A novel approach model applying SA-PSO algorithms for rainfall prediction in Vietnamese Mekong Delta

Tung Kieu
Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
Email: kvttung@gmail.com

Duong Tran Anh
Institute of Hydraulic and Water Resources Engineering, Technical University of Munich, Munich, Germany
Email: tran.duong@tum.de

Tram Nguyen
Faculty of Information Technology, University of Agriculture and Forest, Ho Chi Minh City, Vietnam
Email: phuongtram.itnl@gmail.com

***Abstract*—Time series (TS) forecasting is widely applied and plays important roles in various real-life implications such as finances, water resources, and environmental studies. Among time series forecasting models, Seasonal Autoregressive Moving Average (SARMA) model was extended from Autoregressive Moving Average (ARMA) model to improve the accuracy of predictions, in particular in trended and seasonal behaviors. However, an S-ARMA model contains several limitations such as inaccuracy and noise sensitivity. Furthermore, bioinspired algorithms consisting of Simulated Annealing algorithm (SA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are progressive techniques to obtain an optimum state of a nonlinear problem. This paper overcomes limitations of traditional S-ARMA model by the combination of two state-of-the-art computing algorithms which are SA and PSO to learn and discover parameters for the S-ARMA model. Our experiments have shown remarkable encouraging results compared with conventional forecasting methods including Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing (ES) and Meta-Genetic Algorithm (MetaGA) on two trended and seasonal time series datasets, which are precipitation datasets of Mekong River Delta.**

***Keywords: Time series forecasting, simulated annealing, particle swarm optimization*.**

I. INTRODUCTION

In the domain of environmental studies, predicted values like rainfalls and discharges are particularly critical in water resources management, which predicted results contribute significantly to response and mitigation of natural disaster management. Prediction of precipitation is a crucial task in hydrological studies [2] because precipitation is the fundamental input for further simulations. A precipitation data contains seasonal and trended features. Furthermore, precipitation is a complex phenomenon, which is affected by many factors such as cloud cover, humidity, geography and local hydrological characteristics. Rainfall forecasting is still a challenge in term of historical analysis in hydrological and environmental studies. Mekong River is the longest river in South East Asia and plays a critical role in Vietnamese economy. Prediction of rainfall of Mekong river is the fundamental input for further analysis and assessment of water resource in Vietnamese Mekong Delta [17], [16].

One of the most important data structure to represent rainfall in environmental sciences is TS [5]. Many researchers have been attempted to simulate the future precipitation based on different TS methods [2] such as ARMA [4] and ARIMA [7]. ARMA and ARIMA are ancient techniques. However, despite its age, ARMA and ARIMA are still two of the most popular models for predicting TS. Seasonal ARMA (S-ARMA) is an extension of ARMA model. It is a model which can be used for representing seasonal and trended data. However, S-ARMA has several limitations such as inaccuracy and noise sensitivity. Computing suitable coefficients for S-ARMA model is the most significant problem. S-ARMA will outperform if we find out appropriate coefficients for the model.

Finding appropriate coefficients can be represented as an optimization problem. In optimization problems, meta-heuristic algorithms play major roles. In metaheuristic techniques, bioinspired algorithms are the most important approach. Furthermore, bioinspired algorithms consisting of Simulated Annealing algorithm (SA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are progressive techniques to obtain an optimum state of a nonlinear problem. SA are the alternative techniques, which are based on the principles of liquids freezing, or metals recrystallize in the process of annealing [13]. However, that has rarely applied in the statistical analysis. Within our knowledge, we have not experienced the application of SA in time series predictions although that is widely employed in other domains such as graph coloring and timetabling. GA are optimization techniques, which are based on natural evolution process [10]. Motivated by its abilities to find globally optimal solutions over feature space. Another popular meta-heuristic algorithm is PSO. It solves 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 particle’s position and velocity.

Furthermore, researchers combine several methods to achieve better results. GA often handles to learn parameters for ARMA model [14], parameters or configuration weights for ANN [18]. Cortez et al. [8] proposed a Meta-Genetic (Meta-GA) algorithm to learn models and parameters for ARMA models with two layers algorithm. The Meta-layer (top layer) utilizes genetic algorithm with binary value encoding to represent ARMA models, and the bottom layer applies genetic algorithm with real value encoding to estimate ARMA model parameters with each model acquired from meta-layer. Valenzuela et al. [20] developed a hybrid ARMA-ANN model to improve the accuracy of model skills. An alternative approach is a combination of ANN and PSO. Daniel Alba-Cullar et al. [1] proposed the algorithm to forecast values for univariate time series datasets, based on ANN ensembles. Each ensemble is trained with the PSO algorithm. The results showed that PSO was a good algorithm to train the TS datasets, and produced robust predictions. Unfortunately, PSO spent much time to reach the optimum state. It is, therefore, necessary to develop a new method, which is more accurate and more applicable to obtain better rainfall predictions with a short runtime.

