technical paper

Evaluation of support vector machine based forecasting tool in electricity price forecasting for Australian national electricity market participants *

DC Sansom, T Downs and TK.Saha

School of Information Technology and Electrical Engineering
University of Queensland

SUMMARY: *In this paper we present an analysis of the results of a study into wholesale (spot) electricity price forecasting utilising Neural Networks (NNs) and Support Vector Machines (SVM). Frequent regulatory changes in electricity markets and the quickly evolving market participant pricing (bidding) strategies cause efficient retraining to be crucial in maintaining the accuracy of electricity price forecasting models. The efficiency of NN and SVM retraining for price forecasting was evaluated using Australian National Electricity Market (NEM), New South Wales regional data over the period from September 1998 to December 1998. The analysis of the results showed that SVMs with one unique solution, produce more consistent forecasting accuracies and so require less time to optimally train than NNs which can result in a solution at any of a large number of local minima. The SVM and NN forecasting accuracies were found to be very similar.*

1   INTRODUCTION

Electrical Supply Industries (ESI) worldwide have been restructured (deregulated) with the intention of introducing levels of competition into energy generation and retail energy sales. In any market with levels of competition information of future market conditions can contribute to giving market participants a competitive advantage over their fellow market participants.

In an open auction style electricity market such as the Australian National Electricity Market (NEM) a large volume of information on historical and predicted market conditions is available to all market participants. As the ESI is a large volume industry all market participants can gain advantages from even a small increase in the accuracy of their electricity price forecasts.

However, maintaining optimum accuracy of a forecasting model requires time and expertise, both of which can be costly to an electricity market participant. In this study we construct, train and test price forecasting models based on Neural Networks

(NNs) and Support Vector Machines (SVMs) with the goal of investigating the following two hypotheses:

1.   SVM require less time and expertise to train than NN.
2.   SVM optimise threshold functions and so should consistently preform forecasting more accurately than NN.

Electricity markets have frequent regulatory changes and quickly evolving market participant pricing strategies so the accuracy of electricity price forecasting models will degrade significantly faster than the accuracy of demand forecasts over time. Keeping price forecasting models with good accuracy requires more frequent and complete retraining than demand forecasting models required. Thus the ease and automation of retraining is crucial in developing a price forecasting tool useful to electricity market participants.

In showing our investigation of these hypotheses this paper has been set out to:

1.   Introduce the need for this research.
2.   Give a brief introduction to NN for electricity demand and price forecasting. NN theory is not discussed in this paper as there are many texts

---

* Paper presented to the Australasian Universities Power Engineering Conference, Melbourne 2002

available on NN theory and operation.[1]
3. SVM structure, operation and theory.
4. Outline of Procedures and methods of producing the required results.
5. Results.
6. Conclusions.

2   NEURAL NETWORK FOR ELECTRICITY DEMAND AND PRICE FORECASTING

Neural Networks (NN) are highly parallel models which have advantages of being flexible and can be used to extract (to learn) complex linear and non-linear relationships from the data. The suitability of NN for forecasting time series problems like electricity demand[2] and price[3,4] is shown in literature available on NN forecasting studies. NNs have two levels of training:

1. The first level of training is to train the NN weights from the training data set. A number of automatic algorithms are available to train the weights, the most commonly used is known as feed-forward back-propagation (bp) algorithm. The bp algorithm is designed to minimise the error through an iterative process which can be visualised as similar to the iterative method of solving a load flow. The drawback of this algorithm is that the initial NN weights are randomised. For the system under study this training algorithm may give the global or any of a number of local minima depending on the randomised initial weights. So in NN parameter optimisation studies the results of a number of forecasts need to be averaged to allow for the random differences in the accuracy of individual forecasts.

2. The second level of training is to optimise the parameters that describe the NN structure. These parameters include the number of hidden layers, the number of neurons in each hidden layer, the momentum, the learning rate and others such as weight decay parameters. The drawback is that optimisation of these parameters is performed by a human trainer expertise, utilising previous studies in the literature and time expensive trial and error methods. There is no commonly accepted algorithm to globally optimise these parameters.

