ICIC Express Letters
Part B: Applications
Volume **11**, Number **7**, July **2020**

pp. **631–639**

EVALUATION PERFORMANCE OF SVR GENETIC ALGORITHM AND HYBRID PSO IN RAINFALL FORECASTING

Rezzy Eko Caraka[1,3], Rung Ching Chen[1,*], Toni Toharudin[2]
Muhammad Tahmid[4], Bens Pardamean[3,5] and Richard Mahendra Putra[4]

[1]College of Informatics
Chaoyang University of Technology
Jifong East Road, Wufong Dist., Taichung City 41349, Taiwan
s10714905@cyut.edu.tw; *Corresponding author: crching@cyut.edu.tw

[2]Department of Statistics
Padjadjaran University
Bandung, West Java 45363, Indonesia

[3]Bioinformatics and Data Science Research Center
[5]Department of Computer Science, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia

[4]Meteorological, Climatological, and Geophysical Agency (BMKG)
Jl. Angkasa 1 No. 2 Kemayoran, Jakarta 10720, Indonesia

Received April 2019; accepted May 2020

Abstract. *Climate is an essential natural factor which is dynamic and challenging to predict. The accurate climate prediction is needed. In this paper, we use support vector regression (SVR) with different kernels such as polynomial, sigmoid and RBF. At the same time, we employ genetic algorithm, particle swarm optimization, and hybrid particle swarm optimization. SVR-GA are population-based algorithms that allow for optimization of problems with the search space that is very broad and complex. This property too allows genetic algorithms to jump out of the local area optimum. In contrast with SVR-PSO and SVR-HPSO they do not have the genetic operation. In PSO only use internal velocity and have the memory which is essential to the algorithm. In this paper we compare SVR-PSO, SVR-HPSO and SVR-GA by comparing the input from the correlation and ARIMA in rainfall data. It was found that the input using correlation provides better accuracy than ARIMA.*
Keywords: SVR, GA, HPSO, PSO

1. **Introduction.** Climate is a natural phenomenon that is very important and influential for human life. Knowledge management of weather patterns and climate, especially rainfall, is needed in many sectors such as agriculture, plantations, and transportation. In the agriculture and plantation sectors, information that can predict the size of the amount of monthly rainfall in each region, will be beneficial to be able to determine the right cropping patterns and varieties of plants to produce good product. This is because rainfall has a direct effect on water availability. Therefore, an accurate, fast and site-specific information forecast is needed to predict future rainfall to minimize the impact of losses. Recently, models based on combining concepts have been paid more attention in climatology forecasting. Depending on different combination methods, combining models can be categorized into ensemble models and modular (or hybrid) models. Those machine learning methods commonly applied for rainfall forecasting include artificial neural

DOI: 10.24507/icicelb.11.07.631

networks (ANN). [1] compared ANN, with singular spectrum analysis (SSA) and multi-response support vector regression (MSVR). Comparison results indicated that modular models (referred to as ANN-SVR for daily rainfall simulations and MSVR for monthly rainfall simulations) outperformed other models. However, [2] developed vector autoregressive (VAR) and generalized space time autoregressive (GSTAR) as feature selection in SVR. The main concept of SVR is to maximize the margin around the hyper plane and to obtain data points that become the support vectors. In this research, we apply SVR-PSO, SVR-HPSO and SVR-GA to modeling the rainfall. We used the best input SVR based on ARIMA and correlation rainfall in each year.

2. **Methods.** SVR is an extended model of support vector machine (SVM) [3] for prediction and regression cases [4]. SVM applies the concept of a ε-incentive loss function. SVR has reliable performance in predicting time-series data [5]. If the value of $\varepsilon = 0$, a perfect regression is obtained [6]. The concept of SVR is based on risk minimization [7], which is estimating a function by minimizing the upper limit of the generalization error, so that SVR can overcome overfitting:

$$f(x) = w \cdot x + b$$

In the case of nonlinearity, nonlinear mapping: $R^1 \to F$, where F is a feature space of ϕ which is introduced to explain the complexity of nonlinear regression problems at R^1 for a simple linear regression problem at F. The regression function after transformation becomes as follows:

$$f(x) = w \cdot \phi(x) + b$$

where w is a weighting vector, $\phi(x)$ is a function that maps x in a dimension and b is a bias. To evaluate how well the regression function is, the function ε-incentive loss is used as follows.

