

## Forecasting earning of VLCC tankers using artificial neural networks

Young-Jun Jung<sup>1</sup> · Sung-hee Kang<sup>2</sup> · Deog-hee Doh<sup>†</sup>

(Received November 7, 2018 ; Revised December 1, 2018 ; Accepted December 18, 2018)

**Abstract:** Shipping companies consistently want to minimize losses and maximize profits by accurately predicting the direction and magnitude of fluctuations in maritime situations. However, the determinants of maritime markets are diverse and volatile, and the decision mechanism is complex. Accurately predicting the direction and magnitude of fluctuations remains a difficult challenge. This study was conducted to evaluate the accuracy of multi-step-ahead forecasting of VLCC tanker markets by using an artificial neural network (ANN) training algorithm with the Levenberg-Marquardt and Bayesian regularization algorithms. The advanced time for forecasting was divided into one, three, six, nine, 12, and 15 months. The ANN predictions were conducted on the earnings for VLCC markets, and the datasets of the variables used in the forecast were 204 monthly time-series data from January 2000 to December 2016.

**Keywords:** Artificial neural networks, VLCC, Forecasting

### 1. Introduction

The world crude oil production in 2016 was 4448 million tons [1], and the total amount of crude oil transported by sea was 1949 million tons, which accounted for 43.8% of total crude oil production. The crude tanker demand for world seaborne trade was 178.5 million dwt for VLCC (200,000 dwt and over), 56.8 million dwt for SUEZMAX (125–199,999 dwt), and 55.5 million dwt for AFRAMAX (85–124,999 dwt) [2]. However, the oil tanker markets, which account for a large portion of world maritime transport, is highly influenced by the interaction of supply and demand in tanker transportation services and is highly volatile [3]. Therefore, predicting the changes in these markets is critical for all stakeholders, particularly those of the tanker market demand and supply sides.

This study focuses on predicting ANN for earnings for VLCC, which plays a major role in the marine transportation of crude oil. In addition, we evaluate the accuracy of an ANN prediction model when used with the Levenberg–Marquardt and Bayesian regularization algorithms.

Operating earnings can be derived from the time-charter rates or time-charter equivalent of spot rates when a vessel is

operating in the spot market. Earnings are more representative of what an operating tanker produces [4]. Therefore, when choosing an ANN prediction target for the VLCC market, we selected the VLCC market earnings (USD/day) instead of the freight index (world scale, or WS).

### 2. Training algorithm of ANN

#### 2.1 Mean squared error of ANN training algorithm

Learning is the fundamental capability of neural networks. Supervised learning adjusts network parameters by directly comparing the desired and actual network output. Supervised learning is a closed-loop feedback system in which the error is the feedback signal. The error measure, which indicates the difference between the output from the network and from training samples, is used to guide the learning process. The error measure is usually defined by the mean squared error (MSE) [5].

$$\text{MSE} = \frac{1}{Q} \sum_{q=1}^Q (t_q - a_q)^2 \quad (1)$$

where  $Q$  is the number of pattern pairs in the sample set,  $t_q$  is the output part of the  $q$ th pattern pair, and  $a_q$  is the network

<sup>†</sup> Corresponding Author (ORCID: <http://orcid.org/0000-0001-6093-2975>): Professor, Division of Mechanical Engineering, Korea Maritime and Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: doh@kmou.ac.kr, Tel: 051-410-4364

<sup>1</sup> Visiting Professor, Division of Mechanical Engineering, Korea Maritime and Ocean University, E-mail: yjjung8619@kmou.ac.kr, Tel: 051-410-4675

<sup>2</sup> Chief Engineer, LG Electronics, E-mail: kebin.kang@lge.com, Tel: 055-717-1780

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

output corresponding to the pattern pair  $q$ . The MSE is calculated anew after each epoch. The learning process is terminated when MSE is sufficiently small or the failure criterion is met.

### 2.2 Levenberg–Marquardt algorithm

The Levenberg–Marquardt algorithm, which is a numerical optimization technique [6], was designed to minimize functions that are sums of squares of other nonlinear functions in neural network training, where the performance index is the MSE.

When the performance index is  $F(\mathbf{x})$ , the Levenberg–Marquardt algorithm for optimizing the performance index  $F(\mathbf{x})$  is represented as [7]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [J^T(\mathbf{x}_k)J(\mathbf{x}_k) + \mu_k]^{-1}J^T(\mathbf{x}_k)V(\mathbf{x}_k) \quad (2)$$

Here, as the changing value of  $\mu_k$ , the performance index  $F(\mathbf{x})$  of the network can be adjusted through the optimization algorithms with a small learning rate, where  $J(\mathbf{x}_k)$  and  $V(\mathbf{x}_k)$  are the matrix elements used to compute the gradient.

