International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-4 Issue-9, February 2015

Nonlinear Autoregressive (NAR) Forecasting Model for Potomac River Stage using Least Squares Support Vector Machines (LS-SVM)

Nian Zhang, Tilaye Alemayehu, Pradeep Behera

***Abstract*—This paper investigates the ability of a least-squares support vector machine (LS-SVM) model to improve the accuracy of streamflow forecasting. Cross-validation and grid-search methods are used to automatically determine the LS-SVM parameters in the forecasting process. To assess the effectiveness of this model, streamflow records from Geological Survey (USGS) gaging station 1652500 on Four Mile Run of the Potomac River, were used as case studies. The performance of the LS-SVM model is compared with the recurrent neural networks model trained by Levenberg-Marquardt backpropagation algorithm. The results of the comparison indicate that the LS-SVM model is a useful tool and a promising new method for streamflow forecasting.**

***Index Terms*—Water Quantity Prediction, Least Squares Support vector Machine.**

I. INTRODUCTION

In regard to stormwater runoff, how urbanized a watershed is or how developed a watershed is can be characterized by the degree of imperviousness found in the watershed [1]. A more urbanized watershed will have a greater percentage of area covered by impervious structures, i.e., roadways, rooftops, sidewalks, parking lots, etc. The effects of these impervious areas create higher peak flows and lower base flows in the watershed tributaries. These effects are most evident in the higher frequency rain/flood events, and they diminish as the range of magnitudes increases, i.e. the initial abstractions (infiltration, interception, and surface storage) become less significant when measured against rainfall for a large event, e.g. a 100-year rainfall event. Potomac River was determined to be one of the most polluted water bodies in the nation mainly due to the CSOs and stormwater discharges and wastewater treatment plant discharges. This highly urbanized Potomac River watershed suffers from serious water quantity problems including flooding and stream bank erosion. Of approximately 10,000 stream miles assessed in the watershed, more than 3,800 miles were deemed “threatened” or “impaired”.

Manuscript published on 28 February 2015.
***Correspondence Author(s)
Dr. Nian Zhang, Department of Electrical and Computer Engineering, University of the District of Columbia, 4200 Connecticut Ave. NW, Washington, DC, 20008, USA.
Mr. Tilaye Alemayehu, Department of Electrical and Computer Engineering, University of the District of Columbia, 4200 Connecticut Ave. NW, Washington, DC, 20008, USA.
Dr. Pradeep Behera, Department of Civil Engineering, University of the District of Columbia, 4200 Connecticut Ave. NW, Washington, DC, 20008, USA.

The middle Potomac sub-watershed, including Washington, DC, contains both the greatest percent impervious area and the greatest population density, which is home to 3.72 million or about 70% of the watershed’s population. In the next 20 years, the population of the Potomac watershed is expected to grow 10% each decade, adding 1 million inhabitants to reach a population of 6.25 million. In this regard, it is imperative to provide a reliable streamflow forecasting tool at various locations on the middle Potomac sub-watershed. Engineers, water resources professionals, and regulatory authorities need this streamflow information for planning, analysis, design, and operation & maintenance of water resources systems (e.g., water supply systems, dams, and hydraulic structures). Currently USGS provides the streamflow data at various locations in the form of gage height and discharge volume at specific locations, and we used this input to design a reliable prediction model. Recently a variety of computational intelligence has been proposed to address the water quantity prediction problem. In [2][3][4], a predictive model based on recurrent neural networks with the Levenberg-Marquardt backpropagation training algorithm to forecast the stormwater runoff. In [5], a recurrent neural network based predictive model was trained by a combination of particle swarm optimization and evolutionary algorithm to forecast the stormwater runoff discharge. Recent developments of least squares support vector machine (LS-SVM) has attracted an increasing attention in the fields of time series prediction [6]-[17]. However the investigation of the LS-SVM method on water quantity prediction has been very limited. Therefore, this paper will present a promising nonlinear autoregressive (NAR) model optimized by the LS-SVM using the previous discharge time series. This paper is organized as follows. In Section 2, the modeling and prediction with NAR Model with time-delay is brifly introduced. The Least Squares Support Vector Machines (LS-SVM) method is illustrated in detail. The practical implementation is introduced. In Section 3, the training data and time delays are depicted. The training method is designed. The model predicts future values on the testing data, as well as the training data. The experimental results of LS-SVM predictions on the water data are demonstrated. In Section 4, the conclusions are given.