$$L_\varepsilon(y, f(x)) = \begin{cases} 0, & \text{for } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon, & \text{otherwise} \end{cases}$$

The function of ε-incentive loss is used to measure empirical risk the target difference with estimation results [8]. Therefore, the parameter ε must be set to minimize empirical risk by using the slack variable ξ, ξ^* which describes the deviation from training data outside the ε-incentive zone. Besides minimizing empirical errors with ε-incentive, it must also minimize the Euclidean norm of linear $||w||$ which is related to the generalization ability of the SVR model trained. The regression problem can be expressed as the following quadratic optimization problem:

$$L(w, \xi) = \frac{1}{2}||w||^2 + C\left[\sum_{i=1}^{n}(\xi_{2i} + \xi'_{2i})\right], \quad C > 0$$

$$\text{subject to} \begin{cases} y_i - w * \phi(x_i) - b \le \varepsilon + \xi_i \\ w * \phi(x_i) + b - y_i \le \varepsilon + \xi_i^* \\ \phi_i, \xi_i^* \ge 0 \end{cases}$$

where C states the penalty coefficient which determines the trade-off between the incision and the generalization error in which the C value needs to be regulated. However, the value of the optimal parameter C is in the range 1-1000. To solve the quadratic optimization problem in equation we can use dual Lagrangian:

$$f(x_i) = (w\phi(x_i) + b) = \sum \alpha_i K(x_i, x_j) + bn_j = 1$$

$$f(x_i) = w\phi(x_i) + b$$

$$f(x_i) = \sum_{j=1}^{n} \alpha_i K(x_i, x_j) + b$$

where $K(x_i, y_i)$ is a kernel function. SVR uses kernel functions to transform non-linear inputs into higher-dimensional feature spaces [9]. Generally, problems in the real world are rarely linear separable [10]. The kernel function can solve this separable non-linear case [11]. This paper implements a merger of two optimization methods to find optimal kernel function parameters, such as genetic algorithm (GA) [12], particle swarm optimization (PSO) [13] and hybrid PSO [14]. Based on Figure 1 the population of chromosomes is generated randomly and allows it to multiply according to the law of evolution in hopes of producing a prime individual chromosome.

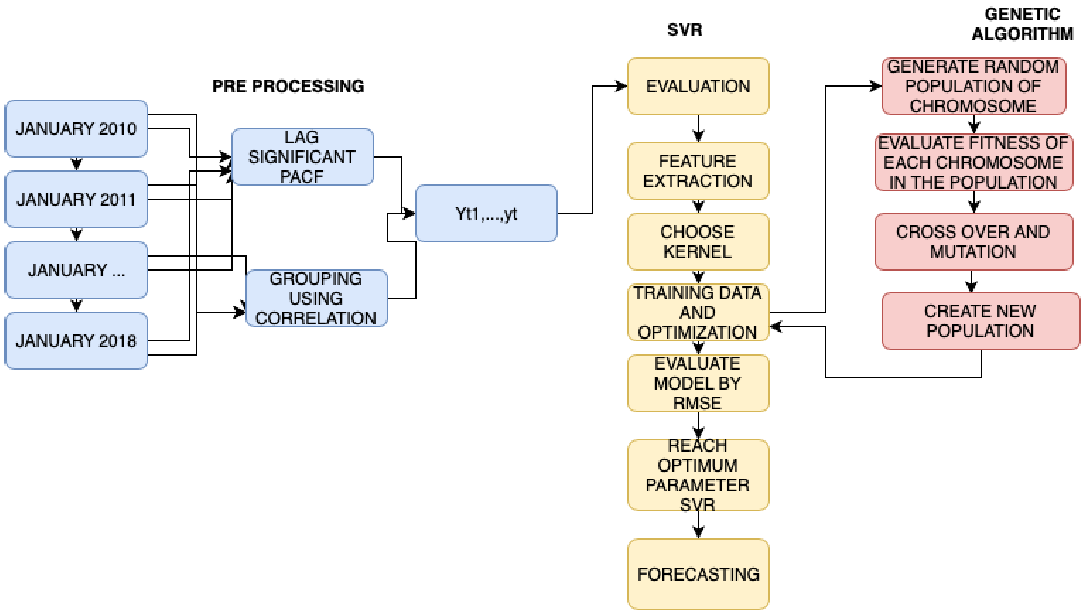

FIGURE 1. SVR-GA model

To optimize the SVR (C and γ) kernel parameters simultaneously [15], each chromosome is defined as two parts, first, the C gene and γ gene. C consists of $C_1, C_2, \ldots, C_n$ and γ consists of $\gamma_1, \gamma_2, \ldots, \gamma_n$. Second, the binary coding is used to represent chromosomes. Figure 2 illustrates the number of genes for each parameter is determined by the range of values given for that parameter. The gene contains parameter values in the form of numbers 0 or 1 which will be processed through crossover and mutation.