### 2.3 Bayesian regularization algorithm

The complexity of a neural network is determined by the number of free parameters of weights and biases, which is determined by the number of neurons. If the network is too complex for a given dataset, then it is likely to overfit and have poor generalization [8]. The simplest method for improving generalization is early stopping. Another method is known as regularization [9]. One of two possible approaches can be used to improve the generalization capability of a neural network: restricting either the number of weights or the magnitude of weights, where the latter is called regularization.

The Bayesian regularization algorithm can be written as the sum of squares of network weights as follows:

$$F(\mathbf{x}) = \beta E_D + \alpha E_W = \beta \sum_{q=1}^Q (\mathbf{t}_q - \mathbf{a}_q)^T (\mathbf{t}_q - \mathbf{a}_q) + \alpha \sum_{i=1}^n \mathbf{x}_i^2 \quad (3)$$

where  $F(\mathbf{x})$  is the regularized performance index. Here, the regularization ratio  $\alpha/\beta$  controls the effective complexity of the network solution.

## 3. Methodology

### 3.1 Data collection

The tanker freight market is characterized by interaction between many determinants of supply and demand in tanker

transportation services [10]. To forecast the dynamics and fluctuations of freight rates in tanker freight markets, considerable research has been conducted using univariate or multivariate time-series analytical techniques [11] and ANN models [12].

For all data used in forecasting of oil tanker markets in this study, global oil production, world GDP, active fleets, new building prices, second-hand ship prices, demolition prices, time-charter rates, bunker prices, and crude oil prices were selected as independent variables, whereas dirty tanker earnings was selected as the dependent variable.

Thus, the aggregated data were composed of nine independent variables and one dependent variable, where each variable was based on 204 monthly observations from January 2000 to December 2016.

### 3.2 Identification of ANN architecture

After the data for forecasting tanker markets were collected, the type of ANN architecture used to solve the problem of tanker market prediction, as well as the number of neurons and layers used in the network, were all determined. In ANN dynamic networks, the output depends not only on the current input to the network but also on previous inputs, outputs, and/or states of the network. Tanker markets prediction is part of an analysis that predicts the future value of a time series. Therefore, in this study, we selected dynamic networks as an appropriate ANN model to forecast dirty tanker markets. The non-linear autoregressive model with exogenous inputs (NARX networks) [5][8], which is a widely used network for applying predictions, is a recurrent dynamic network with feedback connections that encompass multiple layers of the network. This is shown in **Figure 1**.

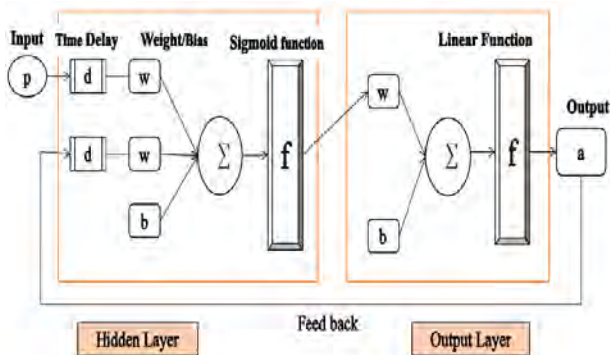
Until now, prediction using ANN for VLCC tanker markets has been performed using the Levenberg–Marquardt training algorithm [12]–[15]. However, no Bayesian regularization algorithm has been used to predict earnings for such markets. Therefore, this study focuses on using ANN for prediction of earnings for VLCC tanker markets and evaluates the accuracy of the ANN prediction model using the Levenberg–Marquardt and Bayesian regularization algorithms.

After we identified the network structure, the number of hidden layers in these two learning algorithms was determined to enable us to compare performance results and functions more easily.

To determine the number of neurons in the hidden layer to identify the best prediction performance for the VLCC tanker

market without overfitting, we adjusted the number of neurons in the hidden layer of the ANN structure using the Levenberg-Marquardt algorithm. In addition, using the Bayesian regularization algorithm, we had the ANN perform prediction for two cases using eight and 10 neurons of the hidden layer, respectively, based on nine total input variable numbers [7]. We then evaluated the performance results.

Only one neuron was present in the output layer, which had the same size as that of the target.



**Figure 1:** NARX network (closed loop) for VLCC tanker market prediction

The prediction for VLCC markets earnings with the advanced periods of one, three, six, nine, 12, and 15 steps (months) were performed using MATLAB with the neural network toolbox. The inputs of the testing ANN model had nine nodes for input signals. In addition, the hidden layer was composed of neurons, with the tan-sigmoid transfer function selected as the neuron activation function. The value of the tagged delay line as a time delay was adjustable to avoid producing correlation effects.