3   SUPPORT VECTOR MACHINE THEORY

With the goal of reducing the time and expertise required to construct and train price forecasting models we considered the next generation of NNs called support vector machines (SVM). SVM have fewer obvious tuneable parameters than NNs and the choice of parameter values may be less crucial for good forecasting results. The SVM is designed to systematically optimise its structure (tune its parameter settings) based on the input training data. The Training of a SVM involves solving a quadratic optimisation which has one unique solution and does not involve the random initialisation of weights as training NN does. So any SVM with the same parameter settings trained on identical data will give identical results. This increases the repeatability of SVM forecasts and so greatly reduces the number of training runs required to find the optimum SVM. parameter settings.

The following explanation of SVM is combination of information from sources[5,6,7] more information regarding SVMs can be obtained from the kernel machines web site.[8]

To explain the principles of SVM we begin with an explanation of the application of a SVM to classify data points as high or low in a two dimensional input space. The basic principal of SVM is to select the support vectors (shaded data points) that describe a threshold function (boundary) for the data that maximises the classification margin (as in figure 1) subject to the constraints that at the support vectors the absolute value of the threshold function must be greater than one as in eq (1) (see figure 2). The non support vector data points (unshaded points) do not effect the position of the boundary.
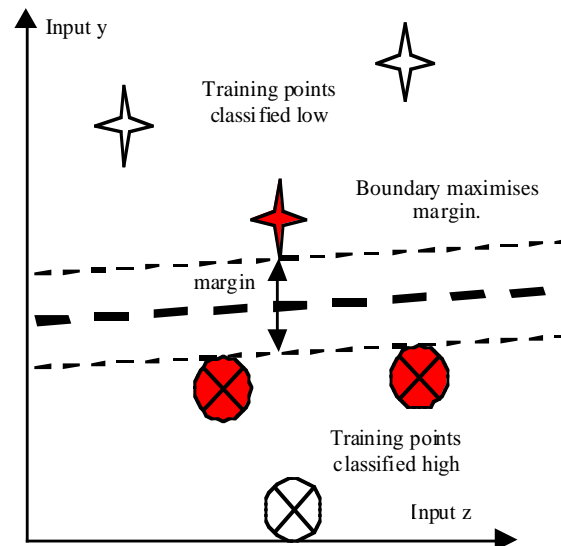

Figure 1:   Maximum margin of support vector machine

Optimisation to minimise margin:

$$\text{minimise} \quad F(W) = \frac{1}{2}(W \bullet W^T)$$

$$\text{subject to} \quad y_k(W \bullet X_k + b) - 1 \geq 0$$

$$\text{for data points} \quad k = 1, \dots, l$$

$$\text{where} \quad y_k \quad \text{is target of data point k}$$
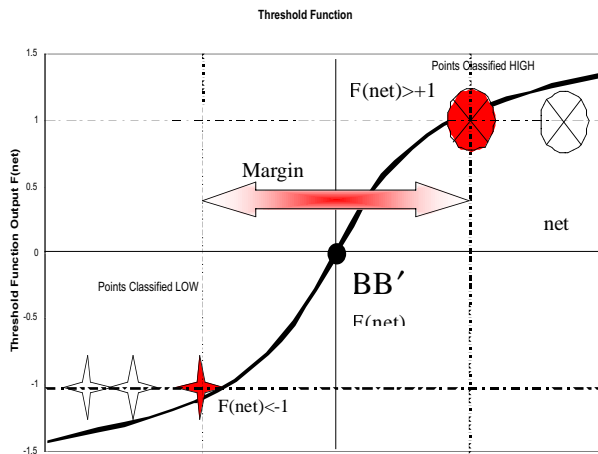
$$\tag{1}$$