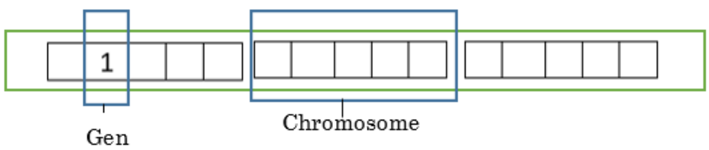

FIGURE 2. The chromosome of GA

This parameter value will be decoded as input in the SVR process. The assessment criteria used in this study are the accuracy of the SVR model that is determined by the value of root mean square error (RMSE). The results of the crossover will mutate in specific genes and its position (x). Next, each particle knows the best value in all data ($Gbest$). Particles always move towards the optimum potential solution. The movement speed is influenced by the velocity that is renewed every iteration. The change in velocity of each particle is influenced by the value of the previous velocity, the $Pbest$ position, and the $Gbest$. Different random values are generated as $Pbest$ and $Gbest$ accelerations. Particle swarm optimization (PSO) is an algorithm to find the minimum or maximum function values based on a new population. PSO has the advantage of finding complex non-linear optimization values. PSO has similarities with the genetic algorithm which starts with a random population in the form of a matrix. However, PSO does not have evolution operators, namely crossover and mutation like those in the genetic algorithm. The line on the matrix is called particle or in the genetic algorithm as a chromosome that consists of the value of a variable. Each particle moves from its original position to a better position with a velocity. Initialization of HPSO is the same as initializing PSO, which is the number of particles distributed for global best position search in the optimization process.

Based on Figure 3, each particle considers itself the owner of the best location ($Pbest$) in each iteration. Determination of the best global position ($Gbest$) is determined after the search for the most optimal position of all particles that have considered themselves as $Pbest$. The $Gbest$ parameter is essential information for the movement of other particles in position search because it influences particle movement for the next iteration. The particles that occupy the position have not reached optimal move towards a particle that finds the best equation for updating the speed and position of particles is written in the equation as follows:

$$V_{id}^{t+1} = w \cdot V_{id}^{t} + c_1 \cdot r_1 \cdot (pbest_{id} - x_{id}) + c_2 \cdot r_2 \cdot (gbest_{id} - x_{id})$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$$

where V_{id}: velocity, x_{id}: particle position, t = iteration, d = initial number of particles, c_1, c_2: constants velocity, and r_1, r_2: random value. We can use inertia weight. We set $\alpha = 0.9$, $\beta = 0.4$ and maximum iteration 1000.

