{v}_{1,1},\mathbf{v}_{1,2},...\mathbf{v}_{1,p}]$ represents n intervals of equal length of universe of discourse $\mathbf{V_1}$ for first secondary-factor fuzzy time series. Thus we have $(m-1)\times l$ matrix of intervals for secondary-factors.

Step 3) Define the linguistic term A_i represented by fuzzy sets of the main factor shown as follows:

$$
\begin{aligned}
A_1 &= \tfrac{1}{u_1} + \tfrac{0.5}{u_2} + \tfrac{0}{u_3} + \tfrac{0}{u_4} + \dots + \tfrac{0}{u_{l-2}} + \tfrac{0}{u_{l-1}} \\
A_2 &= \tfrac{0.5}{u_1} + \tfrac{1}{u_2} + \tfrac{0.5}{u_3} + \tfrac{0}{u_4} + \dots + \tfrac{0}{u_{l-2}} + \tfrac{0}{u_{l-1}} \\
&\;\;. \\
&\;\;. \\
A_n &= \tfrac{0}{u_1} + \tfrac{0}{u_3} + \tfrac{0}{u_4} + \dots + \tfrac{0}{u_{l-2}} + \tfrac{0.5}{u_{l-1}} + \tfrac{1}{u_l}
\end{aligned}
\quad (8)
$$

Similarly, for ith secondary fuzzy time series, we define the linguistic term $B_{i,j}, i = 1, 2, ..., m-1, \; j = 1, 2, ..., n$ represented by fuzzy sets of the secondary-factors,

$$
\begin{aligned}
B_{i,1} &= \tfrac{1}{\mathbf{v}_{i,1}} + \tfrac{0.5}{\mathbf{v}_{i,2}} + \tfrac{0}{\mathbf{v}_{i,3}} + \tfrac{0}{\mathbf{v}_{i,4}} + \dots + \tfrac{0}{\mathbf{v}_{i,l-1}} + \tfrac{0}{\mathbf{v}_{i,l}} \\
B_{i,2} &= \tfrac{0.5}{\mathbf{v}_{i,1}} + \tfrac{1}{\mathbf{v}_{i,2}} + \tfrac{0.5}{\mathbf{v}_{i,3}} + \tfrac{0}{\mathbf{v}_{i,4}} + \dots + \tfrac{0}{\mathbf{v}_{i,l-1}} + \tfrac{0}{\mathbf{v}_{i,l}} \\
&\;\;. \\
&\;\;. \\
B_{i,n} &= \tfrac{0}{\mathbf{v}_{i,1}} + \tfrac{0}{\mathbf{v}_{i,2}} + \tfrac{0}{\mathbf{v}_{i,3}} + \dots + \tfrac{0}{\mathbf{v}_{i,l-2}} + \tfrac{0.5}{\mathbf{v}_{i,l-1}} + \tfrac{1}{\mathbf{v}_{i,l}}
\end{aligned}
\quad (9)
$$

Step 4) Fuzzify the historical data described as follows. Find out the interval $u_l, l = 1, 2, ..., p$ to which the value of the main factor belongs. Table 2 shows Fuzzified yearly data for accidents mortalities.

Case 1) If the value of the main factor belongs to u_1, then the value of the main factor is fuzzified into $\tfrac{1}{A_1} + \tfrac{0.5}{A_2} + \tfrac{0.0}{A_3}$, denoted by X_1.

Case 2) If the value of the main factor belongs to $u_l, l = 2, 3, ..., p-1$ then the value of the main factor is fuzzified into $\tfrac{0.5}{A_{i-1}} + \tfrac{1}{A_i} + \tfrac{0.5}{A_{i+1}}$, denoted by X_i.

Case 3) If the value of the main factor belongs to u_p, then the value of the main factor is fuzzified into $\tfrac{0}{A_{n-2}} + \tfrac{0.5}{A_{n-1}} + \tfrac{1}{A_n}$, denoted by X_n.

