Epidemiological Dynamics Modeling by Fusion of Soft Computing Techniques

Thanh Nguyen Abbas Khosravi Douglas Creighton Saeid Nahavandi

***Abstract*—Infectious disease prevention and control are important in improving, promoting and protecting the health of communities. Epidemiological data analysis plays a crucial role in disease prevention and control. Conventional methods such as moving average or autoregressive analysis normally require the assumption of stationarity, which is often violated in epidemiologic time series. This paper proposes the fusion of neural networks, fuzzy systems and genetic algorithms, with the aim to strengthen the modeling power for epidemiological dynamics. We deploy an additive fuzzy system into a neural network architecture in order to incorporate recurrent nodes to enable the fuzzy system to handle temporal data. The genetic algorithm is employed to optimize the fuzzy rule structure before supervised training is applied to adjust parameters. As epidemiological time series exhibit complex behavior and possibly cyclic patterns, the addition of recurrent nodes to the fuzzy system improves the modeling capability. The proposed model dominates the benchmark feedforward neural network and adaptive neuro-fuzzy inference system model regarding modeling performance. Through real applications for epidemiologic time series modeling, the fusion of soft computing techniques offer accurate forecasts that have considerable meaning in planning infectious disease-control activities.**

***Index Terms*—*epidemiological modeling; disease trend detection; neural network; fuzzy system; standard additive model (SAM); Recurrent SAM; genetic algorithm*

I. Introduction

Careful examination regarding the dynamics and variations of notifiable diseases is crucial in epidemiological researches in the increasing global infectious disease risks. The pandemic H1N1 influenza A occurred from April 2009 to August 2010 is a typical example, which killed 18,500 people worldwide [1]. The number of deaths associated with the pandemic is actually much higher than that laboratory-confirmed number. There were approximately 60.8 million cases and 12,469 deaths occurred in the United States alone between April 2009 and April 2010, the estimation performed by the U.S. Centers for Disease Control and Prevention [2]. The pandemic particularly affected young individuals, causing a substantial number of years of life lost, between 334,000 and 1,973,000, from May to December 2009 in the U.S. [3, 4].

Epidemiology is the essential means used by public health practitioners to understand causes of disease and to develop control measures. Epidemiology is able to render early detection of infectious cases and rapid public health intervention so that the public's health can be protected and improved. Public health authorities rely on epidemiological studies to deliver better health outcomes, improve longevity of people, reduce mortality rate in the communities. The quality of health care provided to the public can be constantly improved and ensuring people living healthier and more independent lives. Time series analysis is an important and useful tool to delineate disease patterns in the population over time, examine and quantify the evolution of the infectious diseases. It provides a quick and feasible means for analyzing disease trends, detecting sudden changes in disease occurrence and distribution, and determining relative disease burdens to plan disease-control activities. Watier and Richardson [5] modeled the time series of reported cases of Salmonella typhimurium in France where nonlinearity is present using the seft-exciting threshold autoregressive model. Koelle and Pascual [6] identified the respective contributions of extrinsic and intrinsic factors in the historical cholera mortality data in Dhaka in Bangladesh using another nonlinear time series model. Interannual fluctuations and long-term changes in transmissibility were addressed properly through reconstruction of the pattern of decaying immunity from time series data on cases and population sizes. Cazelles *et al.* [7] alternatively detected the shifts in the periodic components in time series of the weekly measles notifications in the city of York (UK) using wavelet analysis. The wavelet approach was additionally employed to examine the transient relationship between cholera incidence in Ghana and the El Niño Southern oscillation and also to explore the transient nature of the spatial synchrony between dengue incidence in Bangkok and that in the rest of Thailand.

Epidemiologic time series are noisy, dynamic and non-stationary so that traditional time series methods usually demonstrate inadequate handling capability. Researchers therefore advocate the use of soft computing techniques, e.g. neural network (NN), fuzzy systems (FS), to model such nonlinear data. NN and FS are both universal approximators but either of these methods exhibits advantages and also disadvantages. The fusion of these two techniques thus has soon emerged. This fusion can be divided into two main streams. The first is to use NN to estimate parameters of FS and another is to implement FS in adaptive NN [8, 9]. We focus on the later in this study.

Building FS in adaptive NN is motivated by a number of reasons. Firstly there are some disadvantages of FS such as the difficult determination of the correct set of rules and membership functions, and also the limitation in learning

Thanh Nguyen, Abbas Khosravi, Douglas Creighton, and Saeid Nahavandi are with the Centre for Intelligent Systems Research (CISR), Deakin University, Victoria, Australia (email: {thanh.nguyen, abbas.khosravi, douglas.creighton, and saeid.nahavandi}@deakin.edu.au)

and generalization capability. With fuzzy systems, there are no standard methods to transform human knowledge to fuzzy rules [10]. Furthermore, the learning and generalization capabilities of NN are essential to be integrated into a FS to enable it to deal with nonlinear, time variant problems more effectively [9, 11]. FS thus can be equipped with the diverse and powerful learning capacity of NN. As a FS is represented by a network, it is intuitively plausible to include extensions or modifications that adapt specific applications: advanced features available in NN can be utilized in FS such as the possibility to include recurrent cycles, which enable embedding temporal knowledge or relations [11, 12]. This feature is not possible in conventional fuzzy logic [9]. In general, deployment of FS in adaptive NN helps integrate advantages of NN and eliminate disadvantages of FS.

Fuzzy inference systems are commonly categorized in three types: Tsukamoto, Mamdani, and Takagi-Sugeno-Kang (TSK) models [10, 13]. With the Tsukamoto and TSK models, equivalent adaptive network architectures and learning procedures have been suggested by Jang [10]. There, however, has not yet been an equivalent adaptive neural network designed for the Mamdani models in the literature. This study proposes an equivalent adaptive network for the Standard Additive Model (SAM) that belongs to the Mamdani fuzzy system category. Subsequently, to demonstrate advantages of the SAM-equivalent NN, we incorporate recurrent nodes into it to formulate the so-called Recurrent SAM (RSAM) that is capable of handling temporal relations. Given that the epidemiological time series show strong cyclic or periodic oscillations, they are modeled effectively by the proposed RSAM. The next section presents SAM and its proposed equivalent adaptive NN. Section III is devoted to the RSAM structure and the integration of genetic algorithm to optimize the RSAM rule structure for better modeling. Experimental results and discussions are presented in Section IV followed by conclusions.