Because of preeminent features, in this study, we combine SA and PSO to make use of the simplicity of SA and the accuracy of PSO to forecast the rainfall in Vietnamese Mekong Delta. This algorithm can be known as SA-PSO. SA-PSO will learn from historical data of rainfall of Mekong river delta and find out suitable coefficients for a TS model. More precisely, S-ARMA model will be chosen. Finally, this S-ARMA model is used for forecasting the rainfall of Mekong River Delta at two locations including Chau Doc and Can Tho. The experiments of observed rainfall at two stations in Vietnamese Mekong River Delta are verified. The experimental results show not only a remarkable precision but also the preeminence of our proposed algorithm.

The structure of this paper is organized as follows. Section 1 introduces the problem of TS forecasting and overviews several algorithms. Section 2 presents some works related to TS, SA, and PSO. Section 3 formalizes the problem in a mathematical representation. The proposed model will also be described in this section. Section 4 presents the results of extensive experiments conducted with two TS datasets to evaluate the accuracy of the proposed algorithms and compare with other conventional methods. Section 5 presents the conclusions, the practical applications of this approach in TS forecasting, and suggestions for the further development of SA-PSO.

II. METHODOLOGY

A. Seasonal ARMA model

A Time Series (TS) [4] is a sequence of data points, measured typically at successive times and spaced at (often a uniform) time intervals. The time series forecasting models assume that past patterns will be valid in the future. The forecasting error e of a model is given by the difference between predicting value yb$_t$ and actual value y_t of time series.

$$e = y_t - \widehat{y_t} \tag{1}$$

ARMA model is one of the most common methods for time series forecasting and is the combination of the Autoregressive model (AR) and the Moving Average model (MA). We denote $ARMA(p,q)$ as a linear combination of p past values, q errors and white noise μ as show in the equation below.

$$y_t = \mu + \sum_{i=1}^{p} A_i x_{t-i} + \sum_{i=1}^{q} M_i e_{t-i} \tag{2}$$

An S-ARMA model is an extension of an ARMA model. The seasonal element of an ARMA model, denoted by $ARMA(P,Q)_h$, which is of the form

$$\Phi(B^h)X_t = \Theta(B^h)Z_t \tag{3}$$

where

$$\Phi(B^h) = 1 - \Phi_1 B^h - \Phi_2 B^{2h} - \ldots - \Phi_P B^{Ph} \tag{4}$$

and

$$\Theta(B^h) = 1 + \Theta_1 B^h + \Theta_2 B^{2h} + \ldots + \Theta_Q B^{Qh} \tag{5}$$

For example seasonal attribute $ARMA(1,1)_{12}$ can be written as

$$(1 - \Phi B^{12})X_t = (1 + \Theta B^{12})Z_t \tag{6}$$

or

$$X_t - \Phi X_{t-12} = Z_t + \Theta Z_{t-12} \tag{7}$$

When written as

$$Xt = \Phi X_{t-12} + Z_t + \Theta Z_{t-12} \tag{8}$$

and compare to $ARMA(1,1)$ which can be written as

$$X_t = \varphi X_{t-1} + Z_t + \theta Z_{t-1} \tag{9}$$

As can be seen that the S-ARMA presents the series regarding its past values at lag equal to the length of the period (here h=12), while the non-seasonal ARMA does it regarding its past values at lag 1. Seasonal ARMA incorporates the seasonality into the model.