Now, for ith secondary-factor, find out the interval $\mathbf{V}_{i,l}$ to which the value of the secondary-factor belongs.

Case 1) If the value of the ith secondary-factor belongs to $\mathbf{V}_{i,1}$, then the value of the secondary-factor is fuzzified into $\tfrac{1}{B_{i,1}} + \tfrac{0.5}{B_{i,2}} + \tfrac{0}{B_{i,3}}$, denoted by $\mathbf{Y}_{i,1} = \left[Y_{1,1}, Y_{2,1}, ..., Y_{m-1,1} \right]$.

Case 2) If the value of the ith secondary-factor belongs to $\mathbf{v}_{i,l}, l = 2, 3, .., p-1$, then the value of the ith secondary-factor is fuzzified into $\tfrac{0.5}{B_{i,j-1}} + \tfrac{1}{B_{i,j}} + \tfrac{0.5}{B_{i,j+1}}, \; j = i = 2, 3, ..., n-1$ denoted by $Y_{i,j}$, where $j = 2, 3, ..., n-1$.

Case 3) If the value of the ith secondary-factor belongs to $\mathbf{V}_{i,p}$, then the value of the secondary-factor is fuzzified into $\tfrac{0}{B_{i,n-2}} + \tfrac{0.5}{B_{i,n-1}} + \tfrac{1}{B_{i,n}}$, denoted by $Y_{i,n}$.

Step 5) Get the m-factors kth-order fuzzy logical relationships based on the fuzzified main and secondary factors from the fuzzified historical data obtained in Step 3).

Table 2. Fuzzified yearly data for mortality accidents from 1974 to 2004

Year	Mortality Accidents	Fuzzified Mortality Accidents
2004	953	0.5/A1 + 1.0/A2 + 0.5/A3 (X2)
2003	1,035	0.5/A1 + 1.0/A2 + 0.5/A3 (X2)
2002	1,145	0.5/A2 + 1.0/A3 + 0.5/A4 (X3)
2001	1,288	0.5/A4 + 1.0/A5 + 0.5/A6 (X5)
2000	1,253	0.5/A4 + 1.0/A5 + 0.5/A6 (X5)
1999	1,173	0.5/A3 + 1.0/A4 + 0.5/A5 (X4)
1998	1,224	0.5/A3 + 1.0/A4 + 0.5/A5 (X4)
1997	1,150	0.5/A3 + 1.0/A4 + 0.5/A5 (X4)
1996	1,122	0.5/A2 + 1.0/A3 + 0.5/A4 (X3)
1995	1,228	0.5/A3 + 1.0/A4 + 0.5/A5 (X4)
1994	1,415	0.5/A5 + 1.0/A6 + 0.5/A7 (X6)
1993	1,346	0.5/A4 + 1.0/A5 + 0.5/A6 (X5)
1992	1,380	0.5/A5 + 1.0/A6 + 0.5/A7 (X6)
1991	1,471	0.5/A6 + 1.0/A7 + 0.5/A8 (X7)
1990	1,574	0.0/A6 + 0.5/A7 + 1.0/A8 (X8)
1989	1,488	0.5/A6 + 1.0/A7 + 0.5/A8 (X7)
1988	1,432	0.5/A5 + 1.0/A6 + 0.5/A7 (X6)
1987	1,390	0.5/A5 + 1.0/A6 + 0.5/A7 (X6)
1986	1,456	0.5/A6 + 1.0/A7 + 0.5/A8 (X7)
1985	1,308	0.5/A4 + 1.0/A5 + 0.5/A6 (X5)
1984	1,369	0.5/A5 + 1.0/A6 + 0.5/A7 (X6)
1983	1,479	0.5/A6 + 1.0/A7 + 0.5/A8 (X7)
1982	1,464	0.5/A6 + 1.0/A7 + 0.5/A8 (X7)
1981	1,564	0.0/A6 + 0.5/A7 + 1.0/A8 (X8)
1980	1,616	0.0/A6 + 0.5/A7 + 1.0/A8 (X8)
1979	1,572	0.0/A6 + 0.5/A7 + 1.0/A8 (X8)
1978	1,644	0.0/A6 + 0.5/A7 + 1.0/A8 (X8)
1977	1,597	0.0/A6 + 0.5/A7 + 1.0/A8 (X8)
1976	1,536	0.5/A6 + 1.0/A7 + 0.5/A8 (X7)
1975	1,460	0.5/A6 + 1.0/A7 + 0.5/A8 (X7)
1974	1,574	0.0/A6 + 0.5/A7 + 1.0/A8 (X8)