II. Additive Fuzzy System and the Proposed Equivalent Neural Network

A. The additive fuzzy system

The additive fuzzy system (Fig. 1) or SAM introduced by Kosko [14-16] is characterized by the Mamdani fuzzy rules, sum-product inference, and CoG defuzzifier.

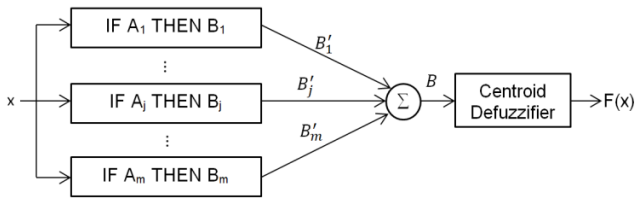

Fig. 1. The parallel structure of SAM

The additive fuzzy system $F: R^n \to R^p$ stores m if-then rules pertaining to different weights w and can be used to uniformly approximate functions, which are continuous and bounded measurable on a compact domain. Any choice of the if-part fuzzy sets $A_j \subset R^n$ that have joint set function a_j: $R^n \to [0,1]$ and factors: $a_j(x) = a_j^1(x_1) \dots a_j^n(x_n)$ can be employed. From the vector input $x \in R^n$, any choice of the then-part fuzzy sets $B_j \subset R^p$ can be deployed as only the centroid c_j and volume V_j of B_j of the system are used to calculate the output $F(x)$

$$F(x) = Centroid\left(\sum_{j=1}^{m} w_j a_j(x) B_j\right) = \frac{\sum_{j=1}^{m} w_j a_j(x) V_j c_j}{\sum_{j=1}^{m} w_j a_j(x) V_j}$$

B. SAM-equivalent NN structure

The following 7-layer neural network (Fig. 2) is proposed to be equivalent to the SAM with n inputs and m fuzzy rules.

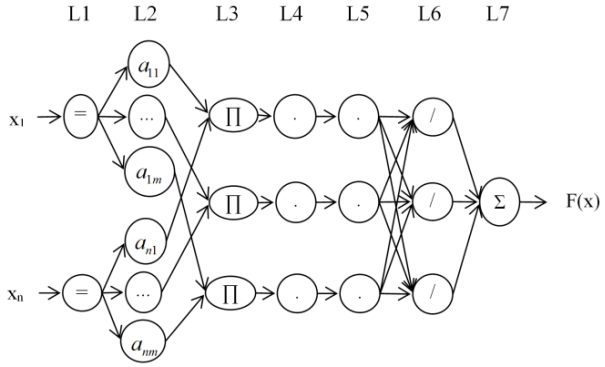

Fig. 2. SAM-equivalent NN structure

Let us denote $u_i^{(k)}$ and $o_i^{(k)}$ are input and output values of the i^{th} node in the k^{th} layer.

Layer 1:

$o_i^{(1)} = u_i^{(1)} = x_i(t)$, where $i = 1,2,\dots,n$.

Layer 2: Various kinds of fuzzy sets can be used in this layer. There is no loss of generality to assume the Gaussian function is used:

$$a_{ij}(x_i) = o_{ij}^{(2)} = \exp\left[-\frac{(o_i^{(1)} - m_{ij})^2}{(\sigma_{ij})^2}\right], \text{ where } i = 1,2,\dots,n,$$

$j = 1,2,\dots,m$ and m_{ij} and σ_{ij} are respectively centre and width of the Gaussian fuzzy set.

Layer 3:

$$a_j(x) = o_j^{(3)} = \prod_{i=1}^{n} o_{ij}^{(2)} = \prod_{i=1}^{n} a_{ij}(x_i)$$

Layer 4:

$$o_j^{(4)} = o_j^{(3)} w_j = a_j(x) w_j$$

Layer 5:

$$o_j^{(5)} = o_j^{(4)} V_j = a_j(x) w_j V_j$$

Layer 6:

$$p_j(x) = o_j^{(6)} = \frac{o_j^{(5)}}{\sum_{j=1}^{m} o_j^{(5)}} = \frac{a_j(x) w_j V_j}{\sum_{i=1}^{m} a_i(x) w_i V_i} \quad (1)$$

Layer 7:

$$F(x) = o^{(7)} = \sum_{j=1}^{m} o_j^{(6)} c_j = \frac{\sum_{j=1}^{m} w_j a_j(x) V_j c_j}{\sum_{j=1}^{m} w_j a_j(x) V_j} \quad (2)$$

Two conditions for the equivalence are established as follows:

a) The number of nodes in Layer 3 of the adaptive network is equal to the number of fuzzy rules in SAM.

b) Membership functions of the adaptive network and of the if-part of SAM are of the same type. It is important to note that the membership function a_{nm} of the adaptive network is corresponding to the component n-th of the if-part of the rule m-th in SAM.

C. Supervised learning in SAM

Kosko [16] applied the gradient descent learning law for a parameter ξ in SAM:

$$\xi(t+1) = \xi(t) - \mu_t \frac{\partial E}{\partial \xi}$$

where μ_t is the learning coefficient. The aim is to minimize the square of errors:

$$E(x) = \frac{1}{2}(f(x) - F(x))^2 \quad (3)$$

The derivatives of the error function are determined as follows:

$$\frac{\partial E}{\partial F} = -(f(x) - F(x)) = -\varepsilon(x),\ \frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial F}.\frac{\partial F}{\partial w_j},$$

$$\frac{\partial E}{\partial V_j} = \frac{\partial E}{\partial F}.\frac{\partial F}{\partial V_j}, \text{ and } \frac{\partial E}{\partial c_j} = \frac{\partial E}{\partial F}.\frac{\partial F}{\partial c_j}.$$