Figure 2:   Threshold function for SVM

To overcome the limitation that the SVM only applies to linearly separable systems the inputs (X_k) are mapped through a transform function $\Phi(X)$ into a higher dimensional space where the system is linearly separable. This can be understood with the help of the very simple example in figure 3 where the one-dimensional system is not linearly separable however if the system is mapped by a dot product into two-dimensional space the system becomes linearly separable.
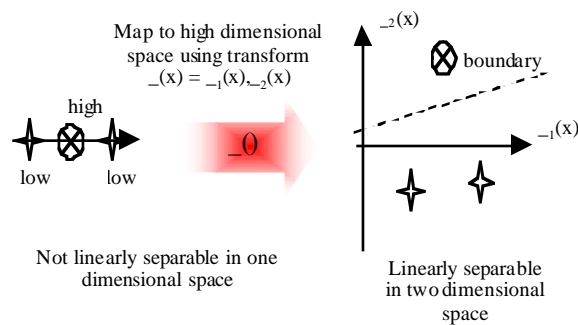

Figure 3:   Example of mapping to higher dimension to make linearly separable

This method of mapping to higher dimensions to make the system linearly separable creates two challenges how to choose a valid mapping transform $\Phi(X)$ and that it may be impractical to perform the dot product required for the margin optimisation in higher dimensional space. To overcome these two challenges a Kernel function is used as shown in eq (2). This Kernel function can implement the dot product between two mapping transforms without needing to know the mapping transform function itself.

Kernel function perform dot product of two mapping functions:

$$K(X_k, X_j) = \Phi(X_k) \bullet \Phi(X_j)$$

$$(2)$$

Once the Kernel function has been included the SVM training can be written as the quadratic optimisation problem in lagrangian multiplier form as:

$$max \quad W(\Lambda) = \Lambda^T \bullet \tilde{1} - \frac{1}{2}\left[\Lambda^T D\Lambda\right] \quad (3)$$

where

$$D_{k,j} = d_k y_j K(x_k, x_j)$$

and the vector of lagrangian multipliers is

$$\Lambda = (\lambda_1, \lambda_2, ..., \lambda_l)$$

Solving this quadratic optimisation gives the vector of lagrangian multipliers (shadow prices). Support vectors are the only data points with non-zero lagrangian multipliers so only support vectors are required to produce a forecasting model (ie. describe the boundary in figure 1).

$$s \text{ support vectors} \quad S_s = X_s \quad only\ if\ \lambda_s \neq 0$$

To produce forecast implement eq (4) below as in figure 4.

Output of SVM:

$$f(X_k) = sign(net(X_k))$$

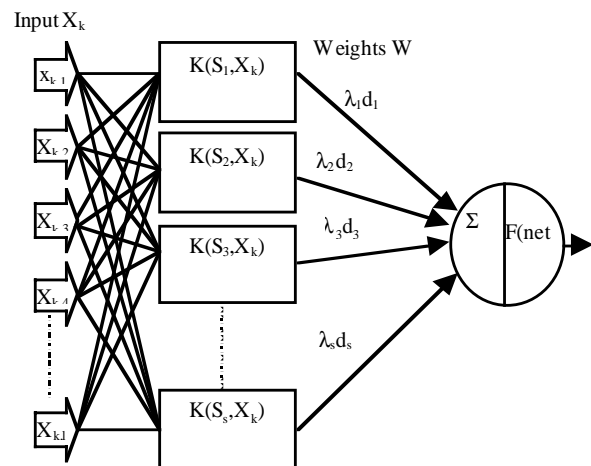$$net(X_k) = \sum_s \lambda_s d_s K(S_s, X_k) + b$$

$$(4)$$

Figure 4:   Structure of SVM

To apply SVM to regression forecasts a slack variable ξ_k is applied for each data point, which allows for an error between the target price y_k and the output of the SVM. The optimisation becomes:

SVM training for regression

$$minimise \quad F(W) = \frac{1}{2}(W \bullet W^T) + C\sum_k \xi_k$$

$$subject\ to \quad y_k(W \bullet X_k + b) - 1 + \xi_k \geq 0$$

$$for\ data\ points\ k = 1, ..., l$$

$$where \quad y_k \quad is\ price\ of\ data\ point\ k \quad (5)$$

C is a parameter chosen by the user to assign penalties to the errors. A large C assigns more penalty to the errors so the SVM is trained to minimise error, can be considered lower generalisation. A small C assigns less penalty to errors so SVM is trained to minimise margin while allowing errors, higher generalisation.