$$w = \alpha + \frac{\alpha - \beta}{\text{maximum iteration}} \times t$$

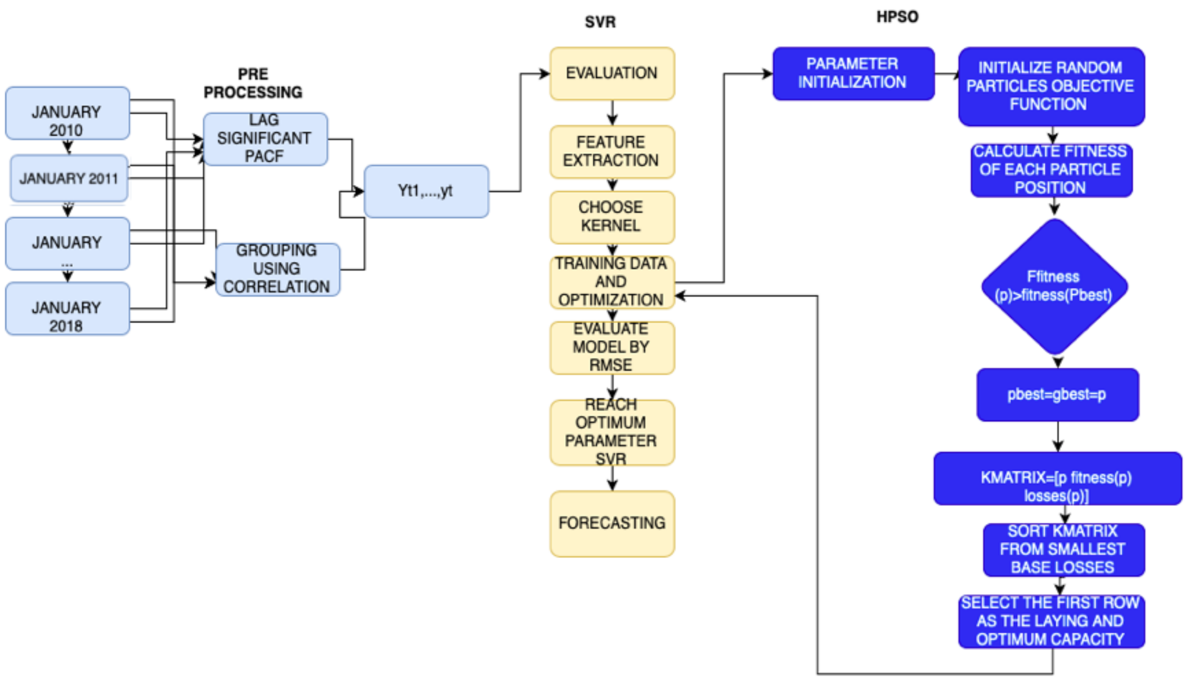

FIGURE 3. SVR-hybrid particle swarm model

3. **Analysis.** The research location is at Manado's Sam Ratulangi Meteorology Station. Climatologically, rain in Manado is a type of Monsunal rain with peak rainfall averages in January. Moreover, July is the peak of the dry season. From the opposite characters, atmospheric interactions are also different, from cloud cover conditions, sea surface temperature, monsoon winds, convectivity. Forecasting with the SVR method will be based

on input from the ARIMA model which already has significant parameters and compared with input from grouping data per year based on correlation. Figure 4 represents the correlation of January rainfall in 2010 until 2018, 3 main groups can be formed: Group 1 (January 2012, January 2015, January 2016), Group 2 (January 2010, January 2014, January 2018) and Group 3 (January 2011, January 2013, and January 2017). Then a comparison of inputs based on groups will be conducted based on correlation and input based on ARIMA.

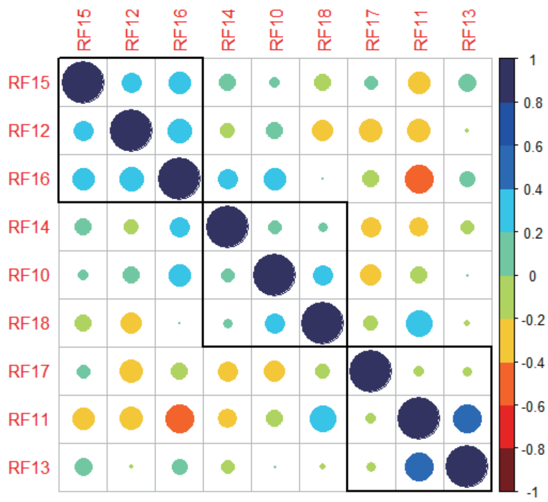

Figure 4. Group input SVR from correlation

Based on the decomposition of the ARIMA model in Table 1 the lag Y_t were obtained Y_{t-1}, Y_{t-2}, Y_{t-4} and Y_{t-12}. The following are the results of decomposition of several ARIMA models that already have significant parameters along with Y lag input.

Table 1. ARIMA

	Parameter	LAG	White Noise	Normality Test	Lag Input SVR
ARIMA $([1,2,4,12],0,0)$	$\widehat{\phi}_1 = 0.6112$	6	0.2017	< 0.0100	Y_1
		12	0.1651		
	$\widehat{\phi}_2 = 0.1721$	18	0.2851		Y_2
		24	0.4039		
	$\widehat{\phi}_4 = 0.1569$	30	0.5698		Y_4
		36	0.2450		
	$\widehat{\phi}_{12} = 0.2415$	42	0.2341		Y_{12}
		48	0.2212		

Based on the input lag, the lag will be input (X) in forecasting with SVR. After getting the input, the next step is to determine the parameters of the SVR model. We perform the SVR by different kernel functions by parameter setting $C(\text{cost}) = 100$, $\gamma = 0.01$ and $\varepsilon = 0.5$. To adjust the amount of these parameters, this study will use variations of trial and error. To get a good forecasting result, it will be combined several choices of parameter range values. To simplify the selection of parameters, the optimal range of the optimal parameter ε will be sought, where the range C and γ are set. The chromosomes that have been selected as prospective parents are given a uniform random number $(0, 1)$. Figure 5 illustrates the crossovers on chromosomes. If the value of the amount is less than the probability of crossing $(Pc = 0.8)$, the chromosome is selected as the parent, and a crossing process occurs.