When we combine seasonal and non-seasonal operators as a linear combination of p, q, h and white noise μ, we obtain a model

$$\Phi(B^h)\phi(B)X_t = \Theta(B^h)\theta(B)Z_t \tag{10}$$

which is called mixed Seasonal-ARMA or S-ARMA and it can be denoted by

$$ARMA(p,q) \times (P,Q)_h \tag{11}$$

or

$$y_t = \mu + \sum_{i=1}^{p} A_i x_{t-i} + \phi x_{t-h} + \sum_{i=1}^{q} M_i e_{t-i} + \theta e_{t-h} \tag{12}$$

B. SA-PSO stack approach for S-ARMA model

There are two meta-heuristic algorithms which are SA and PSO in this model. The stacked algorithm is applied for optimizing the parameters of S-ARMA model. This algorithm consists of two layers which top layer performs to find optimal models and then the bottom layer searches in real value searching space to obtain optimal parameters for that model. Basically, the top layer employs an array of bit values to represent S-ARMA model. Each value of array represents a possible coefficient. If the value is true, the corresponding coefficient exists; otherwise, it is not considered in the model. The bottom layer is an array of real values, which contains real-value parameters for each corresponding coefficient represented by top layer structure. If a coefficient does not exist in the top layer, the following value in the bottom layer is ’0.0’. The processing of stacked algorithm can be described as following. Firstly, SA algorithm creates an initial array of bit values to represent the shape of the model. Secondly, based on the initial array of bit values, PSO algorithm creates the initial array of real values to represent the coefficients of the model. PSO algorithm will perform some loops to update the coefficients and evaluate the quality of the model. After that, the SA algorithm performed the optimization of the initial model and based on it the PSO will run to find a better model. This process repeats many times, and definitely, the best model will be created. Finally, the stacking algorithm will show the best model for end-user.

The next figure describes the structure of S-ARMA parameters, which will be constructed by our stack algorithm.

A given training dataset comprised past values of time series. The algorithm will find the most appropriate model and values for each coefficient in this model. After that, we use this model to forecast future values. The following flowchart describes details of the algorithm.

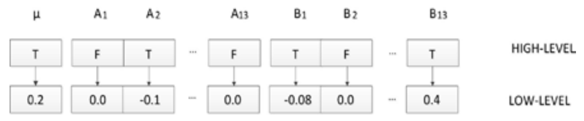

Fig. 1: Structure of the S-ARMA parameters

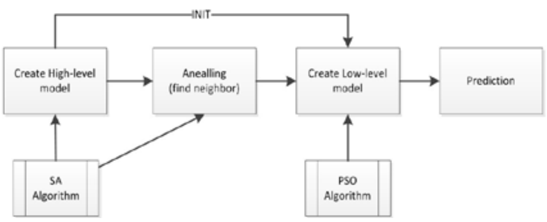

Fig. 2: Stack structure for our algorithm

1) Bottom layer algorithm: The function of the bottom layer is to find the best coefficients for the S-ARMA model. PSO algorithm is occupied in this layer to attain the optimal parameters for the model. PSO algorithm is a meta-heuristic without assumptions about the problem being optimized and can search considerable large spaces of potential solutions [12]. The canonical PSO algorithm is applied to compute the real-value parameters. Besides that, additional options are considered to extend the PSO algorithm. These options are based on previous researches [3] to improve canonical PSO algorithm.

Firstly, the initial particles are randomly assigned to a real value position and a real value velocity. The fitness function of each individual is measured by *RMSE* cost over the training dataset. Secondly, at each epoch, the particles will update their velocities and positions base on the other positions of particles and velocities of particles. The termination criterion can be many iterations performed, or a solution with adequate objective function value is found. Finally, the goal is to find the best solution in the search space, which would mean this solution is the global optimum.

Two of the most complex problem are maintaining the diversity of particles in PSO [3] and velocity control [3]. Particles which are a long distance from p^{best} and g^{best} will tend to produce very large velocity updates, and therefore, there will be large position oscillations from one iteration of the algorithm to the next. In order to overcome this problem, a number of methods can be applied to constrain the magnitude of the velocity vector such as *momentum weight* [19], *velocity clamping* [9] and *constriction coefficient* [6]. Besides that, to maintain the diversified of the particles, the problem is to avoid premature convergence of the population. The simplest solution is running PSO several times. This solution will reduce the rate that PSO algorithm is trapped in local optimum, but not assure the PSO algorithm will find out global optimum.

2) Top layer algorithm: SA is chosen because of the efficiency. The basic SA algorithm takes place for finding suitable S-ARMA predicting models. Firstly, the initial creates randomly an initial model *s* which is an array of bit values. Then, from this array, PSO from bottom layer will create the initial coefficients for S-ARMA model, PSO will run many epochs to create the best S-ARMA model which is based on the array of bit values. After PSO finishes creating and evaluating. After that, the SA heuristic considers some neighboring state s^0 of the current state *s* and probabilistically decides between moving the system to state s^0 or staying in state *s*. This step will repeatedly run to find the best S-ARMA model.