### 3.3 Post-training validation

As an essential tool for neural network validation, the regression coefficient between the network output and target, known as the R value, should approximate 1 to ensure reliable ANN performance results. In addition, when applying dynamic networks for prediction, such as the focused time-delay neural network, two crucial factors must be considered when analyzing the trained prediction network: prediction errors should not be correlated either in time or with the input sequence.

## 4. Implementation

### 4.1 Data processing

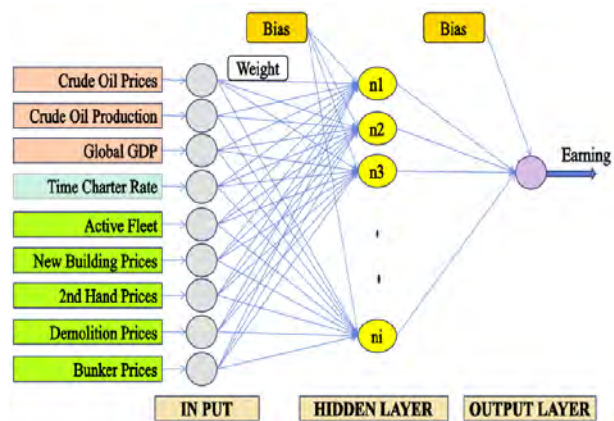
The previously mentioned 204 data points of each variable in the period from January 2000 to December 2016 were randomly

sampled during computation and divided into three datasets for training, validation, and testing. When the Levenberg-Marquardt algorithm was applied for iterative computing, the training set comprised approximately 70% of the full dataset, with the validation and test datasets each consisting of approximately 15%. In addition, when the Bayesian regularization training technique was applied, the testing data set was assigned as only 15% of the full dataset because the validation sequence was not applied to the algorithm.

### 4.2 ANN model for VLCC market prediction

A schematic of the ANN network for tanker market prediction is shown in **Figure 2**. When the Levenberg-Marquardt algorithm was applied, the number of neurons in the hidden layer was adjusted to improve the accuracy of the prediction performance. In addition, when the Bayesian regularization algorithm was applied, the number of neurons in the hidden layer was fixed to eight and 10 to compare the performance results of these two cases with those from the Levenberg-Marquardt algorithm.

The number of neurons in the output layer was the same as the size of the target. The output layer was composed of one neuron with the linear function as its activation function.



**Figure 2:** ANN schematic of NARX network for VLCC tanker market prediction

### 4.3 Computation

Each implementation for prediction was repeated several times to identify the optimal parameters and conditions of the network. When the results were unsatisfactory, training was repeated after weights and biases were initialized. When the Levenberg-Marquardt algorithm was trained, the number of neurons was adjusted to prevent overfitting or extrapolation.

The value of the tagged delay line as a time delay was 2 (months) without any change made during implementation. The computational results from the Levenberg–Marquardt algorithm were considered as reasonable when the algorithm was fitted with the following considerations:

- The final mean performance index (MSE) was small
- The test set error (test performance index (MSE)) and validation set error (validation performance index (MSE)) had similar characteristics
- Expert judgment was used in determining various parameters and performance indices

When the Bayesian regularization technique was trained, 221 parameters in the 9-10-1 (input to number of neurons of the hidden layer to output) network, and 177 parameters in the 9-8-1 network were employed. The effective number of parameters was a minimum of 27 and a maximum of 172 when the 9-10-1 network was trained. The training of the 9-10-1 network effectively used less than 77% of the total number of weights and biases. The computational results for the training of the Bayesian regularization algorithm were considered as reasonable when the following were considered:

- The final mean performance index (MSE) was small
- The training error (training performance index (MSE)) was small
- Expert judgment was used in determining various parameters and performance indices

The computer used for the calculation had an Intel® Core™ i5-5200U CPU @ 2.20 GHz.

#### 4.4 Validation

The regression plots display the network outputs with respect to targets for training, validation, and test datasets. For this problem, the fit was reasonably good for all datasets, The validation for this problem was satisfactory for all data sets in each case with an R value of at least 0.93. The autocorrelation function of the prediction error and the cross-correlation function to measure the correlation between the input and prediction error were used during ANN prediction model validation.

## 5. Results

### 5.1 Comparison of prediction performance for VLCC

All implementation results of the mean performance index (MSE) are presented in **Table 2 - Table 7** and **Figure 3 - Figure 8** for use in evaluating the accuracy of the ANN prediction model.