If the fuzzified historical data of the main-factor of ith day is X_i, then construct the m-factors kth-order fuzzy logical relationships,

$$
\begin{aligned}
&\left(X_{j-k}; Y_{2,j-k}, ..., Y_{m-1,j-k} \right), ..., \left(X_{j-2}; Y_{2,j-2}, ..., Y_{m-1,j-2} \right), \\
&\left(X_{j-1}; Y_{1,j-1}, Y_{2,j-1}, ..., Y_{m-1,j-1} \right), \rightarrow X_j
\end{aligned}
\quad (10)
$$

where $j > k$. X_{j-k} shows the k-step dependence of jth value of main factor X_j, $Y_{i,j-k}, i = 1, ..., m-1, \; j = 1, ..., k$. Then, divide the derived fuzzy logical relationships into fuzzy logical relationship groups based on the current states of the fuzzy logical relationships. The secondary factors acts like a secondary component to the m-dimensional state vector.

Step 6) For m-factor kth order fuzzy logical relationship, the forecasted value of day j based on history of third order is calculated as follows,

$$t_j = \frac{\sum_{j-1}^{j+1} w}{\frac{w_{j-1}}{a_{j-1}} + \frac{w_j}{a_j} + \frac{w_{j+1}}{a_{j+1}}} \quad (11)$$

Where a_{l-1}, a_l and a_{l+1} are the midpoints of the intervals u_{l-1}, u_l and u_{l+1} respectively. Above forecasting formula fulfills the axioms of fuzzy sets like monotonicity, boundary conditions, continuity and idempotency. For measurement of accuracy of forecasting for fuzzy time series forecasting, we use average forecasting error rate (AFER) as the performance criteria, defined as

$$AFER = \frac{\sum_{i=1}^{n} \left(\left| Forecated\ value\ of\ day\ j - Actual\ value\ of\ day\ j \right| / Actual\ value\ of\ day\ j \right)}{n} \times 100\% \quad (12)$$

Step 7) Compare AFER value of current particle with local best particle's AFER value and set current particle as local best in case current AFER is lower than local best AFER value.

Step 8) Repeat steps 2 to 7 for all the particles in swarm, and now pick up the particle with minimum AFER from all the particles' local bests. Set this as global best particle.

Step 9) For each particle, calculate particle velocity according to Eq.

$$v_i^{(t+1)} = w^{(t)} \times v_i^{(t)} + c_1 \times r_1 \times (pbest - x_i^{(t)}) + c_2 \times r_2 \times (gbest - x_i^{(t)}) \quad (13)$$

This is based on the gbest and the pbest values and then update particle position according to Eq.

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (14)$$

This step will lead the each particle to a more promising solution for optimal tuning of two-factors nth-order fuzzy time series.

Step 10) Update the value of the weight factor w according to Eq.

$$w^{(t)} = w_{\max} - t \times (w_{\max} - w_{\min}) / iter_{\max} \quad (15)$$

Step 11) Stop if the stop criterion of the PSO, the maximum number of iteration, is satisfied. Otherwise, go to Step 2. The final gbest value after stopping PSO iterations will be our required result.