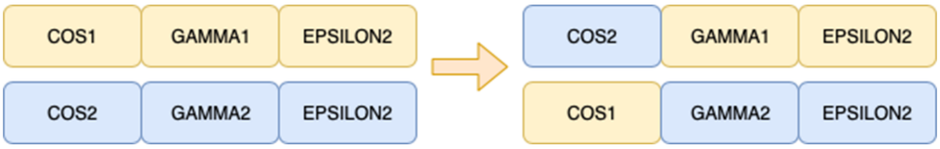

FIGURE 5. Crossover on chromosome

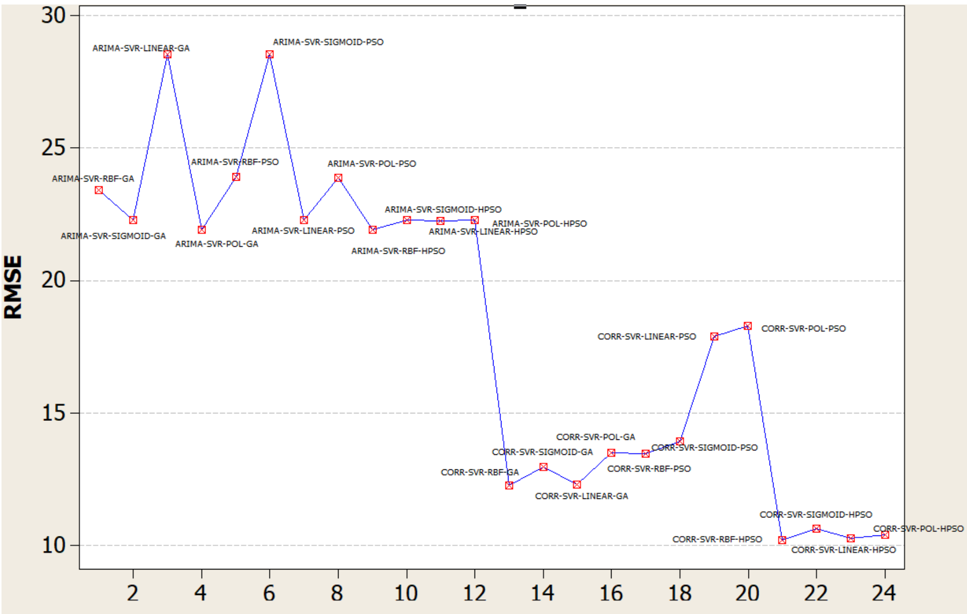

FIGURE 6. Accuracy

Based on Figure 6, RMSE is a more intuitive alternative than MSE because it has the same measurement scale as the data being evaluated. A low RMSE amount indicates that the variation in value produced by a forecast model is close to the variation in the value of its observations. Moreover, input SVR model based on correlation provides higher accuracy than ARIMA. Besides the smaller RMSE, the computational time required for the SVR-PSO and SVR-HPSO methods is faster than SVR-GA. The thing that causes quick time computing is the accuracy of the coefficient used so that the number of iterations is not too high.

The procedure of SVR-PSO and SVR-HPSO has many similarities to SVR-GA, where the system begins with a population formed from random solutions then the system looks for optimality through random generation updates. However, SVR-PSO does not have evolution operators, such as mutations and crossover. Conversely, potential solutions, namely individuals, or what are called particles, "fly" follow the optimum individuals at this time and reach the optimum particles. Each individual keeps track of its position in problem space. The traces of the position are interpreted as the best solution, or fitness in SVR-GA, which has been obtained so far. Thus, the information sharing mechanism owned by SVR-PSO and SVR-HPSO differs significantly from that of SVR-GA. In SVR-GA, each individual, called a chromosome, shares information with each other, so that the entire population moves as a whole towards optimality. In SVR-PSO and SVR-HPSO, only *gbest*, or *lbest*, gives information to others. This is a one-way information sharing mechanism. The evolutionary process is just looking for the best solution. Thus, all individuals, called particles, move converging rapidly to the best solution. Figure 7 explains the actual data (*) with predict data (–). If the predict data (–) follows the actual data (*) then the model can learn very well.