To evaluate the fitness of each found model, the cost function which we used *Bayesian Information Criterion* (*BIC*) [8], which adds a penalty to the model that is a function its complexity, as described below:

$$BIC = N \times \ln(\frac{SSE}{N}) + p \times \ln(N) \qquad (13)$$

Where N denotes the number of training examples and p is the number of model parameters. The initial model is generated randomly with boolean values. With each model, the neighbor model is generated via three mechanisms: perturbation, swap, and flip.

- Perturbation: each coefficient has a chance to switch from false and vice versa.
- Swap: swap value from two random coefficients
- Flip: switch value of any coefficients from true to false and vice versa

The utilization of perturbation mechanism allows us to jump into different search spaces to ensure the diversity of solutions and to increase the ability to escape local optimums. The taking account of swap and flip mechanism allows us to explore search spaces carefully. We named this technique variant neighborhood selection mechanism.

SA can apply several extensions such as Adaptive Simulated Annealing (ASA) [11]. The algorithm parameters such as temperature, neighborhood radius, step size, etc. are adjusted by the algorithm during its run, with the intention that end-users do not need to select precise starting values of these (SA performance can be very sensitive to the initial selection of the parameter values). Before the algorithm running, ASA will have a coarse resolution.

III. EXPERIMENTS

A. Datasets

We use two datasets to run the model using rainfall values from Jan 2001 to Dec 2010. Those are two rainfall datasets were collected by two weather stations in Chau Doc and Can Tho, Vietnamese Mekong Delta, Vietnam. Those TS are categorized into groups namely Seasonal + Trended.

Each TS dataset was divided into a training set with the first 90% values and a test set with the last 10%. We only use the training set for model selection and parameter optimization. The test set is used to compare the forecasting ability of our proposed algorithm with other forecasting methods. The following table shows the datasets which are used in the experiment.

TABLE I: Time series datasets

Name	Type	Description
Chau Doc	Seasonal & Trended	Rainfall of Chau Doc
Can Tho	Seasonal & Trended	Rainfall of Can Tho

The following graphics show the visualization of datasets which are used in experiment.

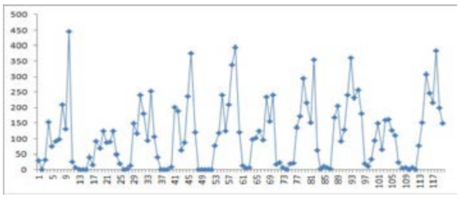

Fig. 3: Monthly Rainfall at Chau Doc

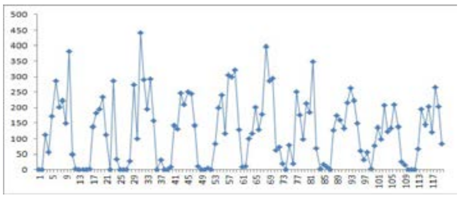

Fig. 4: Monthly Rainfall at Can Tho

B. Implementation

In this work, trended AR and MA orders (p and q) were set to 12, and seasonal AR and MA orders (h) were set to 13, these values which were considered sufficient to encompass seasonal and trended effects. So the size of array in top layer and bottom layer is 27 (1 for the constant μ, 12 for each trended AR and MA coefficients and 13th coefficient of AR and MA describes the seasonal value). The following formula illustrates the proposed model

$$y_t = \mu + \sum_{i=1}^{12} A_i x_{t-i} + \phi x_{t-13} + \sum_{i=1}^{12} B_i e_{t-i} + \theta e_{t-13} \tag{14}$$

The most difficult work in our experiments was to choose the proper input parameters for both SA and PSO. To do that, we tried many parameter sets, some of which were suggested by other authors. Because our results are compared with Meta-GA [8], used parameters must guarantee the equality between our algorithm and Meta-GA. Table II describes parameters used in our experiment.

TABLE II: Characteristics of algorithms

SA	Encoding	Boolean value
	Neighborhood selection	Perturbation (20%), swap, flip
	Cost function	*BIC*
	Initial solution	Random in {true,
	Number of phase (*maxphase*)	false} 5000
	Number of iteration (*nrep*)	5
	Starting temperature (t_0)	10000
	Temperature reduction coefficient (α)	0.98
PSO	Encoding	Real value
	Cost function	*RMSE*
	Number of particle	70
	Initialization	Random in [-1, 1]
	Velocity update	Normal, momentum weight, constriction coefficient
	Number of epoch	1000
	Number of iteration	500
	Self confidence factor	2
	C_1	2
	Swarm confidence factor C_2	

C. Results

Model skill is evaluated by statistical performance indices including Sum Square Error (*SSE*), Root Mean Square Error (*RMSE*) and Normalized Mean Square Error (*NMSE*) [8]. In this paper, we chose *RSME* to measure the accuracy of proposed method. *RSME* formula is described as follows:

$$RMSE = \sum_{i=1}^{l} \sqrt{\frac{e_i^2}{l}} \qquad (15)$$

Where *l* is the number of predicted values and *x* is the mean of time series.