At the two extremes:

*   At C=infinity. No errors are tolerated so the SVM is trained to memorise and so correctly forecast every data point in the training set. Results in a complex model with low generalisation.

*   At very small C. All errors are tolerated which results in a low complexity model (in traditional regression this would be a 'smooth' function y=f(x)) with higher generalisation.

4   PROCEDURE

The SVM training and forecasts were performed with the mySVM program developed by Stefan Rüping.[9] The program was designed to solve the dual of the optimisation in eq (5) by dividing the training set into small working sets or chunks.[10]

All NN and SVM price forecasting models were trained with 90 days of data and tested by forecasting the next seven days of NSW regional electricity price. Nine weeks of price forecasts were carried out (see table 1). This data was obtained from the SEM (NSW State Electricity Market) a forerunner of the NEM (data now available on the NEMMCO web site).

Table 1
Dates of training and test data sets

Week	Training files 1998			Testing files 1998		
	from	to	days	from	to	days
1	9/07	6/10	90	7/10	13/10	7
2	16/7	13/10	90	14/10	20/10	7
3	23/7	20/10	90	21/10	27/10	7
4	30/7	27/10	90	28/10	3/11	7
5	06/8	03/11	90	04/11	10/11	7
6	13/8	10/11	90	11/11	17/11	7
7	20/8	17/11	90	18/11	24/11	7
8	27/8	24/11	90	25/11	01/12	7
9	03/9	01/12	90	02/12	08/12	7

All forecasting models utilised the same set of 11 input variables.

Table 2
Inputs variables

Inputs to NN and SVM			
Input number	Input Name	Half-hour delay. t=0 time of forecast	Comment
Target	Pool Price	t=0	Cents/MW
1	Half-hour of week	t=0	1-336
2	Half-hour of day	t=0	1-48
3	Regional Demand in MW	t=0	Actual Demand
4	Pool Price	t= 383	8 days
5	Pool Price	t=384	8 days
6	Pool Price	t=385	8 days
7	Pool Price	t=671	2 weeks
8	Pool Price	t=672	2 weeks
9	Pool Price	t=673	2 weeks
10	Pool Price	t=1008	3 weeks
11	Pool Price	t=1344	4 weeks

The scope of this research was limited to older market data from 1998 and a standard set of 11 input variables to enable researchers and the readers to make comparisons with previous studies.[11] Investigation into the data available on the NEMMCO web site has shown more informative data variables are now available however these were not used in this study to maintain consistency with previous studies.

5   RESULTS

5.1   Neural network training

The results (table 3) are shown from an investigation into the variation in NN forecasts due to the random initialisation of weights and the variation of accuracy attributed to network architecture. The architecture is defined by the code 10,5 which is ten neurons in the first hidden layer and five neurons in the second hidden layer. Each set of results is from two NN first and repeat, which were trained with identical inputs and parameter settings. Thus the only difference between the two networks is the value of the randomly initialised weights. The networks were trained with 56 days and tested over 21 days of data. The data was divided into five partitions based on the 336 half-hours of the week (only results for the first two partitions shown): partition 1 1-48, partition 2 49-96.
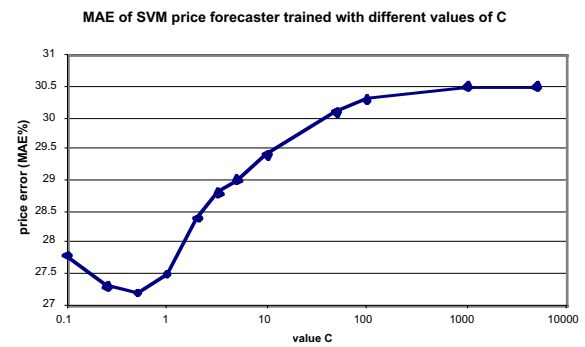
Table 3
Results MAE% of NN identical runs