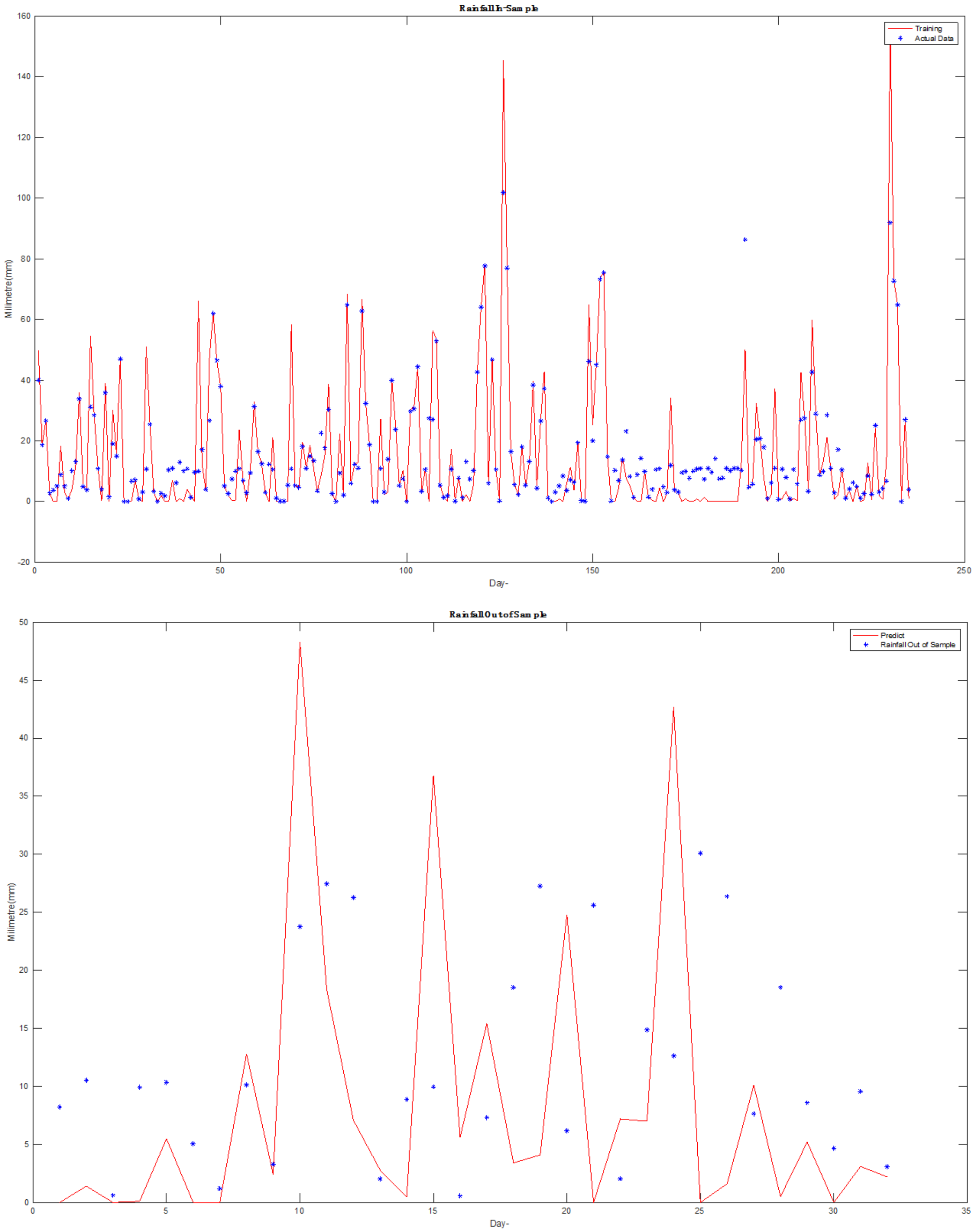

FIGURE 7. BEST-SVR-HPSO in Sample (top), and out Sample (bottom)

Figure 7 represents that even we propose hybrid methods still but there are still weaknesses to predict accurate data because the events of rain are so dynamic. Besides, in Figure 8 there are differences in precipitation from 2015 to 2018. If the colour gets red, it will increase the potential for rain in the area. While the blue colour explains that the potential for rain will decrease in that month.