Experimental results show that SA-PSO has consistently outperformed the other algorithms such as ES, SARIMA, ANN and SA-GA [15] in the accuracy. The following table shows results of experimental. The values in table III describe the *RMSE* value

TABLE III: Experimental results

Time Series/ Algorithms	ES	SARIMA	ANN	Meta-GA	SA-GA	SA-PSO
Chau Doc	0.310	0.152	0.228	0.152	0.132	0.121
Can Tho	0.276	0.151	0.185	0.144	0.128	0.094

Table III shows SA-PSO algorithm can produce a promising result with rainfall datasets. The *RSME* value of Chau Doc data for SA-PSO is better than 61 % and 47 % compared to ES and ANN methods, respectively. In addition, SA-PSO also slightly improved
8 % and 5 % compared to SA-GA for Chau Doc and Can Tho datasets, respectively.

Figure 5 and 6 show forecasting values for the last 10% of Chau Doc and Can Tho rainfall time series, considering the average of five runs over the optimized S-ARMA model.

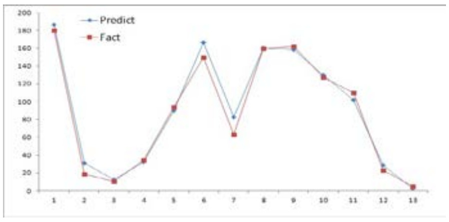

Fig. 5: Predicted values of Chau Doc time series

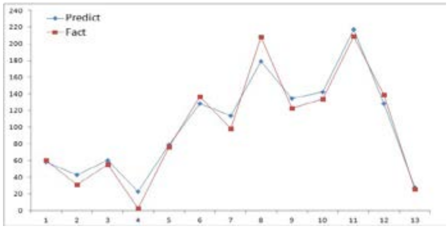

Fig. 6: Predicted values of Can Tho time series

Figure 7 and 8 show the relationship between factual values and forecasting values for the last 10% of Chau Doc and Can Tho rainfall time series, considering the average of five runs over the optimized S-ARMA model. Both figures show the good fitness between factual values and forecasting values.

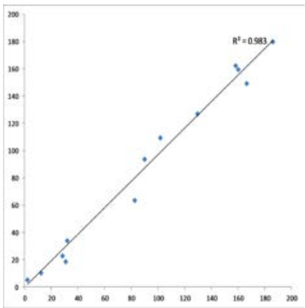

Fig. 7: Scatter plot for rainfall at Chau Doc (testing data)

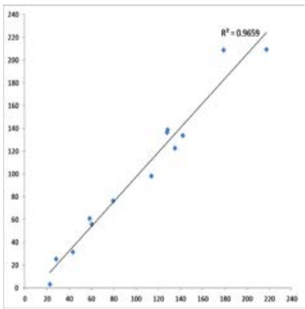

Fig. 8: Scatter plot for rainfall at Can Tho (testing data)

The experimental results of S-ARMA showed the accuracy of proposed algorithm in TS predicting particularly rainfall dataset. Recently, issues of climate changing pose new challenges for computer science and environmental science, especially in forecasting. Typically, the time series of rainfall in Can Tho and Chau Doc comprised nonlinear elements. These nonlinear elements lead to inaccuracies when using the forecasting method based on classical statistics such as SARIMA, or ES. AI algorithms such as SA-GA and SA-PSO now demonstrate their superiority when increased accuracy than doubled compared with the old algorithm.

IV. CONCLUSIONS

Natural computing algorithms are state-of-the-art techniques. These algorithms play a major role in numerous field studies such as optimization, machine learning, and environmental science. Applications of natural computing algorithms in TS forecasting are more successful with supporting new statistical methods. This paper attempted to combine two natural-based algorithms to find out the best parameter for S-ARMA model. This approach is the stack of two algorithms, the first algorithm is SA in the top layer and the second one is PSO in the bottom layer. Experimental results showed promising outcomes for the forecast time series particular in the seasonal and trended attributes. Our finding also showed the improvement of predicted results compared to conventional methods in seasonal and trended feature time series data such as precipitation or discharge. Through this work, the utilization of SA and PSO has confirmed that this model was capable of handling the seasonal and trended statistical data.