II. GENERAL METHODOLOGY

A. NAR Model with Time-Delay

In the nonlinear autoregressive model (NAR) time series predictive model, the output is feedback to the input and the future values of time series y(t) could be predicted from past values of that time series, as shown in Fig. 1. Extending backward from time t, we have time series (y(t), y(t-1), y(t-2), • • •).

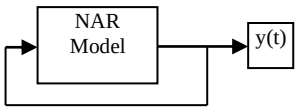

Fig. 1. The NAR based prediction model. The future values of y(t) can be predicted from past values of y(t).

This form of prediction can be written as follows:

$$y(t+s) = f(y(t-1),\ldots,y(t-d))$$

where s is called the horizon of prediction. If s = 1, then this prediction is called one time step ahead prediction; otherwise, it is called multi-step ahead prediction. d is the time delay, giving the number of past predictions fed into the model.

B. Least Squares Support Vector Machine Regression with Symmetry Constraints

Least Squares Support Vector Machines (LS-SVM) is a powerful nonlinear kernel methods, which use positive-definite kernel functions to build a linear model in the high-dimensional feature space where the inputs have been transformed by means of a nonlinear mapping $\emptyset$ [18]. This is converted to the dual space by means of the Mercer's theorem and the use of a positive definite kernel, without computing explicitly the mapping $\emptyset$. The LS-SVM formulation solves a linear system in dual space under a least-squares cost function [19], where the sparseness property can be obtained by sequentially pruning the support value spectrum [20] or via a fixed-size subset selection approach. The LS-SVM training procedure involves the selection of a kernel parameter and the regularization parameter of the cost function, which can be done e.g. by cross-validation, Bayesian techniques [21] or others. Given the sample of N points $\{x_i, y_i\}_{i=1}^{N}$, with input vectors $x_i \in \mathbb{R}^p$ and output values $y_i \in \mathbb{R}$, the goal is to estimate a model of the form:

$$y_i = w^T \emptyset(x_i) + b + \varepsilon_i (i=1,2,\ldots,l) \quad (1)$$

where $\emptyset(\cdot)$: $\mathbb{R}^p \rightarrow \mathbb{R}^{n_h}$ is the mapping to a high dimensional (and possibly infinite dimensional) feature space, and the residuals e are assumed to be independent and identically distributed with zero mean and constant and finite variance. Least squares support vector machine (LS-SVM) formulates a regularized cost function and changes its inequation restriction to equation restriction. As a result, the solution process becomes a solution of a group of equations which greatly accelerates the solution speed [19]. The following optimization problem with a regularized cost function is formulated:

$$\min_{w,b,\varepsilon_i} \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{l} \varepsilon_i^2 \quad (2)$$

The solution of LS-SVM regressor will be obtained after we construct the Lagrangian function. The extreme point of Q is a saddle point, and differentiating Q can provide the formulas as follows, using Lagrangian multiplier method to solve the formulas. The conditions for optimality are

$$\frac{\partial Q}{\partial w} = w - \sum_{i=1}^{l} \alpha_i \emptyset(x_i) = 0 \quad (3)$$

$$\frac{\partial Q}{\partial b} = -\sum_{i=1}^{l} \alpha_i = 0 \quad (4)$$

$$\frac{\partial Q}{\partial \alpha} = w^T - \emptyset(x_i) + b + \varepsilon_i - y_i = 0 \quad (5)$$

$$\frac{\partial Q}{\partial \varepsilon_i} = C\varepsilon_i - \alpha_i = 0 \quad (6)$$

where $\alpha \in \mathbb{R}$ are the Lagrange multipliers. From formulas above, we can obtain:

$$\frac{1}{2} \sum_{i=1}^{l} \alpha_i \emptyset(x_i) \sum_{j=1}^{l} \alpha_j \emptyset(x_j) + \frac{1}{2C} \sum_{i=1}^{l} \alpha_i^2 + b \sum_{i=1}^{l} \alpha_i = \sum_{i=1}^{l} \alpha_i y_i \quad (7)$$

