

Grid Resource Prediction based on Support Vector Regression and Simulated Annealing Algorithms

Ying Zheng

Editorial Department of Journal

Inner Mongolia University for Nationalities

Tongliao 028043, China

E-mail: mindaxuebao@163.com

Abstract

Accurate grid resources prediction is crucial for a grid scheduler. In this study, support vector regression (SVR), which is a novel and effective regression algorithm, is applied to grid resources prediction. In order to build an effective SVR model, SVR's parameters must be selected carefully. Therefore, we develop a simulated annealing algorithm-based SVR (SA-SVR) model that can automatically determine the optimal parameters of SVR with higher predictive accuracy and generalization ability simultaneously. The performance of the hybrid model (SA-SVR), the back-propagation neural network (BPNN) and traditional SVR model whose parameters are obtained by trial-and-error procedure (T-SVR) have been compared with benchmark data set. Experimental results demonstrate that SA-SVR model works better than the other two models.

Keywords: Grid resource prediction, Support vector regression, Simulated annealing

1. Introduction

Grid computing (I. Foster, C. Kesselman, 1999) derives its name from the analogy with the electricity grid. A commonly used definition of grid is the internet-based infrastructure that aggregates geographically distributed and heterogeneous resources as an ensemble to solve large-scale problems. In the grid environment, the resources are provided and managed by different administrators. The availability of grid resources vary over time and such changes will affect the performance of the tasks running on the grid. If we can predict the future information of grid resources, the scheduler will be able to manage the grid resources more effectively.

In grid resource prediction, many relevant research models (P. Dinda ,D. O'Hallaron., 1999)(R. Wolski, N. T. Spring, and J. Hayes, 1999)(Z. X. Liu, X. P. Guan, H. H. Wu, 2006)(A. Eswaradass, X. H. Sun, M. Wu, 2005) have been developed and have generated accurate prediction in practice. These prediction models that provide future resources information generally apply the time series prediction models which mostly use statistical and artificial intelligent approach.

Resource Prediction System (RPS) (P. Dinda ,D. O'Hallaron., 1999) is a project in which grid resources are modeled as linear time series process. Multiple conventional linear models are evaluated, including AR, MA, ARMA, ARIMA and ARFIMA models. Their results show that the simple AR model is the best model of this class because of its good predictive power and low overhead.

The Network Weather Service (NWS) (R. Wolski, N. T. Spring, and J. Hayes, 1999) uses a combination of several models for the prediction of one resource. NWS allows some adaptation by dynamically choosing the model that has performed the best recently for the next prediction, but its adaptation is limited to the selection of a model from several candidates that are conventional statistical models.

With the development of artificial neural networks (ANNs), ANNs have been successfully employed for modeling time series. Liu et al. (Z. X. Liu, X. P. Guan, H. H. Wu, 2006) and Eswaradass et al. (A. Eswaradass, X. H. Sun, M. Wu, 2005) applied ANNs to grid resources prediction successfully. Experimental results showed the ANN approach provided an improved prediction over that of NWS. However, ANNs have some drawbacks, such as hard to pre-select the system architecture, spending much training time, and lacking knowledge representation facilities.

In 1995, support vector machine (SVM) was developed by Vapnik (Vapnik, V., 1998) to provide better solutions than ANNs. SVM can solve classification problems (SVC) and regression problems (SVR) successfully and effectively. However, the determination of SVR's parameters is an open problem and no general guidelines are available to select these parameters (Vapnik, V., 1998). Simulated annealing algorithm (SA) (Van Laarhoven

PJM, Aarts EHL, 1987) is an effective optimization algorithm, and it has been successfully applied to various NP-hard combinatorial optimization problems. Therefore, in this study, SA was adopted to automatically determine the optimal hyper-parameters of SVR.

The rest of the literature is as follows. In Section 2, we describe the theory of SVR in detail. Section 3 describes the SA-SVR prediction model. In Section 4, we introduce the performance evaluation. Finally in Section 5, we conclude the paper.

2. Support vector regression

Because it is out of the scope of this study to explain the theory on SVR completely, this section focuses on some highlights representing crucial elements in using this method. Detailed descriptions of SVR can be found in Vapnik (Vapnik, V., 1998), Smola and Schölkopf (Smola A, Schölkopf, B., 1998) and L.P.Wang (L.P.Wang, 2005).