On the other hand:

$$\frac{\partial F}{\partial w_j} = \frac{a_j(x)V_j[c_j - F(x)]}{\sum_{i=1}^{m} w_i a_i(x) V_i} = [c_j - F(x)]\frac{p_j(x)}{w_j}$$

$$\frac{\partial F}{\partial V_j} = \frac{w_j a_j(x)[c_j - F(x)]}{\sum_{i=1}^{m} w_i a_i(x) V_i} = [c_j - F(x)]\frac{p_j(x)}{V_j}$$

$$\frac{\partial F}{\partial c_j} = \frac{w_j a_j(x) V_j}{\sum_{i=1}^{m} w_i a_i(x) V_i} = p_j(x)$$

Then the learning laws for the SAM parameters are as follows:

$$w_j(t+1) = w_j(t) + \mu_t.\varepsilon(x).[c_j - F(x)]\frac{p_j(x)}{w_j} \quad (4)$$

$$V_j(t+1) = V_j(t) + \mu_t.\varepsilon(x).[c_j - F(x)]\frac{p_j(x)}{V_j} \quad (5)$$

$$c_j(t+1) = c_j(t) + \mu_t.\varepsilon(x).p_j(x) \quad (6)$$

To derive the learning laws for parameters of if-part fuzzy sets, we first state the derivative:

$$\frac{\partial F}{\partial a_j} = \frac{w_j V_j[c_j - F(x)]}{\sum_{i=1}^{m} w_i a_i(x) V_i} = [c_j - F(x)]\frac{p_j(x)}{a_j(x)}$$

Assume the Gaussian fuzzy set is used for SAM:

$$a_{ij}(x_i) = \exp\left\{-\left(\frac{x_i(t) - m_{ij}}{\sigma_{ij}}\right)^2\right\}$$

Then the learning laws for parameters of this fuzzy set are:

$$m_{ij}(t+1) = m_{ij}(t) + 2\mu_t.\varepsilon(x).p_j(x).[c_j - F(x)]\frac{x - m_{ij}}{\sigma_{ij}^2} \quad (7)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + 2\mu_t.\varepsilon(x).p_j(x).[c_j - F(x)]\frac{(x - m_{ij})^2}{\sigma_{ij}^3} \quad (8)$$

The learning laws for other kinds of membership functions are shown in [17, 18].

III. Recurrent Standard Additive Model (RSAM) and integration of GA

A. RSAM configuration

The successful design of the equivalent neural network model with SAM helps to flexibly extend conventional SAM to incorporate various advanced features of neural networks.

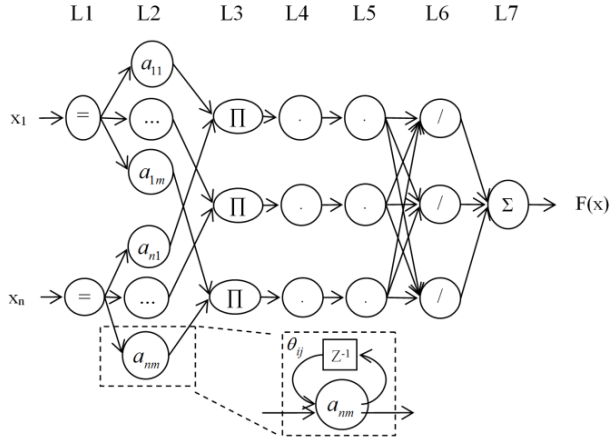

Fig. 3. Recurrent Standard Additive Model (RSAM)

This section presents a proposal of Recurrent SAM (RSAM) (Fig. 3) that incorporates recurrent nodes into SAM. The proposal herein is inspired from the recurrent characteristic of the recurrent fuzzy neural network proposed by Lee and Teng [19] or Sun and Wang [20].

Layer 1:

$o_i^{(1)} = u_i^{(1)} = x_i(t)$, where $i = 1,2,..., n$

Layer 2: Assume the Gaussian function is used herein:

$$o_{ij}^{(2)} = \exp\left[-\frac{(u_{ij}^{(2)} - m_{ij})^2}{(\sigma_{ij})^2}\right], \text{ where } i = 1,2,..., n,\ j = 1,2,..., m$$

The input of nodes in this layer is:

$$u_{ij}^{(2)}(t) = o_i^{(1)} + \theta_{ij} o_{ij}^{(2)}(t-1)$$

where θ_{ij} is the weights of current nodes. The input of nodes in this layer contains $o_{ij}^{(2)}(t-1)$ which stores previous information of the model. This is an extension compared to the SAM system.

Then denote $a_{ij}(x_i)$ as:

$$a_{ij}(x_i) = o_{ij}^{(2)} = \exp\left[-\frac{[o_i^{(1)} + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^2}\right] = \exp\left[-\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^2}\right] \quad (9)$$

Layer 3: Denote $a_j(x)$ as:

$$a_j(x) = o_j^{(3)} = \prod_{i=1}^{n} a_{ij}(x_i) = \prod_{i=1}^{n} o_{ij}^{(2)} = \prod_{i=1}^{n} \exp\left[-\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^2}\right]$$

Layer 4:

$$o_j^{(4)} = o_j^{(3)} w_j = w_j a_j(x)$$

Layer 5:

$$o_j^{(5)} = o_j^{(4)} V_j = w_j V_j a_j(x)$$

Layer 6: Denote $p_j(x)$ as:

$$p_j(x) = o_j^{(6)} = \frac{o_j^{(5)}}{\sum_{j=1}^{m} o_j^{(5)}} = \frac{w_j V_j a_j(x)}{\sum_{j=1}^{m} w_j V_j a_j(x)}$$

In details:

$$p_j(x) = \frac{w_j V_j \prod_{i=1}^{n} \exp\left[-\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^2}\right]}{\sum_{j=1}^{m} w_j V_j \prod_{i=1}^{n} \exp\left[-\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^2}\right]} \quad (10)$$

Layer 7: The final output $F(x)$ of RSAM is then defined:

$$F(x) = o^{(7)} = \sum_{j=1}^{m} o_j^{(6)} c_j = \frac{\sum_{j=1}^{m} w_j V_j a_j(x) c_j}{\sum_{j=1}^{m} w_j V_j a_j(x)}$$

$$= \frac{\sum_{j=1}^{m} w_j V_j c_j \prod_{i=1}^{n} \exp\left[-\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^2}\right]}{\sum_{j=1}^{m} w_j V_j \prod_{i=1}^{n} \exp\left[-\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^2}\right]} \quad (11)$$

B. Supervised learning in RSAM

Similar to the SAM, the tuning laws applied to the rule weights, the volumes and the centroids of the then-part fuzzy sets also use formulae (4), (5) and (6) except $p(x)$ and $F(x)$ are now replaced by (10) and (11) rather than (1) and (2). The major difference in learning laws between SAM and RSAM occurs in the learning of parameters of the if-part fuzzy sets.