The formula above can be expressed in matrix form:

$$\begin{bmatrix} 0 & e^T \\ e & \Omega + C^{-1}I \end{bmatrix} (l+1)(l+1) \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \quad (8)$$

In this equation,

$$e = [1, \ldots, 1]_x^T$$

$$\Omega_{ij} = K(x_i, x_j) = \emptyset(x_i)^T \emptyset(x_j) \quad (9)$$

Formula (7) is a linear equation set corresponding to the optimization problem and can provide us with α and b. Thus, the prediction output decision function is:

$$\bar{y}(x) = \sum_{i=1}^{l} \alpha_i K(x_i x) + b \quad (10)$$

where $K(x_i, x)$ is the core function.

C. Practical Implementation

The training process of LS-SVM involves the selection of kernel parameter: the squared bandwidth, σ2 (sig2) and the regularization constant, γ (gam). The regularization constant, γ (gam) determines the trade-off between the training error minimization and smoothness A good choice of these parameters is crucial for the performance of the estimator. The tuning parameters were found by using a combination of coupled simulated annealing (CSA) and a standard simplex method. The CSA finds good starting values and these values were passed to the simplex method in order to fine tune the result. We use 10-fold cross-validation for selecting these parameters. Another important choice is the selection of regressors, i.e., which lags of inputs and outputs are going to be included in the regression vector. This selection is done by using a large number of initial components and then performing a greedy search to prune non-informative lags on a cross-validation basis. Therefore an initial model containing all regressors is estimated and optimal choices for the parameters are made.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

On each stage of the greedy backwards elimination process, a regressor is removed if the cross-validation mean absolute error or mean squared error improves. For the purpose of model estimation, all series are normalized to zero mean and unit variance. Once the parameters are calculated, the final set of regressors is then used for the predictions. By using only a subset of the total data available, we can compare the predictions against real values to see how accurate the prediction is.

III. EXPERIMENTAL RESULTS

A. Study Area

The study area will focus on the Four Mile Run at Alexandria, VA, as shown in Fig. 2. The US Geological Survey (USGS) gaging station 1652500 on Four Mile Run located at the Shirlington Road Bridge has collected stream flow data since 1951 [1]. The Four Mile Run is 9.2 miles long, and is a direct tributary of the Potomac River.

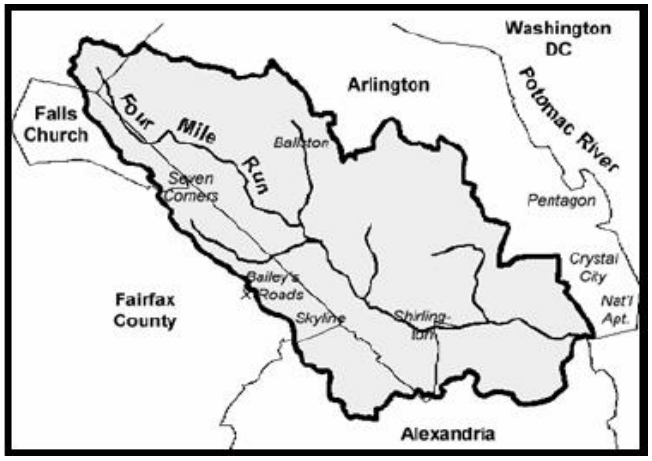

Fig. 2. Four Mile Run at Alexandria, VA is a nine-mile long stream located in a highly urbanized area in Northern Virginia. It is a direct tributary of the Potomac River, which ultimately carries the water flowing from Four Mile Run to the Chesapeake Bay.

The entire watershed can be classified as highly urbanized, which ultimately flows through some of Northern Virginia's most densely populated areas to the Chesapeake Bay. In addition, because of the highly urbanized nature of the Four Mile Run watershed, the neighborhoods and businesses adjacent to this portion of the run were subjected to repeated flooding, beginning in the 1940s. Therefore, the flood-control solutions are the major concern. Runoff prediction would provide a promising solution for flood-control.