2.1 Linear SVR

In order to solve regression problems, we are given training data (x_i, y_i) ($i=1, \dots, l$), where x is a d -dimensional input with $x \in R^d$ and the output is $y \in R$. The linear regression model can be written as follows (L.P.Wang, 2005):

$$f(x) = \langle \omega, x \rangle + b, \quad \omega, x \in R^d, b \in R \tag{1}$$

where $f(x)$ is a target function and $\langle \cdot, \cdot \rangle$ denotes the dot product in R^d .

In order to measure the empirical risk (L.P.Wang, 2005), we should specify a loss function. Several alternatives are available. The most common loss function is the ϵ -insensitive loss function.

The ϵ -insensitive loss function proposed by Vapnik is defined in the following function:

$$L_\epsilon(y) = \begin{cases} 0 & \text{for } |f(x) - y| \leq \epsilon \\ |f(x) - y| - \epsilon & \text{otherwise} \end{cases} \tag{2}$$

and the optimal parameters and b in Eq.(1) are found by solving the primal optimization problem(L.P.Wang, 2005):

$$\min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^l (\xi_i^- + \xi_i^+) \tag{3}$$

with constraints:

$$\begin{aligned} y_i - \langle \omega, x_i \rangle - b &\leq \epsilon + \xi_i^+, \\ \langle \omega, x_i \rangle + b - y_i &\leq \epsilon + \xi_i^-, \\ \xi_i^-, \xi_i^+ &\geq 0, \quad i = 1, \dots, l, \end{aligned} \tag{4}$$

where C is a pre-specified value that determines the trade-off between the flatness of $f(x)$ and the amount up to which deviations larger than the precision are tolerated. The slack variables ξ^+ and ξ^- represent the deviations from the constraints of the ϵ -tube.

This primal optimization problem can be reformulated as a dual one defined as follows:

$$\begin{aligned} \max_{x, x^*} \quad & -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (a_i^* - a_i)(a_j^* - a_j) \langle x_i, x_j \rangle + \\ & \sum_{i=1}^l y_i (a_i^* - a_i) - \epsilon \sum_{i=1}^l (a_i^* + a_i) \end{aligned} \tag{5}$$

with constraints:

$$\begin{aligned} 0 &\leq a_i, a_i^* \leq C, \quad i = 1, \dots, l \\ \sum_{i=1}^l (a_i - a_i^*) &= 0. \end{aligned} \tag{6}$$

Solving the optimization problem defined by Eq.(5) and (6) gives the optimal Lagrange multipliers α and α^* , while $\bar{\omega}$ and \bar{b} are given by

$$\begin{aligned}\bar{w} &= \sum_{i=1}^l (a_i^* - a_i) x_i, \\ \bar{b} &= -\frac{1}{2} \langle \bar{w}, (x_r + x_s) \rangle,\end{aligned}\quad (7)$$

where x_r and x_s are support vectors.

According to the computed value of \bar{w} , the $f(x)$ in Eq.(1) can be written as:

$$f(x) = \sum_{i=1}^N (a_i - a_i^*) \langle x_i, x \rangle + b \quad (8)$$

Because of the specific formulation of the cost function and the use of the Lagrange theory, the solution has several interesting properties. It can be proven that the solution found is always global because the problem is convex. In addition, not all training points contribute to the solution found because of the fact that some Lagrange multipliers are zero. If these training points would have been removed in advance, the same solution would have been obtained (sparseness). Training points with nonzero Lagrange multipliers are called support vectors and give shape to the solution. The smaller the fraction of support vectors, the more general the obtained solution is and less computation is required to evaluate the solution for a new and unknown object.

2.2 Nonlinear SVR

Sometimes nonlinear functions should be optimized, so this approach has to be extended. This is done by replacing x_i by a mapping into feature space (L.P.Wang, 2005), $\varphi(x_i)$, which linearizes the relation between x_i and y_i . In the feature space, the original approach can be adopted in finding the regression solution.