The following presents the learning laws for parameters of if-part fuzzy sets in RSAM where the Gaussian membership function is utilized as an example. The learning laws for other kinds of membership functions can be deployed by similar procedure.

In RSAM, the Gaussian fuzzy set is formulated as in (9):

$$a_{ij}(x_i) = \exp\left\{-\left(\frac{x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}}{\sigma_{ij}}\right)^2\right\}$$

Then the derivatives:

$$\frac{\partial a_{ij}}{\partial m_{ij}} = 2a_{ij}(x_i)\frac{x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}}{(\sigma_{ij})^2}$$

$$\frac{\partial a_{ij}}{\partial \sigma_{ij}} = 2a_{ij}(x_i)\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^3}$$

$$\frac{\partial a_{ij}}{\partial \theta_{ij}} = -2a_{ij}(x_i)\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]o_{ij}^{(2)}(t-1)}{(\sigma_{ij})^2}$$

On the other hand, the chain rule allows:

$$\frac{\partial E}{\partial m_{ij}} = \frac{\partial E}{\partial F}\cdot\frac{\partial F}{\partial a_j}\cdot\frac{\partial a_j}{\partial a_{ij}}\cdot\frac{\partial a_{ij}}{\partial m_{ij}}, \quad \frac{\partial E}{\partial \sigma_{ij}} = \frac{\partial E}{\partial F}\cdot\frac{\partial F}{\partial a_j}\cdot\frac{\partial a_j}{\partial a_{ij}}\cdot\frac{\partial a_{ij}}{\partial \sigma_{ij}}$$

$$\frac{\partial E}{\partial \theta_{ij}} = \frac{\partial E}{\partial F}\cdot\frac{\partial F}{\partial a_j}\cdot\frac{\partial a_j}{\partial a_{ij}}\cdot\frac{\partial a_{ij}}{\partial \theta_{ij}}$$

where E is the error function denoted in (3). So that:

$$\frac{\partial E}{\partial m_{ij}} = -2\varepsilon(x)p_j(x)[c_j - F(x)]\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]}{(\sigma_{ij})^2}$$

$$\frac{\partial E}{\partial \sigma_{ij}} = -2\varepsilon(x)p_j(x)[c_j - F(x)]\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^3}$$

$$\frac{\partial E}{\partial \theta_{ij}} = 2\varepsilon(x)p_j(x)[c_j - F(x)]\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]o_{ij}^{(2)}(t-1)}{(\sigma_{ij})^2}$$

Then the tuning law for parameters of the if-part fuzzy sets:

$$m_{ij}(t+1) = m_{ij}(t) + 2\mu_t\varepsilon(x)p_j(x)[c_j - F(x)]\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]}{(\sigma_{ij})^2} \quad (12)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + 2\mu_t\varepsilon(x)p_j(x)[c_j - F(x)]\frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]^2}{(\sigma_{ij})^3} \quad (13)$$

$$\theta_{ij}(t+1) = \theta_{ij}(t) - 2\mu_t\varepsilon(x)p_j(x)[c_j - F(x)]\times \frac{[x_i(t) + \theta_{ij} o_{ij}^{(2)}(t-1) - m_{ij}]o_{ij}^{(2)}(t-1)}{(\sigma_{ij})^2} \quad (14)$$

Note the difference between (12), (13) in RSAM and (7), (8) in SAM. The formula (14) is the current parameter needing to be trained that exists only in RSAM and makes the difference between RSAM and SAM. When this parameter equals zero then RSAM diminishes to SAM. Therefore SAM is a special case of RSAM.

C. Fusion of soft computing techniques: Integration of Genetic Algorithm (GA) to optimize RSAM structure

The exponential rule explosion is a typical curse of a fuzzy system using factorable if-part fuzzy sets in high dimensions [21]. A fuzzy system with more rules can better approximate nonlinear functions but it would lead to overfiting and more training computational cost compared to the one with less rules. In Nguyen at al. [22] we employed GA to optimize the fuzzy rule structure of SAM for function approximation and found the efficiency of GA optimizer. Likewise, we use GA to optimize the structure of RSAM fuzzy rules in this research. The RSAM is the fusion of neural network and fuzzy system. This fusion is further integrated with GA so that the proposed model is henceforth called Neuro-Fuzzy-Genetic Fusion (NFGF) which is graphically presented in Fig. 4.

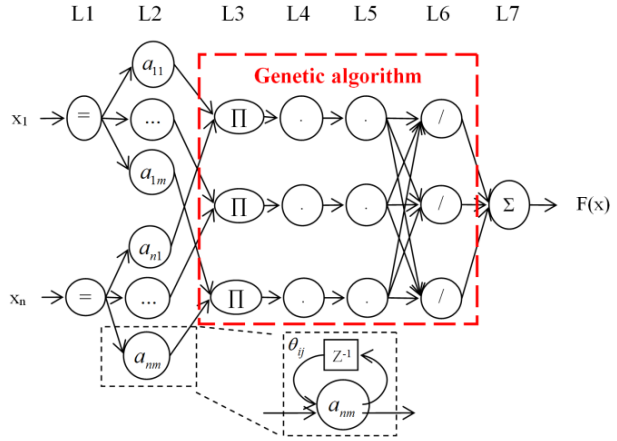

Fig. 4. Integration of GA to optimize RSAM rule structure

The GA [23] is an algorithm operating based on a population of individuals, which interact together and evolved through generations following the Darwinian principle of survival of the fittest. The GA can solve problems where objective functions are complicated and cannot be addressed by conventional methods [24, 25].