6. EXPERIMENT

We have applied this new technique on car road accident data taken from National Institute of Statistics, Belgium for the period of 1974-2005. In this data, the main factor of interest is the yearly road accident casualties and secondary factors are mortally wounded, died within one month, severely wounded and light casualties. We have used 11 particles in a swarm and executed PSO for 500 iterations to get best results.

We assumed eight intervals of equal length for the main and secondary fuzzy time series'. For main factor, we assume $D_{\min}$ =953 and $D_{\max}$ =1644, thus for main factor time series we get $U = [850, 1650]$. Similarly for secondary factors Y_1, Y_2, Y_3 and Y_4, we assumed that $\mathbf{E}_{\min} = [90, 1094, 5949, 38390]$ and $\mathbf{E}_{\max} = [819, 2393, 16645, 46818]$

Table 3. Forecasted yearly car road accident casualties from 1974-2004

Actual Killed Ai	Forecasted Kills Fi	Fi - Ai	$\left\lvert \frac{F_i - A_i}{A_i} \right\rvert$
953	966.7723	13.772	0.01445152
1,035	1022.684	-12.316	0.01189952
1,145	1155.396	10.396	0.00907948
1,288	1318.751	30.751	0.023875
1,253	1231.045	-21.955	0.01752195
1,173	1155.396	-17.604	0.01500767
1,224	1231.045	7.045	0.00575572
1,150	1155.396	5.396	0.00469217
1,122	1092.966	-29.034	0.02587701
1,228	1231.045	3.045	0.00247964
1,415	1399.624	-15.376	0.01086643
1,346	1318.751	-27.249	0.02024443
1,380	1399.624	19.624	0.01422029
1,471	1483.872	12.872	0.00875051
1,574	1561.875	-12.125	0.0077033
1,488	1483.872	-4.128	0.00277419
1,432	1399.624	-32.376	0.02260894
1,390	1399.624	9.624	0.00692374
1,456	1483.872	27.872	0.01914286
1,308	1318.751	10.751	0.00821942
1,369	1318.751	-50.249	0.03670489
1,479	1483.872	4.872	0.00329412
1,464	1483.872	19.872	0.01357377
1,564	1561.875	-2.125	0.0013587
1,616	1561.875	-54.125	0.03349319
1,572	1561.875	-10.125	0.00644084
1,644	1561.875	-82.125	0.04995438
1,597	1561.875	-35.125	0.02199436

1,536	--	--	--
1,460	--	--	--
1,574	--	--	--

to determine $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$. Selection of $D_{\min}$, $D_{\max}$, $\mathbf{E}_{\min}$ and $\mathbf{E}_{\max}$ have significant effects on the accuracy of this new method.

We can introduce learning to stabilize the heuristic selection of these constants.

Using (8) and (9), we formed fuzzy times series from main and secondary factors. Therefore, each observation of a time series is now represented by a combination of fuzzy sets. Using (11), we calculated the forecasted values corresponding to each actual value of the main factor time series in Table 3. Using equation (12) for AFER, we formed Table 4, showing a comparison of actual and forecasted values. Finally, we have red proposed method with [4].

7. CONCLUSION

From Table 4, we can see that our proposed method is better that Lee, Wang Chen [4]. As the work of Lee, Wang Chen [4] outperformed the work of Huarng ([2], [19]) and Chen [3] so indirectly we can conclude that our general class of methods for fuzzy time series modeling and forecasting.

Furthermore, we have shown fuzziness of fuzzy observations by presenting each datum of the main series as composed of many fuzzy sets. Thus, fuzzy time series modeling extends to type-II fuzzy time series modeling. The type-II defuzzified forecasted values (t_j) may also be calculated using some other method, e.g. learning rules from fuzzy time series.