When using a mapping function, the solution of Eq.(8) can be changed into:

$$f(x) = \sum_{i=1}^N (a_i - a_i^*) K(x_i, x) + b \quad (9)$$

In Eq.(9), $K(x_i, x) = \langle \varphi(x_i), \varphi(x) \rangle$ is the so-called kernel function. An inner product in feature space has an equivalent kernel function K in input space (L.P.Wang, 2005)

$$K(x, y) = \langle \varphi(x), \varphi(y) \rangle$$

Any symmetric positive semi-definite function (L.P.Wang, 2005) that satisfies Mercer's Conditions (L.P.Wang, 2005) can be used as a kernel function. Kernel function is proven to simplify the use of a mapping. Finding this mapping can be troublesome for each dimension of the specific mapping has to be known. Of all kernel functions, the radial basis function (RBF) has received significant attention and is used most widely. Our work is also based on the RBF kernel. The form of RBF is:

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (10)$$

where σ is the width of the RBF kernel. Other kernel functions that are usually used include polynomial sigmoid, and B-spline kernel functions (L.P.Wang, 2005), etc.

3. SA-SVR model

Generally, when selecting the parameters, most researchers still follow the trial-and-error procedure. However, this procedure is time-consuming and requires some luck.

The study proposes SA-SVR model which could optimize all SVR's parameters simultaneously. The proposed SA-SVR model dynamically optimizes SVR's parameters, and then uses the acquired parameters to construct optimized SVR model. Fig. 1 illustrates the SA-SVR model.

The definition of a better solution (the better SVR's parameters) is important. In order to overcome over-fitting phenomenon, cross validation technique, which was successfully adopted by Duan (Duan, K., Keerthi, S., & Poo, A. 2001), is used in SA-SVR model. In this study, five-fold cross validation is used (Duan, K., Keerthi, S., & Poo, A. 2001). The performance of the parameter set is measured by the MSE (Mean Square Error) (Eq. (11)) on validation set (Duan, K., Keerthi, S., & Poo, A. 2001).

$$\min \quad MSE_{cv}$$

$$MSE_{cv} = \frac{\sum_{i=1}^l (a_i - p_i)^2}{l} \quad (11)$$

Where a_i is the actual value and p_i is the predicted value. cv means cross validation. The solution with a smaller MSE_{cv} is the better solution.

The work process of SA-SVR is described as follows:

Step 1 (Generate initial solution) Set bounds of SVR's parameters (C , σ and ϵ). Then, generate randomly initial values of the three parameters.

Step 2 (Initialize cooling schedule) Choose $T > 0$ to be the "initial temperature", and identify the length of each Markov chain (L_k) and it must be greater than 0.

Step 3 (Generate new solution) Make a simple move in order to change the existing solution to a new solution. Another set of the three new parameters are generated in this new solution.

Step 4 (Accept or Reject) The criteria of accepting or rejecting the new solutions is introduced as follows (Van Laarhoven PJM, Aarts EHL, 1987):

If $E(s_{new}) \leq E(s_{old})$, the new solution will be accepted. $E(s_{new})$ and $E(s_{old})$ are the energy of new state and old state, respectively.

If $E(s_{new}) > E(s_{old})$, and $p < P(\text{accept } s_{new})$, $0 \leq p \leq 1$, the new solution will be accepted. Where p is a random number between 0 and 1; $P(\text{accept } s_{new})$ is given as follows:

$$P(\text{accept } s_{new}) = \exp\left(-\frac{E(s_{old}) - E(s_{new})}{kT}\right) \quad (12)$$

Where k is Boltzmann constant (Cercignani C, 1988), T is the current temperature.

Otherwise, the new solution will be rejected.

Step 5 (Loop) If the new solution is not accepted, then return to Step 3. Furthermore, if the current solution is better than the optimal solution, then the optimal solution is replaced by the new solution. Otherwise, repeat Steps 3 and 4 until the length of Markov is reduced to 0. Finally, set the current solution as the optimal solution. The length of Markov chain is regarded as the number of loops.

Step 6 (Temperature function). After the new solution is obtained, the temperature is reduced. The new temperature is obtained by Eq. (13)

$$\text{New temperature} = (\text{Current temperature}) \times \rho \quad (13)$$

Where ρ is the parameter of temperature decay scale (Van Laarhoven PJM, Aarts EHL, 1987) and $0.5 < \rho < 0.99$ (Van Laarhoven PJM, Aarts EHL, 1987).

Step 7 (Stop criteria) The final temperature (the lowest temperature) works as the stop criteria. If the final temperature is reached, then stop the algorithm, and the latest solution is an approximate optimal solution. Otherwise, go to Step 3.