Using the GA method, rule nodes in the NFGF model that play insignificant roles in the approximation will be removed and the number of rule nodes therefore is declined. A fuzzy system with less fuzzy rules can be trained and convergent faster compared to the one with more rules.

Depending on the problem being solved, the GA fitness function designed may vary. In this paper, the fitness function is constructed to simultaneously minimize RMSE and the number of fuzzy rules as follows

$$Fit(m) = \ln(RMSE) + \frac{\log_n(m)}{n}$$

where $RMSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - F(x_i))^2$, n is the number of data patterns used for learning and m is the number of fuzzy rules.

The fuzzy rules are coded "0" and "1" where the "0" means that the corresponding rule should be eliminated whilst the "1" means the fuzzy rule should be used. Three basic genetic operators including: selection, crossover and mutation are performed repeatedly through a determined number of evolutionary generations. The individual in the last generation with the lowest fitness value is selected in order to construct the optimal rule structure for RSAM.

IV. Epidemiological time series modeling

A. Datasets and model calibrations

The data used in this study are the occurrence of notifiable infectious diseases in the United States and in the New York City.

Prevalent notifiable diseases including tuberculosis (TB) and varicella (chickenpox) are investigated due to their high risks of infection and the large number of reported cases throughout the U.S. The monthly TB time series data spanning from January 1993 to December 2010 are derived from the summary of notifiable infectious diseases of the Morbidity and Mortality Weekly Report published by the U.S. Centers for Disease Control and Prevention. For the New York City alone, the historical monthly reported number of chickenpox cases from 1931-1972 are also obtained from the Time Series Data Library available on DataMarket.com [26].

All data are transformed using natural logarithm to diminish sensitivity of the modeling to outliers. Lags of time series are utilized as independent variables. For the monthly epidemiologic time series demonstrate strong periodic patterns, we coded months from 1 to 12 as an additional input for the forecasting models. Balkin and Ord [27] determined the number of maximum lags depending on the frequency of the data, i.e. yearly data with 4 lags, quarterly data with 6 lags and monthly data with 15 lags. From the determined maximum number of lags, as with Balkin and Ord [27], we herein employ a forward stepwise regression procedure to eliminate less significant inputs. The use of the forward stepwise regression helps to reduce the number of independent variables so the forecasting mechanism can operate better to avoid excessive number of modifiable parameters. Each experimental dataset is divided into two parts: training (80%) and testing (20%) datasets. After training, the testing dataset is used to test the performance of the forecasting models. The direct strategy is applied to forecasting rather than recursive one to avoid accumulative error, especially for long forecasting period.

The experiments are to compare performance of the proposed NFGF model versus the benchmark feedforward neural network (FFNN) models and the widely used adaptive neuro-fuzzy inference system (ANFIS) proposed in [10]. The major difference between NFGF and ANFIS is in the recurrent nodes and utilization of GA although both ANFIS and NFGF are fuzzy models. FFNN is one of the popular neural network models where the data are input and transferred forwardly without loops or cycles. We deploy the FFNN with two hidden layers for comparisons. The number of nodes in the hidden layers are assigned 10, 30 and 50 corresponding to three models denoted as NN(10-10), NN(30-30) and NN(50-50). As parameters of the NN models are initialized randomly, we repeat running these models 10 times and then average out the forecasting results. The comparisons of the fusion against NN and ANFIS highlight the importance of the recurrent nodes embedded in the proposed models and the efficiency of the GA employment in optimizing the rule structure.

In the fuzzy approximation, there is no theorem stating which membership function is the best for an if-part fuzzy set or how many rules should be used in a fuzzy system. Furthermore, there is no recognized approach for initialization in order to obtain global optima. Different initialization leads to different optimal convergence. For the sake of comparisons, we use the Gaussian fuzzy set for both RSAM and ANFIS. The weights of RSAM fuzzy rules and the volumes of the then-part fuzzy sets are all set at 0.5 whereas the width of the if-part fuzzy sets is set equal to the standard deviation of each series. The centers of the if-part and centroids of the then-part fuzzy sets are assigned by centers of resulting clusters obtained based on the Adaptive Vector Quantization clustering method under the unsupervised competitive learning law (see [14, 21]). The GA population of 20 individuals is initialized for the GA optimizer, which is evolved through 1,000 generations. One individual in the initial population is characterized by all "1" for its genes so that every rule has an equal chance to contribute to the final NFGF model.

Along with RMSE, two other prediction performance criteria are also utilized that are Correlation Coefficient (CC) and coefficient of determination (R^2). Both are expressed in percentage through the following formulae:

$$CC = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(\tilde{y}_i - \bar{\tilde{y}})}{\sqrt{\left[\sum_{i=1}^{N}(y_i - \bar{y})^2 \sum_{i=1}^{N}(\tilde{y}_i - \bar{\tilde{y}})^2\right]}}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

where y_i, $\tilde{y}_i$, $\bar{y}$, $\bar{\tilde{y}}$ are the ith actual value, the ith forecasted value, the mean of actual values, and the mean of forecasted values, respectively. CC is applied to examine how well trends in the predicted values track trends in actual values. CC approaching 100% is an indication that the predicted values from a forecast model 'fit' well the real data. R^2 refers to the fraction of variance explained by a model. Thus R^2 values close to 100% are desirable.

B. Results and discussions

The time series representing the number of TB cases in the U.S. shows clear seasonal patterns and slight downward trend during the investigated years from 1993 to 2010 (see Fig. 5a-5b).