TABLE 4. Comparison of proposed method with other researches for yearly car road accident casualties in Belgium 1974-2004

Reference	AFER
Jilani and Burney (2008)	2.06%
Jilani and Burney (2008)	4.69%
Lee et. Al (2006)	4.72%
Lee L.W. (2006)	5.07%
Jilani and Burney (2007)	5.06%
Proposed Method	**1.49%**

8. ACKNOWLEDGMENTS

The work is completed with partial support from Higher Education Commission of Pakistan and Dean Faculty of Science, University of Karachi.

9. REFERENCES

[1] Q. Song and B. S. Chissom 1993a “Forecasting Enrollments with Fuzzy Time Series—Part I”, *Fuzzy Sets and System*, Vol. 54, No. 1, pp. 1–9.

[2] K. Huarng, 2001a “Heuristic models of fuzzy time series for forecasting,” *Fuzzy Sets and Systems*, Vol. 123, No. 3, pp. 369–386.

[3] S. M. Chen, 2002 “Forecasting Enrollments Based on High-Order Fuzzy Time Series”, *Cybernetic Systems*, Vol. 33, No. 1, pp. 1–16.

[4] L. W. Lee, L. W. Wang, S. M. Chen, Jun. 2006 “Handling Forecasting Problems Based on Two-Factors High-Order Time Series”, *IEEE Transactions on Fuzzy Systems*, Vol. 14, No. 3, pp.468-477.

[5] Melike Sah and Y. D. Konstsntin, 2004 “Forecasting Enrollment Model based on first-order fuzzy time series”, Published in proc., International Conference on Computational Intelligence, Istanbul, Turkey.

[6] T. A. Jilani, S. M. A. Burney and C. Ardil, 2007a “Fuzzy metric approach for fuzzy time series forecasting based on frequency density based partitioning”, *International Journal of Computational Intelligence*, Vol. 4, No. 2, pp 12-18.

[7] T. A. Jilani, S. M. A. Burney, 2007b “*M-Factor High Order Fuzzy Time Series Forecasting for Road Accident Data*”, Advances in Soft Computing vol. 41, pp.246-257.

[8] T. A. Jilani, S. M. A. Burney, 2008a “Multivariate High Order Fuzzy Time Series Forecasting for Car Road Accidents”, International Journal of Computational Intelligence, Vol. 4, No. 1, pp. 15-20.

[9] T. A. Jilani, S. M. A. Burney, 2008b “Multivariate stochastic fuzzy forecasting models”, *Expert Systems With Applications*, vol. 37 No. 2, pp. 691-700.

[10] H. J. Zimmerman, 2001 “*Fuzzy set theory and its applications”*, Kluwer Publishers, Boston, MA.

[11] H. Ishibuchi, R. Fujioka and H. Tanaka, 1993 "Neural Networks that Learn from Fuzzy If-Then Rules", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, pp.85-97.

[12] G. J. Klir and B. Yuan, 2005 “*Fuzzy Sets and Fuzzy Logic: Theory and Applications”*, Prentice Hall, India, Ch. 4.

[13] R. R. Yager and P. P. D. Filev, 2002 “*Essentials of Fuzzy Modeling and Control”*, John Wiley and Sons, Inc..

[14] B. J. MacLennan, 2004, “Natural computation and non-turing models of computation”, Theoretical Computer Science, (1–3), pp. 115–145.

[15] P. Marrow, 2000 “Nature-inspired computing technology and applications”. BT Technology Journal (4), pp. 13–23.

[16] A. P. Deweijer, C. B. Lucasius, L. Buydens 1994, “Curve-fitting using natural computation”, Analytical Chemistry 66(1), pp. 23–31.

[17] J. Kennedy and R. Eberhart, 2001 “Swarm Intelligence”. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

[18] Maurice Clerc, 2006, “Particle Swarm Optimization”, ISTE Ltd, USA.

[19] K. Huarng, 2001b “Effective Lengths of Intervals to Improve Forecasting in Fuzzy Time Series”, *Fuzzy Sets System*, Vol. 123, No. 3, pp. 387–394.