4. Performance evaluation

4.1 Preprocessing for experiment

In our experiment, we chose host load, one kind of typical grid resource, as prediction object. For host load prediction, we chose "mystere10000.dat" as benchmark data set. Figure 2 illustrates the scatter of the proposed data set. We took the last 204 items of the data set for our experiment.

It's very important to scale data before applying SVR method on them. Before the SVR was trained, all the data in the database were linearly scaled to fit within the interval (0, 1).

When artificial intelligence technology is applied to the prediction of time series, the number of input nodes critically affects the prediction performance. According to Kuan (Kuan-Yu Chen, Cheng-Hua Wang, 2007), this study experimented with the number 4 for the order of autoregressive terms. Thus, 204 observation values became 200 input patterns. The prior 150 input patterns were employed for the training set to build model; the other 50 input patterns were employed for test set to estimate generalization ability of prediction models.

The simulation of SVR model had been carried out by using the 'Libsvm', a toolbox for support vector machines, which was originally designed by Chang and Lin (C.C. Chang, C.J. Lin, 2001). The experimental results were obtained using a personal computer with Intel Core™ 2 Duo processor @2.8GHz, 2.79GHz and 2 GB RAM.

Some statistical metrics, such as NMSE and R, were used to evaluate the prediction performance of models (Liang Hu, Guosheng Hu, Kuo Tang, Xilong Che, 2009).

4.2 Parameters determination of three models

4.2.1 SA-SVR model

The five-fold cross validation technique and SA are applied for searching optimal parameters. The optimal parameters are obtained when the MSE of five-fold cross validation is at its minimum.

Table 1 gives an overview of SA parameter settings. According to (Van Laarhoven PJM, Aarts EHL, 1987), the converged solution is mostly affected by parameter settings. In this study, the choices of other parameters are based on numerous experiments, as those values provide the smallest MSE_{cv} on the training set.

According to Wang (L.P.Wang, 2005) and the convenience of computing, we set the parameters searching space: $C (0,200)$, $\sigma (0,200)$ and $\varepsilon (0,1)$;

4.2.2 T-SVR model

The traditional parameter selection procedure of SVR is the trial-and-error procedure, namely T-SVR model. T-SVR model uses the same training set and test set as SA-SVR and have the same parameters searching space: $C (0,200)$, $\sigma (0,200)$ and $\varepsilon (0,1)$ in our experiment.

Considering precision and computing time, we picked 50 discrete points equally from the searching space of C , 50 from σ and 50 from ε . Hence, we got 125000 ($125000=50 \times 50 \times 50$) group of parameters. Cross validation technique was also applied to trial-and-error procedure. The optimal parameters that provide the smallest MSE_{cv} on the training set were obtained after each group of parameters was tried.

4.2.3 BPNN model

In the area of time series prediction, the most popular ANN model is the BPNN due to its simple architecture yet powerful problem-solving ability.

The parameters of BPNN in our experiment were set as follows. Hornik et al. (Hornik, K., Stinchcombe, M., & White, H., 1989) suggested that one hidden layer network is sufficient to model any complex system with any desired accuracy. Hence, a standard three-layer network, including one hidden layer, was used in our experiment. The number of nodes for input layer was set to 10, 4 for hidden layer and 1 for output layer. Rumelhart et al. (Rumelhart, E., Hinton, G. E., & Williams, R. J., 1986) suggested using a small learning rate to set the network parameters. Therefore, the learning rate was set to 0.1. The hidden nodes used the tanh (Eq.(14)) transfer function, and the output node used the linear transfer function.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (14)$$

Considering both the accuracy and time-consuming of BPNN model, the convergence criteria used for the training set was a maximum of 500 iterations.

4.3 Experimental results

Firstly, the results of parameters selection are shown. Table 2 compared the parameters selection results of SA-SVR model with those of T-SVR model. Compared with T-SVR model, SA-SVR model spent even less time, but obtained higher precise parameters with smaller MSE during the procedure of parameters selection. It means SA-SVR model outperforms T-SVR model.

***MSE was from validation set when five-fold cross validation technique is applied.**

Thereafter, the prediction models of SA-SVR and T-SVR were built with the selected parameters. After the training procedure, the BPNN prediction model was also built. So the prediction results of different models were compared. From Figure 3, it is clear that smallest deviations between the predicted and actual values are made by SA-SVR model of all the three models. However, the largest deviations are made by BPNN.