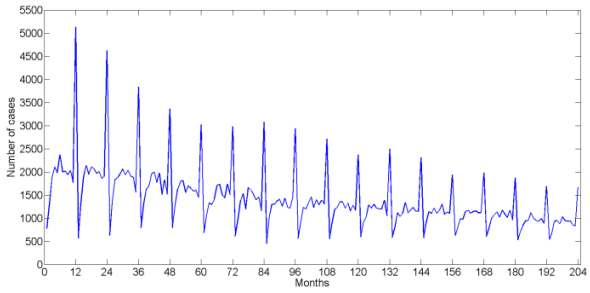

Fig. 5a. Monthly number of cases of TB in the U.S. (1993-2010)

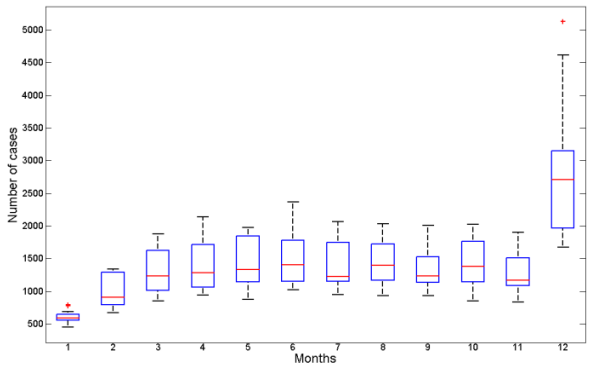

Fig. 5b. Box plot of the TB time series presented in Fig. 5a

As shown in the box plot Fig. 5b, the number of TB infectious cases in U.S. peaks in months of December and dramatically decline to the lowest in months of January. During springs and summers and early autumns, TB cases are relatively stable at the moderate level. This particular characteristic of U.S. TB is rather different from TB variation behaviors in other countries. For example in Japan during 1998 and 2000-2003, the number of TB cases was found lowest from November to January and highest around spring to summer seasons in a research of Nagayama and Ohmori [28]. In a review conducted by Fares [29], the author also found that TB seasonally peaks during the spring and summer seasons in most of the examined countries.

Alternatively, the seasonal patterns are also clear in the New York monthly number of varicella cases during 1931-1972 (Fig. 6a). On the other hand, it is apparent from Fig. 6b that the time series of number of varicella cases in the New York City is seasonal. It is at its lowest in August, September and October. It is then increases until April and then begins declining until August.

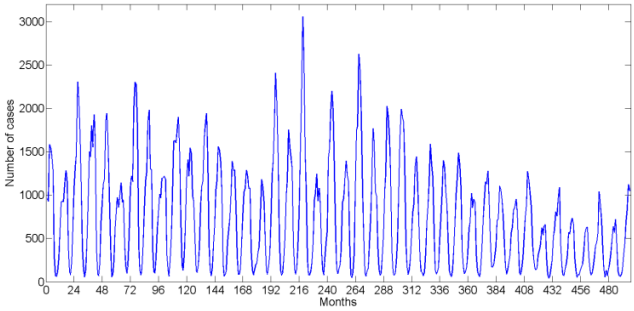

Fig. 6a. New York monthly number of varicella cases (1931-1972)

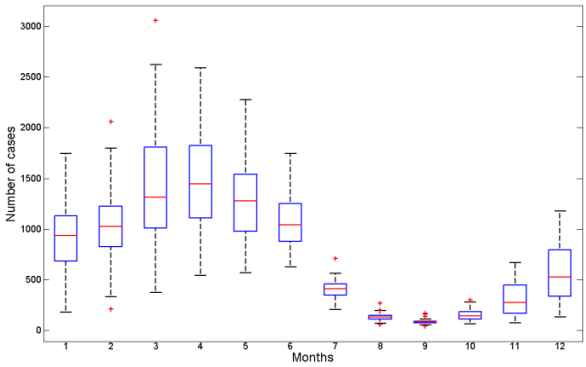

Fig. 6b. Box plot of the varicella time series presented in Fig. 6a

The following presents the competency of the proposed NFGF in modeling such seasonal or complex epidemiological time series. The out-of-sample testing results for the U.S. TB and New York varicella datasets are assembled in Table I and Table II respectively. Based on the forward stepwise regression, three inputs selected for modeling the U.S. TB dataset are lag 4th, lag 11th and the coded month values. Inputs selected in modeling the New York varicella dataset are three lags 5th, 14th and 15th.

TABLE I. U.S. TB FORECASTING RESULTS

Models	RMSE	CC (%)	R^2 (%)
ANFIS	0.9236	09.77	-975.29
NN(10-10)	0.1926	80.24	46.10
NN(30-30)	0.2941	54.24	-16.90
NN(50-50)	0.4807	40.48	-221.47
NFGF 15→13$^{(*)}$	0.1568	90.90	69.00

$^{(*)}$The number of fuzzy rules before and after running GA optimizer

TABLE II. NEW YORK VARICELLA FORECASTING RESULTS

Models	RMSE	CC (%)	R^2 (%)
ANFIS	0.6740	83.53	42.97
NN(10-10)	0.3753	92.06	82.00
NN(30-30)	0.5436	86.05	59.32
NN(50-50)	0.6175	81.19	49.46
NFGF 39→22	0.3228	94.57	86.92

The more accurate forecasting results are obtained with the NFGF models over all analyzed datasets compared to ANFIS and FFNN models. For example, in the U.S. TB dataset, the RMSE on the NFGF is only at 0.1568, which is lower than that of the ANFIS model at 0.9236. The best NN model is where 10 nodes are used in the hidden layers, i.e.

NN(10-10) with RMSE at 0.1926. Similar outcomes are found in the New York varicella dataset where NFGF is ranked first with the lowest RMSE at 0.3228 whilst that of ANFIS and NN(10-10) are respectively 0.6740 and 0.3753. More neurons in the NN hidden nodes do not increase the forecasting accuracy evidenced from results of all datasets by the dominance of the NN(10-10) over the NN(30-30) and NN(50-50) models. This is understandable because more parameters in the model would probably lead to overfiting and lower the generalization capability.