B. Time Series Data from USGS

The real-time USGS data for the Four Mile Run station include the discharge data, which is useful for investigating its impact to the long-run discharge forecast. The discharge is the volume of water flowing past a certain point in a water-flow. For example, the amount of cubic feet passing through a drain per second is a measure of discharge. The discharge data was retrieved for 120 days between August 28, 2010 and December 4, 2010. Because the real-time data typically are recorded at 15-minute intervals, the runoff discharge (cubic feet per second) data plots 34721 data during the 120 days, as shown in Fig. 8. The discharge will be presented to the system as an input. It is a 34721x1 vector, representing dynamic data, i.e. 34721 time steps. It is challenging that these discharge values vary significantly over time. As shown in Fig. 3, the baseline is at around 4 on the Y-axis, with peaks reaching 8, with very little repetition to the pattern, making it more difficult to predict future values.

C. Training Data and Time Delays

The first 500 time series data from the original sample of about 34,721 were used for our analysis. To determine an appropriate time delay or lag, we increase the number of delays lags until the network performed well. After a number of experiments, 80 is determined to be the smallest lag number that ensures a good performance. That means the model will use the past 80 input data to predict a future data. Before parameter tuning and network training, we should use the function windowize to convert the time-series into a Hankel matrix useful for training a nonlinear function approximation [22]. For example, assume there is a matrix X which is defined below.

$$X = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \\ d_1 & d_2 & d_3 \\ e_1 & e_2 & e_3 \\ f_1 & f_2 & f_3 \\ g_1 & g_2 & g_3 \end{bmatrix}$$

1st window

Now we want to convert matrix X to a new matrix Xu by running the Matlab command:

Xu = windowize(X, [1 2 3])

This command will select 3 rows of data (i.e. circled by the blue dashed line) from matrix X to make a window, and pout this window in a row of matrix Xu. For example, row 1 to 3 from matrix X will be selected to make the 1st window, and put in the 1st row of matrix Xu. Similarly, row 2 to 4 from matrix X will be selected to make the 2nd window, and put in the 2nd row of matrix Xu. Thus, the matrix Xu will look as follows.

1st window

$$W = \begin{bmatrix} a_1 & a_2 & a_3 & b_1 & b_2 & b_3 & c_1 & c_2 & c_3 \\ b_1 & b_2 & b_3 & c_1 & c_2 & c_3 & d_1 & d_2 & d_3 \\ c_1 & c_2 & c_3 & d_1 & d_2 & d_3 & e_1 & e_2 & e_3 \\ d_1 & d_2 & d_3 & e_1 & e_2 & e_3 & f_1 & f_2 & f_3 \\ e_1 & e_2 & e_2 & f_1 & f_2 & f_3 & g_1 & g_2 & g_3 \end{bmatrix}$$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

In our case, Xu = windowize(X, 1:lag+1) will convert the discharge data set into a new input vector including the past measurements and the future output by *windowize*. The size of the discharge data set contains 500 data points, which consists of 500 rows. With the 80 lags, it will generate 420 rows and 81 columns. The last column of the resulting matrix Xu contains the future values of the time-series, and the previous 80 columns contain the past inputs. The first 340 data points (i.e. 70%) will be used as training data, and the remaining 160 data (i.e. 30%) will be used as test data. Xtra = Xu(1:end-lag,1:lag) will generate 80 past inputs, i.e. x(t-1), x(t-2), … x(t-80), while Ytra = Xu(1:end-lag,end) contains their actual future value, x(t). Ytra will be used as the target for those past inputs.

D. Tuning the Parameters

In order to build an LS-SVM model, we need to tune the regularization constant, gam and the kernel parameter, sig2. γ (gam) determines the trade-off between the training error minimization and smoothness. In the common case of the Gaussian RBF kernel, the kernel parameter, sig2 is the squared bandwidth. We use the following statement to tune these parameters:

```
[gam,sig2] = tunelssvm({Xtra,Ytra,'f',[],[],'RBF_kernel'},...
'simplex','crossvalidatelssvm',{10,'mae'})
```