Our works in the future will cope with the implication of TS analysis techniques in neighborhood selection mechanism to select appropriate models in top layer and bottom layer algorithms, as well as the employment of this model to achieve better forecast outcomes in different field studies including the price of fuel, death rates which have seasonal and trended characteristics.

ACKNOWLEDGMENT

We acknowledges the financial support from the CUOMO Foundation and the Intergovernmental Panel on Climate Change (IPCC). We special thank Dr. Mai Thai Son for guiding and giving technical advises; Dr. Dang Nguyen Mai and Msc. Ta Huu Chinh - National meteorological center for providing rainfall data in this study.

REFERENCES

[1] D. Alba-Cuellar, A. E. Muoz Zavala, A. H. Aguirre, E. E. Ponce de Leon Senti, and E. D´ ´ıaz-D´ıaz, “Time series forecasting with pso-optimized neural networks,” in Proceedings of 13th Mexican International Conference on Artificial Intelligence, Tuxtla Gutierrez, Mexico, November 16-22, 2014, 2014, pp. 102–111.

[2] S. Araghinejad, Data-driven modeling: using MATLAB R in water resources and environmental engineering. Springer Science & Business Media, 2013, vol. 67.

[3] A. Brabazon, M. O’Neill, and S. an McGarraghy,´ Natural Computing Algorithms, ser. Natural Computing Series. Springer, 2015.

[4] P. J. Brockwell and R. A. m. Davis, Introduction to time series and forecasting, ser. Springer texts in statistics. New York, Berlin, Heidelberg: Springer, 2002.

[5] C. Chatfield, The analysis of time series: an introduction. CRC press, 2016.

[6] M. Clerc, “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization,” in Proceedings of the 1999 IEEE Congress on Evolutionary Computation, Washington, DC, USA, July 6-9, 1999, vol. 3, 1999, pp. 1951–1957.

[7] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, “Arima models to predict next-day electricity prices,” IEEE transactions on power systems, vol. 18, no. 3, pp. 1014–1020, 2003.

[8] P. Cortez, M. Rocha, and J. Neves, “Genetic and evolutionary algorithms for time series forecasting,” in Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Budapest, Hungary, June 4-7, 2001, L. Monostori, J. Vancza,´ and M. Ali, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 393–402.

[9] A. P. Engelbrecht, “Particle swarm optimization: Velocity initialization,” in Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, June 10-15, 2012, 2012, pp. 1–8.

[10] J. H. Holland, Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975, second edition, 1992.

[11] A. L. Ingber and L. Ingber, “Very fast simulated re-annealing,” 1989.

[12] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in Proceedings of the 1995 IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” SCIENCE, vol. 220, no. 4598, pp. 671–680, 1983.

[14] C. Kishtawal, S. Basu, F. Patadia, and P. Thapliyal, “Forecasting summer rainfall over india using genetic algorithm,” Geophysical Research Letters, vol. 30, no. 23, 2003.

[15] N. L. A. T. Mai Thai Son, “A new approach to time series forecasting using simulated annealing algorithm,” in Proceedings of the 2010 International Conference on Advanced Computing and Applications, Ho Chi Minh City, Vietnam, March 3-5, 2010, 2010.

[16] H. N. Nguyen, K. T. Vu, and X. N. Nguyen, “Flooding in mekong river delta, viet nam,” Human development report, vol. 2008, p. 23, 2007.

[17] T. Nguyen, Q. N. Huu, and M. J. Li, “Forecasting time series water levels on mekong river using machine learning models,” in Proceedings of 7th International Conference on Knowledge and Systems Engineering, Ho Chi Minh City, Vietnam, October 8-10, 2015, 2015, pp. 292–297.

[18] B. Oancea and S. C. Ciucu, “Time series forecasting using neural networks,” CoRR, vol. abs/1401.1333, 2014.

[19] G. Pereira, “Particle swarm optimization,” INESCID and Institute Superior Techno, Porto Salvo, Portugal, 2011.

[20] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L. J. Herrera, A. Guillen, L. Marquez, and M. Pasadas, “Hybridization´ of intelligent techniques and arima models for time series prediction,” Fuzzy Sets and Systems, vol. 159, no. 7, pp. 821–845, 2008.