Both CC and R^2 measures provide the same ranking results to the RMSE criterion. NFGF model obtains highest CC and R^2 values, which are congruent with the lowest RMSE values among investigated models. In the U.S. TB dataset, the NFGF forecasts have R^2 reaching 69% whilst the CC value achieves the accuracy of 90.9%. These values are much higher than those of the ANFIS and NN models. NN(10-10) obtains CC and R^2 at 80.24% and 46.1% respectively whilst ANFIS is severely ruined when CC is just at 9.77% and R^2 falls to a negative value. In the New York varicella dataset, NFGF also reaches the highest CC and R^2 at 94.57% and 86.92% correspondingly. Both ANFIS and NN(10-10) models have CC and R^2 lower (see Table II).

Generally, NFGF achieves forecasting values closer to the actual values than other models. This fact demonstrates the advantage of the recurrent nodes of the proposed NFGF in handling the cyclic temporal data. The GA augments this advantage through optimizing the fuzzy rule structure and thus reduces supervised training cost for the NFGF. In the U.S. TB dataset experiment, the initial number of fuzzy rules of NFGF is 15. GA optimizes this initial NFGF rule structure down to 13 significant rules. Likewise, the NFGF rule structure is initialized with 39 rules and it is optimized by GA to realize just 22 necessary rules for supervised training and forecasting afterwards.

The ANFIS model basically does not possess the recurrent nodes so that the ability to handle cyclic patterns in time series is limited. The results from running all datasets confirm this fact as ANFIS model achieves the worst performance ranking compared to NN and NFGF.

Fig 8a. represents the NFGF and NN(10-10) forecasts versus actual values in the TB dataset. It is obvious that NFGF follows the actual series closer than NN(10-10). Fig. 8b details more the comparisons between these models.

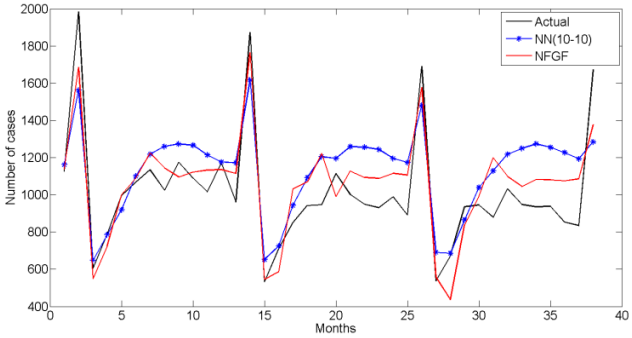

Fig 8a. TB forecasts comparisons between NFGF and NN(10-10)

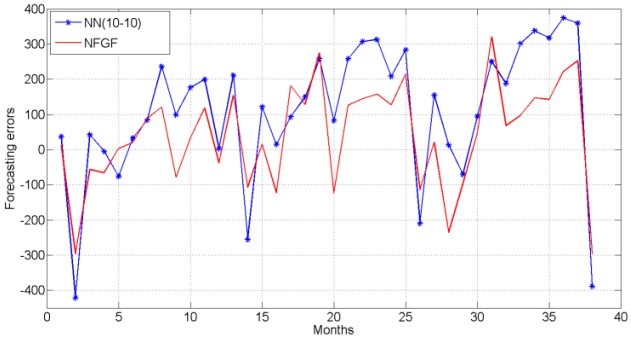

Fig 8b. TB forecasting errors between NFGF and NN(10-10)

Graphical comparisons between NFGF and NN(10-10) forecasts in the New York varicella dataset are presented in Fig. 9a.

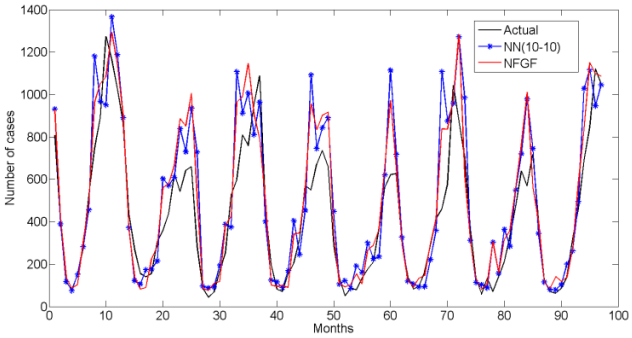

Fig 9a. N.Y. varicella forecasts comparisons between NFGF and NN(10-10)

Forecasting errors of the NFGF vary closer to zero compared to those of the NN(10-10) in the New York varicella dataset (see Fig. 9b).

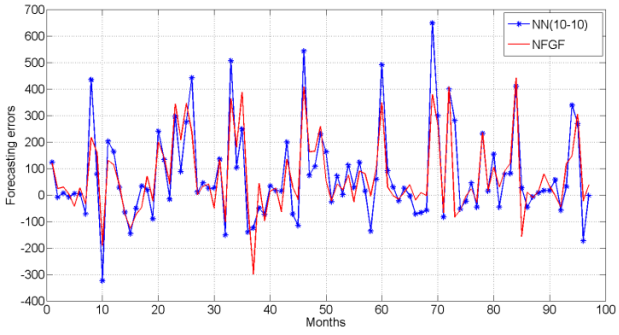

Fig 9b. N.Y. varicella forecasting errors between NFGF and NN(10-10)

V. Conclusions

This paper first transparently transforms SAM to an adaptive neural network model. The successfulness of the transformation opens the possibility for SAM to be extended to include various advantages of NN that are not available in FS. Second, we demonstrate the advantage of the proposed SAM-equivalent NN structure by incorporating recurrent nodes into it in order to formulate RSAM. The learning laws for RSAM parameters also have been derived in detail. The proposed RSAM is the more general form of SAM that is capable of handling temporal relations. We later integrate GA to optimize fuzzy rules of RSAM to formulate the

NFGF aiming at superior approximation ability and reducing computational costs of the RSAM supervised training.

Epidemiological time series data are modeled efficiently by the proposed NFGF. The key success of the NFGF comes from the recurrent nodes and the GA optimizer applied to the fuzzy rule structure. While addition of recurrent nodes increases number of modifiable parameters then GA mitigate this computational burden by reducing fuzzy rules. The harmonious combination of GA and the recurrent node extension brings an optimal fusion of soft computing techniques that is able to cope with non-linear and temporal relations in epidemiological time series. Results of experiments in time series modeling shows superior performance and flexibility of NFGF compared to ANFIS and FFNN.