4. **Conclusion.** The most significant contribution of this paper is to evaluate performance SVR by using a different kernel (linear, RBF, Sigmoid, Polynomial). We apply

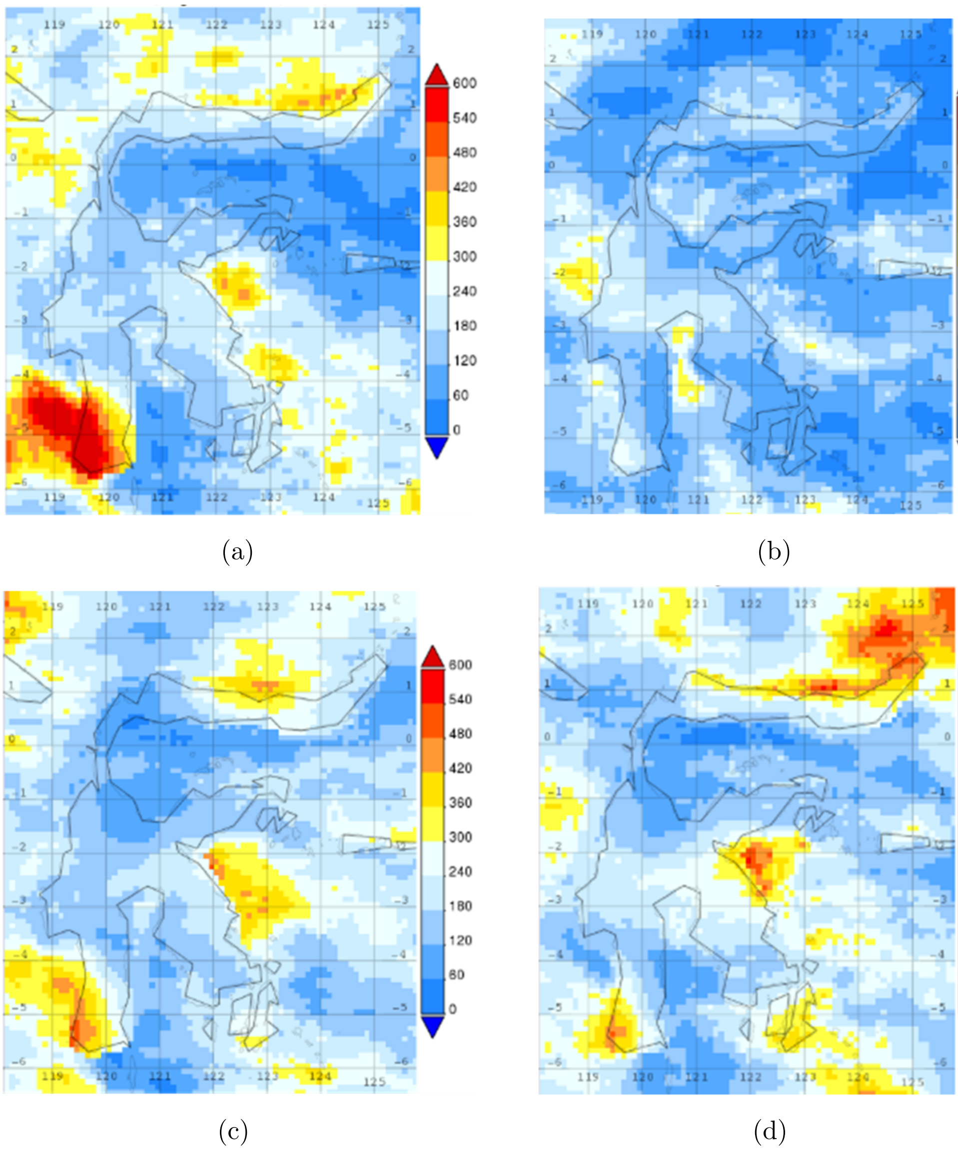

FIGURE 8. (color online) MSG precipitation in Sulawesi January 2015 (a), 2016 (b), 2017 (c), 2018 (d)