In Table 3, the value of NMSE made by SA-SVR model is the smallest. According to Lewis (Lewis, C. D., 1982), we can rate the prediction results made by SA-SVR model to be of highest precision. However, the correlative coefficients (R) from the SA-SVR model are the highest, indicating an extremely high correlation between the

predicted values and the actual values. At the same time, it could also be observed that T-SVR works better than BPNN.

5. Conclusions

Accurate grid resources prediction is crucial for a grid scheduler. A novel SVR model with SA has been applied to predict grid resources. Compared to T-SVR model, the SA-SVR model provides higher prediction precision and spends even less time on parameters selection. It means that SA is applied to SVR's parameters selection successfully. On the other hand, the SA-SVR model provides lower prediction errors than BPNN model. At the same time, T-SVR also worked better than BPNN. The superior performance of SA-SVR model over BPNN approaches is mainly due to the following causes. Firstly, the SVR model have nonlinear mapping capabilities and can easily capture data patterns of grid resources, host load in this study, compared to the BPNN model. Secondly, improper determining of SVR's parameters will cause either over-fitting or under-fitting of a SVR model. In this study, the SA can determine suitable parameters of SVR and improve the prediction performance of the proposed model. Finally, the SA-SVR model performs structural risk minimization (SRM) principle rather than minimizing the training errors. Minimizing the upper bound on the generalization error improves the generalization performance compared to the BPNN model.

The promising results obtained in this study reveal the potential of the SA-SVR model for predicting grid resources. In future, we will study some other advanced searching techniques for suitable parameters selection. Additionally, we will extend SA-SVR prediction method to support other resource types such as bandwidth.

References

- A. Eswaradass, X. H. Sun, M. Wu. (2005). "A neural network based predictive mechanism for available bandwidth", *19th International Parallel and Distributed Processing Symposium*, 2005.
- C.C. Chang, C.J. Lin. (2001). LIBSVM: a library for support vector machines, Software available from <<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>>, 2001.
- Cercignani C. (1988). *The Boltzmann equation and its application*, Springer-Verlag, Berlin, 1988.
- Duan, K., Keerthi, S., & Poo, A. (2001). "Evaluation of simple performance measures for tuning SVM hyper parameters," *Technical report*, National University of Singapore, Singapore, 2001.
- Hornik, K., Stinchcombe, M., & White, H. (1989). "Multilayer feedforward networks are universal approximations," *Neural Networks*, 1989, pp.336-359.
- I. Foster, C. Kesselman. (1999). *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, San Francisco, CA, 1999.
- Kuan-Yu Chen, Cheng-Hua Wang. (2007). "Support vector regression with genetic algorithms in forecasting tourism demand," *Tourism Management*, 2007, pp.215-226.
- L.P.Wang. (2005). *Support Vector Machines: Theory and Application*, Springer, Berlin, 2005.
- Lewis, C. D. (2005). *International and business forecasting methods*, Butterworths, London, 1982.
- Liang Hu, Guosheng Hu, Kuo Tang, Xilong Che. (2009). "Grid Resource Prediction based on Support Vector Regression and Genetic Algorithms", *the 5th International Conference on Natural Computation*, 2009.
- P. Dinda ,D. O'Hallaron. (1999). "Host load prediction using linear models," *The 8th IEEE Symposium on High Performance Distributed Computing*, 1999.
- R. Wolski, N. T. Spring, and J. Hayes. (1999). "The network weather service: A distributed resource performance forecasting service for metacomputing.," *The Journal of Future Generation Computing Systems*, 1999.
- Rumelhart, E., Hinton, G. E., & Williams, R. J. (1986). *Learning internal representations by error propagation in parallel distributed processing*, Cambridge, MA: MIT Press, 1986.
- Smola A, Schölkopf, B. (1998). "A tutorial on support vector regression" *NeuroCOLT Technical Report NC-TR-98-030*, Royal Holloway College University of London, 1998.
- The Department of Computer Science, University of Chicago, Hostload data set, Available at <<http://cs.uchicago.edu/lyang/Load/>>, Jul 19, 2003.
- Van Laarhoven PJM, Aarts EHL. (1987). *Simulated annealing: Theory and applications*, Kluwer Academic Publishers, Dordrecht, 1987.