As epidemiological evidence from time-series studies has played a critical role in public health authorities' decision making, accurate modeling epidemiological data offered by the proposed NFGF helps to estimate the burden of infectious diseases precisely, assess health risks of the communities and further bring great benefits in protecting the public's health. Based on accurate forecasts, public health authorities can evaluate disease trends better, assess the efficiency of control and prevention measures more accurately, allocate resources appropriately and thus reduce financial and human costs, and further devise prevention strategies and establish public health policies effectively.

The objective of this study is just to investigate the univariate time series models for forecasting because we have the limited data available. The utilization of other variables (presumably data are available), e.g. air pollution, temperature, humidity, sunlight, rainfall, populations, people's incomes probably much more useful. As cohort studies are also important in epidemiology, the merging of time-series and cohort studies is an interesting future research direction.

REFERENCES

[1] F. S. Dawood, A.D. Iuliano, C. Reed, *et al.* "Estimated global mortality associated with the first 12 months of 2009 pandemic influenza A H1N1 virus circulation: A modelling study," *Lancet Infectious Diseases*, vol. 12, no. 9, pp. 687-695, 2012.

[2] S. S. Shrestha, D. L. Swerdlow, R. H. Borse, *et al.* "Estimating the burden of 2009 pandemic influenza A (H1N1) in the United States (April 2009-April 2010)," *Clinical Infectious Diseases*, vol. 52, no. Suppl. 1, pp. S75-S82, 2011.

[3] C. Viboud, M. Miller, D. Olson, *et al.* "Preliminary estimates of mortality and years of life lost associated with the 2009 A/H1N1 pandemic in the US and comparison with past influenza seasons," *PLoS Currents Influenza*, vol. 2, doi: 10.1371/currents.RRN1153, 2010.

[4] D. Butler, "Portrait of a year-old pandemic," *Nature*, vol. 464, pp. 1112-1113, 2010.

[5] L. Watier and S. Richardson, "Modeling of an epidemiological time series by a threshold autoregressive model," *The Statistician*, vol. 44, no. 3, pp. 353-364, 1995.

[6] K. Koelle and M. Pascual, "Disentangling extrinsic from intrinsic factors in disease dynamics: A nonlinear time series approach with an application to cholera," *The American Naturalist*, vol. 163, no. 6, pp. 901-913, 2004.

[7] B. Cazelles, M. Chavez, G. C. de Magny, J. F. Guégan, and S. Hales, "Time-dependent spectral analysis of epidemiological time-series with wavelets," *Journal of the Royal Society Interface*, vol. 4, no. 15, pp. 625-636, 2007.

[8] N. K. Kasabov, J. Kim, M. J. Watts, and A. R. Gray, "FuNN/2 – A fuzzy neural network architecture for adaptive learning and knowledge acquisition," *Information Sciences*, vol. 101, no. 3-4, pp. 155-175, Oct. 1997.

[9] L. C. Jain and N. M. Martin, *Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications*, CRC Press LLC, 1998.

[10] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. on Systems, Man and Cybernetics,* vol. 23, no. 3, pp. 665-685, May/June 1993.

[11] R. V. Borges, A. d'Avila Garcez, and L. C. Lamb, "Learning and representing temporal knowledge in recurrent networks," *IEEE Trans. on Neural Networks*, vol. 22, no. 12, pp. 2409-2421, Dec. 2011.

[12] L. R. Medsker and L. C. Jain, *Recurrent Neural Networks: Design and Applications*, CRC Press, 2001.

[13] J. S. R. Jang, and C. T. Sun, "Neuro-fuzzy modeling and control," *Proc. of the IEEE*, vol. 83, no. 3, pp. 378-406, Mar. 1995.

[14] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs: Prentice-Hall, 1991.

[15] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. on Computers*, vol. 43, no. 11, pp. 1329-1333, Nov. 1994.

[16] B. Kosko, *Fuzzy Engineering*, Prentice Hall, 1996.

[17] S. Mitaim and B. Kosko, "What is the best shape for a fuzzy set in function approximation?," *Proc. 5th IEEE Int. Conf. Fuzzy Systems (FUZZ-96)*, vol. 2, pp.1237-1243, 1996.

[18] S. Mitaim, and B. Kosko, "The shape of fuzzy sets in adaptive function approximation," *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 4, pp. 637-656, Aug. 2001.

[19] C. H. Lee and C. C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 4, pp. 349-366, Aug. 2000.

[20] W. Sun and Y. Wang, "A recurrent fuzzy neural network based adaptive control and its application on robotic tracking control," *Neural Information Processing - Letters and Reviews*, vol. 5, no. 1, pp. 19-26, Oct. 2004.

[21] J. Dickerson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 26, no. 4, pp. 542-560, Aug. 1996.

[22] T.T. Nguyen, L. Gordon-Brown, and J. Peterson, "The optimal rule structure for fuzzy systems in function approximation by hybrid approach in learning process," *Proc. of Int. Conf. on Computational Intelligence for Modeling Control and Automation*, pp. 1211-1216, Vienna, Dec. 2008.

[23] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[24] D. E. Goldberg, *Genetic algorithms in Search, Optimization, and* Machine *Learning*, Addison Wesley, Massachusetts, USA, 1989.

[25] C. R. Reeves and J. E. Rowe, *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*, Kluwer Academic Publishers, 2002.

[26] R. J. Hyndman, *Time Series Data Library*, http://data.is/TSDLdemo. Accessed on 1 Jan 2013.

[27] S. D. Balkin and J. K. Ord, "Automatic neural network modeling for univariate time series," *International Journal of Forecasting*, vol. 16, no. 4, pp. 509-515, 2000.

[28] N. Nagayama and M. Ohmori, "Seasonality in various forms of tuberculosis," *International Journal of Tuberculosis and Lung Disease*, vol. 10, no. 10, pp. 1117-1122, 2006.

[29] A. Fares, "Seasonality of tuberculosis," *Journal of Global Infectious Diseases*, vol. 3, no. 1, pp. 46-55, 2011.