Where f stands for function estimation. The Kernel type is chosen to be the default RBF kernel. The optimization function is specified as simplex. The simplex is a multidimensional unconstrained non-linear optimization method. Simplex finds a local minimum of a function starting from an initial point X. The local minimum is located via the Nelder-Mead simplex algorithm [23]. The model adopts crossvalidatelssvm as the cost function. It estimates the generalization performance of the model. It is based upon feedforward simulation on the validation set using the feedforwardly trained model. In addition, 10 means 10-fold. We use 10-fold cross-validation because the input size is greater than 300 points. Otherwise, leave-one-out cross-validation will be used when the input size is less or equal than 300 points. The 10-fold cross-validation method will break data (the size of the data is assumed to be n) into 10 sets of size n/10, then train on 9 datasets and test on 1, and then repeat 10 times and take a mean accuracy. mae is the mean absolute error and is used in combination with the 10-fold cross-validation method. It is the absolute value of the difference between the forecasted value and the actual value. It tells us how big of an error we can expect from the forecast on average. The tuning of the parameters is conducted in two steps. First, a state-of-the-art global optimization technique, Coupled Simulated Annealing (CSA) [24], determines suitable parameters according to some criterion. Second, these parameters are then given to a second optimization procedure simplex to perform a fine-tuning step. The parameter tuning results are shown in Fig. 3. Coupled Simulated Annealing chosen the initial gam to be 1364.706, and sig2 to be 13.989. They serve as the starting values for the simplex optimization routine. After 11 iterations, the gam and sig2 are optimized to be 83.2188 and 15.298, respectively.

E. Network Training and Prediction

Once the gam and sig2 parameters were tuned, we should train the network. It will train the support values and the bias term of an LS-SVM for function approximation. The Matlab command is

```
[alpha,b] = trainlssvm({Xtra,Ytra,'f',gam,sig2,'RBF_kernel'})
```

Xtra and Ytra are the training data we defined before. f stands for function estimation. The Kernel type is chosen to be the default RBF kernel. Because the network has 80 lags, it helps generate 80 past inputs. For each iteration, the past 80 Xtra data points will be used to predict the 81th data point. Ytra is the desired target. The 340 samples in the Xtra and Ytra will be used to train the network. After the network has been well trained, we can test the prediction performance by testing on the new data, which have never been seen by the network. We will use the remaining 160 data points as the testing data. The Matlab command is

```
prediction = predict({Xtra,Ytra,'f',gam,sig2,…
'RBF_kernel'},Xs,500)
```

Xtra and Ytra are the training data we used before. 'f' stands for function estimation. The Kernel type is chosen to be the default RBF kernel. Xs is the starting point for iterative prediction. Since we want to check both the training performance and prediction performance, we set Xs=X(1:end-lag,1). The model will start predicting from the 1st data point, and will predict the next 500 points from the start point. The predicted discharge value and the actual discharge value were shown in Fig. 3.

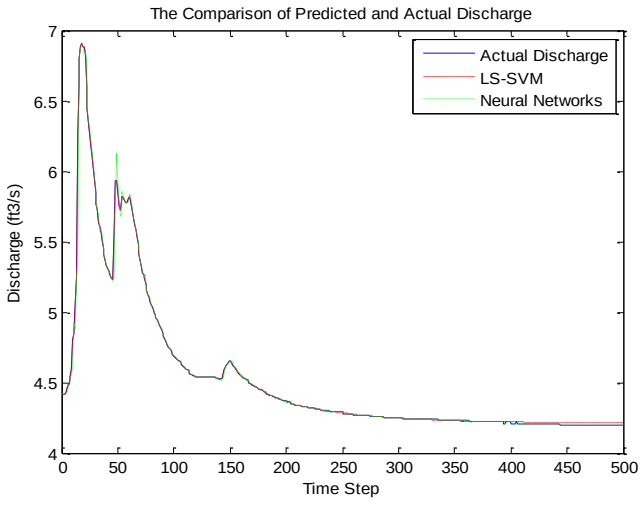

Fig. 3 Performance comparison of the LS-SVM model and the recurrent neural networks model trained by the Levenberg-Marquardt backpropagation algorithm. The actual USGS discharge is shown in blue line, the LS-SVM prediction is shown in red dashdot, and the recurrent neural networks model is shown in green dashed line. The first 340 samples are training data, and the remaining 160 samples are test data.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The actual USGS discharge is shown in blue line, the LS-SVM prediction is shown in red dashdot, and the recurrent neural networks model is shown in green dashed line. The first 340 samples are training data, and the remaining 160 samples are testing data. As shown in Fig. 3, the prediction on the training data matches the actual values perfectly. This makes sense because these training samples have been seen by the network during training. The prediction on these data should have already been trained to be very close to the actual value. In addition, when we test the new data from time step 341 to 500, we find the predicted values match very well with the actual values. This demonstrated that the LS-SVM model has excellent prediction ability. In order to further evaluate the performance of the proposed LS-SVM method, we compare the results with the recurrent neural networks model trained by Levenberg-Marquardt backpropagation algorithm [4]. The simulation result is shown in Fig. 3. The USGS discharge is shown in blue line, the LS-SVM prediction is shown in red dashdot, and the recurrent neural networks model trained by Levenberg-Marquardt backpropagation algorithm is shown in green dash colon. The first 340 samples are training data, and the remaining 160 samples are test data.