correlation and ARIMA to getting the best input in SVR, to take account of both the accuracy and interpretability of the forecast results. Secondly, we employ three suitable evolutionary algorithms to reduce the performance volatility of an SVR model with different parameters and optimization by using GA, PSO, HPSO. Based on the simulation results it can be concluded that the SVR-HPSO was able to handle complex and parallel problems. SVR-HPSO also has other advantages: having a simple concept, easy to implement, and efficient in calculations when compared to mathematical algorithms and other heuristic optimization techniques. However, SVR-GA can handle various kinds of optimization in its objective function (fitness) whether balanced or not balanced, linear or not linear, continuous or non-continuous, or with random noise. Comparing SVR-GA, SVR-PSO, and SVR-HPSO, SVR-HPSO is more flexible in maintaining a balance between global and local searches for its search space. In testing the number of iterations significantly affects the fitness results obtained each time. This is due to the frequent occurrence of the position update process that occurs in each iteration. The number of iterations will make slow or fast computing. We found during the testing inertia weights in SVR-PSO and SVR-HPSO, it can be obtained the high accuracy if the weight is greater than the inertia. Also, it will be a decrease in the speed of each iteration. In other words, the higher weight then the inertia, velocity of the particle will be slowed at the starting point of finding a solution. If the speed slows down at the start of the point, the search for solutions to this will provide an opportunity for local exploitation. However, since the rainfall is dynamic in next research we should concern to treat the outlier (extreme rainfall condition), feature engineering, and try another algorithm tuning.

Acknowledgment. This paper is supported by the Ministry of Science and Technology, Taiwan, under Grant MOST-107-2221-E-324-018-MY2 and MOST-106-2218-E-324-002, and supported by BDSRC Bina Nusantara University, Indonesia.

REFERENCES

[1] C. L. Wu and K. W. Chau, Prediction of rainfall time series using modular soft computing methods, *Eng. Appl. Artif. Intell.*, 2013.
[2] D. D. Prastyo, F. S. Nabila, Suhartono, M. H. Lee, N. Suhermi and S. F. Fam, VAR and GSTAR-based feature selection in support vector regression for multivariate spatio-temporal forecasting, in *Communications in Computer and Information Science*, 2019.
[3] C. Cortes and V. Vapnik, Support-vector networks, *Mach. Learn.*, vol.20, no.3, pp.273-297, 1995.
[4] S. R. Gunn, Support vector machines for classification and regression, *ISIS Tech. Rep.*, vol.14, no.2, pp.230-267, 1998.
[5] R. E. Caraka and S. A. Bakar, Evaluation performance of hybrid localized multi kernel SVR (LMKSVR) in electrical load data using 4 different optimizations, *J. Eng. Appl. Sci.*, vol.13, no.17, 2018.
[6] H. Yasin, R. E. Caraka, Tarno and A. Hoyyi, Prediction of crude oil prices using support vector regression (SVR) with grid search – Cross validation algorithm, *Glob. J. Pure Appl. Math.*, vol.12, no.4, pp.3009-3020, 2016.
[7] R. E. Caraka, S. A. Bakar, B. Pardamean and A. Budiarto, Hybrid support vector regression in electric load during national holiday season, in *Proceedings – 2017 International Conference on Innovative and Creative Information Technology: Computational Intelligence and IoT (ICITech 2017)*, 2018.
[8] C. Li, H. Zhang, H. Zhang and Y. Liu, Short-term traffic flow prediction algorithm by support vector regression based on artificial bee colony optimization, *ICIC Express Letters*, vol.13, no.6, pp.475-482, 2019.
[9] T. Hofmann, B. Schölkopf and A. J. Smola, Kernel methods in machine learning, *Ann. Stat.*, vol.36, no.3, pp.1171-1220, 2008.
[10] R. E. Caraka, S. A. Bakar and M. Tahmid, Rainfall forecasting multi kernel support vector regression seasonal autoregressive integrated moving average, *AIP Conf. Proc.*, vol.020014, 2019.
[11] Y. F. Ju and S. W. Wu, Village electrical load prediction by genetic algorithm and SVR, in *Proceedings – 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010)*, 2010.
[12] D. Whitley, A genetic algorithm tutorial, *Stat. Comput.*, 1994.
[13] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, vol.4, pp.1942-1948, 1995.
[14] K. Premalatha and A. M. Natarajan, Hybrid PSO and GA for global maximization, *Int. J. Open Probl. Compt. Math*, 2009.
[15] H. Yasin, R. E. Caraka, A. Hoyyi and Sugito, Stock price modeling using localized multiple kernel learning support vector machine, *ICIC Express Letters, Part B: Applications*, vol.11, no.4, pp.333-339, 2020.