Vapnik, V. (1998). *Statistical learning theory*, Wiley, New York, 1998.

Z. X. Liu, X. P. Guan, H. H. Wu. (2006). "Bandwidth prediction and congestion control for abr traffic based on neural networks", *ISNN(2)*, 2006, pp.202–207.

Table 1. SA parameter settings

Initial temperature	1000
Markov chain	30
Decay scale	0.95
Final temperature	100

Table 2. Comparison of parameter selection procedure

model	time	optimal(C, σ and ϵ)	MSE ^a
SA-SVR	3min51s	(8.4,0.73,0.07)	0.011
T-SVR	107min45s	(20, 4, 0.08)	0.016

Table 3. Comparison of prediction results

Model	NMSE	R
SA-SVR	0.2040	0.9692
T-SVR	0.2591	0.9684
BPNN	0.3022	0.9671

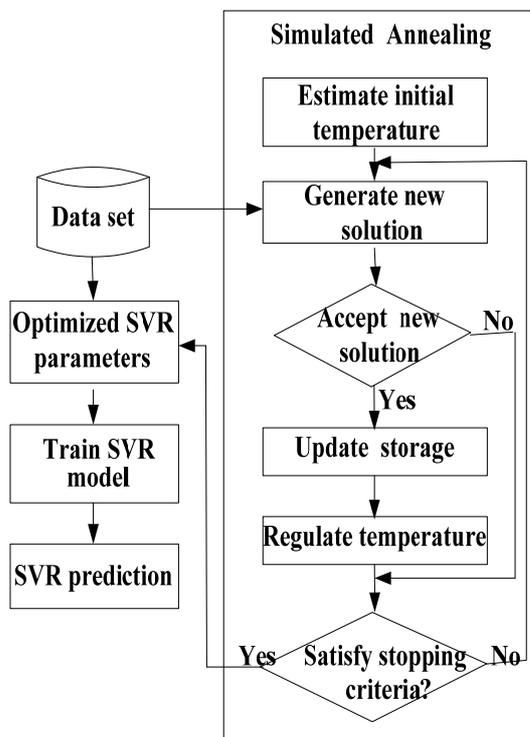


Figure 1. The SA-SVR model

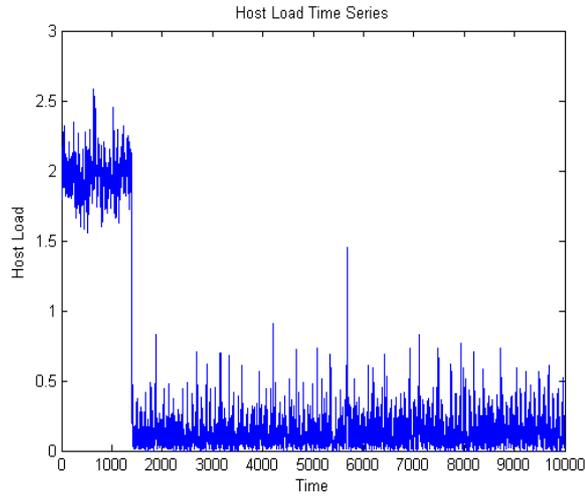


Figure 2. The state of data set

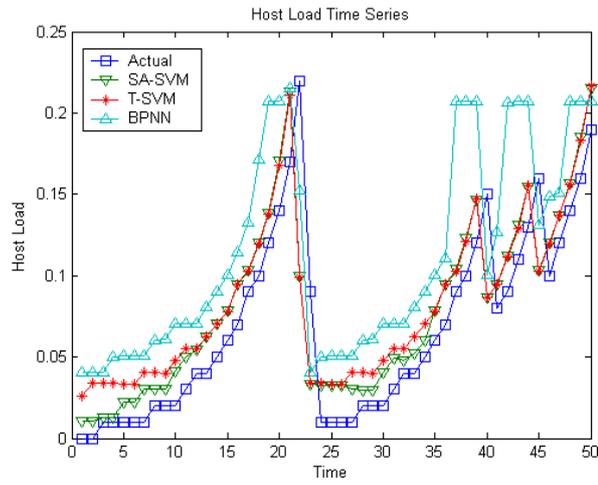


Figure 3. Graphical presentation of different models