IV. CONCLUSIONS

In this paper, the least squares support vector machine (LS-SVM) based algorithm is developed to forecast the future streamflow based on the previous streamflow. The first 340 data points are used as training data, and the remaining 160 data are testing data. First we convert the time-series into a Hankel matrix useful for training a nonlinear function approximation. Next we build an LS-SVM model by tuning the regularization constant, gam and the kernel parameter, sig2. A Gaussian Radial Basis Function (RBF) kernel framework was built on the data set to optimize the tuning parameters. The 10-fold cross-validation method is used to estimate the generalization performance of the model. Then we train the LS-SVM network. It trains the support values and the bias term of an LS-SVM for function approximation. We developed an effective training scheme. After the network has been well trained, we test the prediction performance by predicting new values on the testing samples, as well as the training samples. The performance of the LS-SVM model is compared with the recurrent neural networks model trained by Levenberg-Marquardt backpropagation algorithm. The excellent experimental results of the comparison indicate that the LS-SVM model is a useful tool and a promising new method for streamflow forecasting. The excellent experimental results demonstrated that the proposed LS-SVM based predictive model has superior prediction performance on not only the training samples, but also the testing samples. In addition, the proposed parameter tuning method and the training scheme worked effectively, which ensure an accurate prediction of streamflow.

V. ACKNOWLEDGMENT

The authors would like to express thanks to the University of the District of Columbia STEM Center (NSF/HBCU-UP/HRD-0928444) grant and DC Water Resources Research Institute (WRRI) Grant.

REFERENCES

1. N. Zhang, “Urban Stormwater Runoff Prediction Using Recurrent Neural Networks,” *The Eighth International Symposium on Neural Networks (ISNN)*, Guilin, China, 2011.
2. N. Zhang, “Prediction of Urban Stormwater Runoff in Chesapeake Bay Using Neural Networks,” *The Eighth International Symposium on Neural Networks (ISNN)*, Guilin, China, 2011.
3. N. Zhang and S.H. Lai, “Runoff Quantity Analysis of Urban Catchments Based on Neural Networks Method,” *The IASTED International Symposia on Imaging and Signal Processing in Healthcare and Technology (ISPHT)*, Washington, DC, May 16-18, 2011.
4. N. Zhang and S.H. Lai, “Water Quantity Prediction Based on Particle Swarm Optimization and Evolutionary Algorithm Using Recurrent Neural Networks,” *2011 International Joint Conference on Neural Networks (IJCNN)*, San Jose, CA, July 31-August 5, 2011.
5. G. Ji, J.C. Wang, Y. Ge, and H.J. Liu, “Urban Water Demand Forecasting by LS-SVM with Tuning Based on Elitist Teaching-Learning-Based Optimization”, *The 26th Chinese Control and Decision Conference (2014 CCDC)*, pp. 3997-4002, 2014.
6. Z. Liu, X. Wang, L. Cui, X. Lian, J. Xu, “Research on Water Bloom Prediction Based on Least Squares Support Vector Machine,” *2009 WRI World Congress on Computer Science and Information Engineering*, vol.5, pp. 764 - 768, 2009.
7. Y. Xiang and L. Jiang, “Water Quality Prediction Using LS-SVM and Particle Swarm Optimization,” *Second International Workshop on Knowledge Discovery and Data Mining*, pp. 900- 904, 2009.
8. W. Liu, K. Chen, and L. Liu, “Prediction Model of Water Consumption Using Least Square Support Vector Machines Optimized by Hybrid Intelligent Algorithm,” *2011 Second International Conference on Mechanic Automation and Control Engineering (MACE)*, pp. 3298- 3300, 2011.
9. L. Liang and F. Xie, “Applied Research on Wastewater Treatment Based on Least Squares Support Vector Machine,” *2011 International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE)*, pp. 4825- 4827, 2011.
10. X. Zhang, S. Wang, and Y. Zhao, “Application of Support Vector Machine and Least squares Vector Machine to Freight Volume Forecast,” *2011 International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE)*, pp. 104-107, 2011.
11. R.J. Liao, J.P. Bian, L.J. Yang, S. Grzybowski, Y.Y. Wang, and J. Li, “Forecasting Dissolved Gases Content in Power Transformer Oil Based on Weakening Buffer Operator and Least Square Support Vector Machine–Markov,” *Generation, Transmission & Distribution*, IET, vol. 6, no. 2, pp. 142- 151, 2012.
12. L. Hou, Q. Yang, J. An, “An Improved LSSVM Regression Algorithm,” *International Conference on Computational Intelligence and Natural Computing*, vol. 2, pp. 138- 140, 2009.
13. X. Zhang, Y. Zhao, and S. Wang, “Reliability Prediction of Engine Systems Using Least Square Support Vector Machine,” *2011 International Conference on Electronics, Communications and Control (ICECC)*, pp. 3856- 3859, 2011.
14. N. Zhang, C. Williams, E. Ososanya, and W. Mahmoud, “Streamflow Prediction Based on Least Squares Support Vector Machines,” *ASEE-2013 Mid-Atlantic Fall Conference*, Washington, D.C., October 11-13, 2013.
15. T. A. Stolarski, “System for Wear Prediction in Lubricated Sliding Contacts,” *Lubrication Science*, 1996, 8 (4): 315 -351.
16. Suykens J A K,Vandewalle J, “Least Squares Support Vector Machine Classifiers,” *Neural Processing Letter*, 1999, 9(3):293-300.
17. J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, “Least Squares Support Vector Machines,” *World Scientific*, Singapore, 2002.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