architecture	Partition 1		Partition 2	
	first	repeat	first	repeat
8,5	31.1	30.4	38.7	30.0
10,5	26.7	25.7	39.7	26.0
25,5	29.6	30.3	31.5	34.5
100,10	28.1	30.1	26.9	40.7
60,5	28.7	29.2	28.8	22.9
40,20,15,10	28.6	28.8	37.0	32.2
100,5	27.3	29.3	34.2	36.3
10	30.5	31.9	25.6	18.1

As a researcher attempting second level training to optimise the parameters of the NN forecasting tools the random variation of forecasting accuracy is a frustrating challenge. More importantly this variation in accuracy reduces the confidence of a customer using any forecasting tool . Results show that the training of a SVM produces the same network and so the same accuracy given identical input data and training parameters. Results of repeated SVM training are an identical copy of the results displayed in table 4. This consistent accuracy of SVM is a great advantage in training price forecasting tools and convincing customers to have confidence in the forecasting tool.

5.2   Variation of C parameter in SVM training

The results in figure 5 were from testing to investigate the change in the accuracy of a SVM price forecaster trained on identical data and all parameters except different values of C.

Optimising C in SVM training has an influence on the generalisation of the SVM and can be considered similar to optimising the architecture of a NN. In this research the values of C was varied between 0.1 to 5000 with limited change in forecasting accuracy. The consistent accuracy of SVM forecasts makes optimising C faster and simpler than optimising the architecture of a NN. As the results in table 3 show, in training NN the random variation is often equal or larger than the accuracy difference due to changes in architecture and so the average of many training runs must be taken. This greatly increases the time required to optimise the training of a NN. From these results a C of 0.5 was used for all remaining SVM training.

Figure 5:   SVM trained with different C values

5.3   Accuracy of SVM forecasts

The errors Mean Absolute Error (MAE) of SVM price forecasts are shown in table 4. Each SVM was trained first level only (all parameters held constant) with a 90 day training set (see table 1) and then tested over the remainder of the nine week testing period. Shaded cells are showing results for back-casts (SVM is 'forecasting' data it has seen in the training set) and are not included in the calculation of the total MAE.

Table 4
MAE of SVM results for 7 day ahead
price forecasts

training	Tested with test set from week									Total
week	1	2	3	4	5	6	7	8	9	error
Data used	MAE									MAE
1	24%	21%	35%	39%	31%	24%	22%	18%	18%	26%
2	8%	21%	37%	45%	42%	35%	31%	26%	20%	32%
3	8%	5%	38%	47%	42%	38%	35%	29%	21%	36%
4	8%	5%	9%	44%	40%	36%	32%	27%	20%	33%
5	9%	5%	8%	12%	37%	33%	28%	23%	19%	28%
6	10%	5%	7%	11%	9%	25%	24%	20%	19%	22%
7	11%	6%	7%	10%	8%	8%	20%	17%	19%	18%
8	11%	6%	7%	10%	9%	8%	7%	16%	20%	18%
9	11%	6%	7%	10%	9%	8%	8%	7%	20%	20%

The error of the SVM price forecasts over nine weeks was MAE= 25.8% which was very similar to the accuracy of the NN price forecasting model which had an error of MAE=25.5%. The training data sets were optimised for NN training over a number of months so it is possible that optimising the training length and data variables for the SVM may increase the accuracy of the SVM forecast.

Both the NN and SVM models produced more accurate price forecasts for weeks 1,2,7,8 and 9 than

for weeks 3,4,5 and 6. Thus it can be concluded that both models extracted similar patterns from the data. Visual inspection shows weeks 3,4,5 and 6 included price spikes which helps explains the lower accuracy during these weeks.

6   CONCLUSIONS

From a limited analysis of NN and SVM price forecasting results we make the following conclusions:

1. SVM require less time to optimally train than NN. This is mainly because SVM are trained with a structured algorithm (quadratic optimisation), which has one unique solution and so consistently produces the same results when trained with identical data and parameters. This consistency saves time as the average of multiple training runs are not required to optimise the parameter settings. It is questionable if SVM require less expert knowledge to train than NN however the consistency of SVM results makes programming automatic optimisation algorithms simpler and less time consuming than for NN optimisation.

2. For this study the SVM price forecaster performed with an equivalent accuracy to the NN price forecaster. However to help in comparison the training data sets which were optimised for NN were used to train the SVM so accuracy maybe improved if data is optimised for SVM.

These results did not lead to concrete conclusions but raised a number of questions to be pursued in future research:

1.   Investigation of developing an automatic SVM parameter optimisation program for electricity price forecasting. An over all optimisation algorithm may be impractical but limiting the optimisation to only a SVM price forecasting model in one particular market regulatory framework is worth to investigate.

2. Test to see if optimising the training data set for SVM increases the accuracy of the forecast.

3. Investigate the validity of forecasting models over a longer time period to access the ability of forecasting models to cope with electricity market evolution.

4. Investigate if the unique SVM solution is at or close to the optimum forecasting solution for NEM electricity price forecasting.

Considering that NN and SVM produce similar forecasting accuracies and that SVM are more consistent and so more efficient in retraining we will be favouring SVM over NN in future price and load forecasting studies.

In this paper we have presented the latest results in an ongoing University of Queensland research project to develop an electricity price forecasting tool for electricity market participants, which utilises publicly available historical and predicted market condition information.

REFERENCES

1. Wasserman PD. Neural computing theory and practice, 1989.

2. H.S Hippert. Neural networks for short-term load forecasting: a review and evaluation. IEEE Transactions on Power Systems, 2001;16:44-55.

3. Sapeluk A. Pool price forecasting: a neural network application. UPEC 94 Conference paper, 1994;2:840.

4. Ramsay B. A neural network for predicting system marginal price in the UK power pool. University of Dundee UK, 1995.

5. Burges C. A tutorial on support vector machines for pattern recognition: Kluwer, 1998.

6. Cortes C. Support-vector networks. Machine Learning, 1995;20:73-97.

7. Platt-JC. Sequential minimal optimization: a fast algorithm for training support vector machines. Microsoft research report, 1998.

8. http://www.kernel-machines.org/.(2002)

9. RüpingS. mySVM software. http:// www.kernel-machines.org/ under software -> mySVM. (accessed 2002)

10. Osuna E, Freund R, Girosi F. Support vector machines: training and applications: Massachusetts Institute of Technology, 1997.

11. Sansom D,Saha TK. Neural networks for forecasting electricity pool price in a deregulated electricity supply industry. Proc. AUPEC'99 Darwin Australia, 1999:214.

Damien Sansom email Sansom@itee.ue.edu.au (current April 2003)
Degree of Electrical and Electronic engineering from James Cook University of North Queensland 1995.
Masters of Electrical Engineering from University of Queensland 2000.
Currently (2003) completing PhD at University of Queensland concentrating on developing tools for electricity markets.

TAPAN SAHA

Tapan Kumar Saha was born in Bangladesh and came to Australia in 1989. Dr Saha is a Senior Lecturer in the School of Information Technology and Electrical Engineering, University of Queensland, Australia. Before joining the University of Queensland in 1996 he taught at the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh for three and a half years and at James Cook University, Townsville, Australia for two and a half years. He is a senior member of the IEEE and Fellow of the Institution of Engineers, Australia. His research interests include power systems, power quality and condition monitoring of electrical plants.