18. J.A.K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters*, 9:293–300, 1999.
19. J.A.K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted Least Squares Support Vector Machines: Robustness and Sparse Approximation," *Neurocomputing*, vol. 48, no. 1-4, pp. 85–105, 2002.
20. D.J.C. MacKay, "Comparison of Approximate Methods for Handling Hyperparameters," *Neural Computation*, vol. 11, pp. 1035–1068, 1999.
21. De Brabanter K., Karsmakers P., Ojeda F., Alzate C., De Brabanter J., Pelckmans K., De Moor B., Vandewalle J., Suykens J.A.K., "LS-SVMlab Toolbox User's Guide version 1.8," Internal Report 10-146, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2010.
22. J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Computer Journal*, 7, 308-313, 1965.
23. S. Xavier-de-Souza, J.A.K. Suykens, J. Vandewalle, and D. Bolle, "Coupled Simulated Annealing," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 40, no. 2, pp. 320–335, 2010.

AUTHOR PROFILE

Dr. Nian Zhang, received her B.S. in Electrical Engineering at the Wuhan University of Technology, M.S. in Electrical Engineering from Huazhong University of Science and Technology, and Ph.D. in Computer Engineering from Missouri University of Science and Technology. She is an Assistant Professor of the Department of Electrical and Computer Engineering at the University of the District of Columbia, Washington DC. Dr. Zhang's research expertise and interests include support vector machines, neural networks, fuzzy logic, computational intelligence methods, and their applications on pattern recognition, signal and image processing, time series prediction, renewable energy, biomedical engineering, and autonomous robot navigation.

Mr. Tilaye Alemayehu, is a senior student at the University of the District of Columbia in the Department of Electrical and Computer Engineering. He has a strong interest in time series prediction, optimization, and machine learning

Dr. Pradeep Behera, received his Ph.D. degree from the Civil & Environmental Engineering, University of Toronto, Canada. He is the Professor and Chair of the Department of Civil Engineering at the University of the District of Columbia, Washington DC. His research interests include Urban Stormwater Management, Non-point Source Pollution, Water Resources Engineering, Erosion and Sediment Control, Sustainable Urban Water Systems, Environmental Systems, and Spatio-Temporal Informatics.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)