#### 5.1.1 One-month-ahead prediction

**Figure 3** shows that BRA.TDS-15.NN-10 network had relatively good convergence with the total progression of the observation values. The networks of BRA.TDS-15.NN-10 and BRA.TDS-15.NN-8 showed better convergence-to-target values than did LMA.TDS-15.NN-9. The mean performance indices of the BRA.TDS-15.NN-8 and BRA.TDS-15.NN-10 networks were nearly the same. However, the training performance error of the BRA.TDS-15.NN-10 network was 0.266, which was much lower than that of the BRA.TDS-15.NN-8 network (1.3874), as shown in **Table 2**.

**Table 1** shows the performance results for one-month-ahead prediction using the Levenberg–Marquardt algorithm for different numbers of neurons in the hidden layers. Each case shows best performance results without overfitting. When the number of neurons was greater or less than the number of input variables of 9, the results did not change satisfactorily. Note that during testing, the magnitude of the  $\mu$  value was changed from 0.01 to 0.1 to speed up the conjugation.

**Table 1:** Comparison of one-month-ahead predictions based on different neuron numbers with the Levenberg-Marquardt algorithm

[ One-month-ahead Prediction ]	ANN (Levenberg-Marquardt algorithm)			
	NN-7	NN-8	NN-9	NN-10
<b>Epoch</b>	11	11	12	15
<b><math>\mu</math></b>	0.1	0.01	0.1	0.01
<b>Gradient</b>	8.11	19.9	8.68	9.6
<b>Regression</b>	0.92432	0.93793	0.96639	0.94919
<b>Mean Performance Error</b>	28.2745	24.9203	12.2177	18.7857
<b>Train Performance Error</b>	10.1344	16.0031	9.2426	6.7629
<b>Validation Performance Error</b>	73.9551	44.9173	17.0812	62.3482
<b>Test Performance Error</b>	68.4571	47.1315	21.4366	32.1312

NN-: Number of neurons in the hidden layer

(Unit of Performance Error: x 1000 USD/Day)

#### 5.1.2 Three-months-ahead prediction

**Table 3** shows that with the Bayesian regularization algorithm, as the size of the hidden layer neuron increased, both the number of iterations and computing time increased, and the gradient value and train performance error (MSE) converged to

a smaller value. However, even though the size of the hidden layer neuron increased in the short-term-ahead prediction, the mean performance index of 8.6425 for eight neurons showed significantly better results than 8.9288 for 10 neurons in the same forecasting horizon.

5.1.3 Six-months-ahead prediction

Table 4 shows that even though the number of hidden layer neurons increased from eight to 10 neurons, the gradient value and training performance error (MSE) did not converge to a smaller value. The training, validation, and test performance errors with the Levenberg–Marquardt algorithm showed a significant error in size despite no indication of overfitting or extrapolation.

5.1.4 Nine-months-ahead prediction

Figure 6 shows that the BRA.TDS-15.NN-8 network had relatively good convergence with the total progression of the observation values. However, during sudden up and down changes in the market, unstable prediction also appeared with the 6-months-ahead prediction. The training, validation, and test performance errors with the Levenberg–Marquardt algorithm showed significant errors in size despite no indication of overfitting or extrapolation.

5.1.5 12-months-ahead prediction

Table 6 shows that the mean performance index of 4.9389 with 10 neurons in 12-months-ahead prediction had better results than did the index of 9.3915 with eight neurons and as compared to the other training algorithm in the same forecasting horizon. As shown in the table, in the long-term-ahead forecasting, such as in 12-months-ahead prediction, the training algorithm, which had a larger hidden layer neuron, exhibited better forecasting performance than with the smaller neuron. The Levenberg–Marquardt algorithm showed a significant error in size despite no indication of overfitting or extrapolation.

5.1.6 15-months-ahead prediction

Table 7 shows that the mean performance index of 8.243 for 10 neurons with the Bayesian regularization algorithm generated better results than did the index of 10.5656 for eight neurons and as compared to the other training algorithm in the same forecasting horizon. As shown, similar performance results occurred with the 12-months-ahead prediction as with other long-term-ahead forecasting when the training algorithm was applied, where larger neurons exhibited better forecasting performance than did the smaller.

Table 2: Comparison for one-month-ahead prediction of VLCC

[ One-month-ahead Prediction ]	ANN Architecture		
	BRA .TDS-15. NN-8	BRA .TDS-15. NN-10	LMA .TDS-15. NN-9
Epoch /computing time(sec)	279/5	443/9	12
$\mu$	-	-	0.1
Gradient	0.82	0.329	8.68
Effective number of Parameter(used/total)	120/177	147/221	-
Regression	0.98218	0.98175	0.96639
Mean Performance Error	6.7013	6.7026	12.2177
Train Performance Error	1.3874	0.266	9.2426
Validation Performance Error	-	-	17.0812
Test Performance Error	37.168	43.6063	21.4366

(Unit of Performance Error: x 1000 USD/Day)

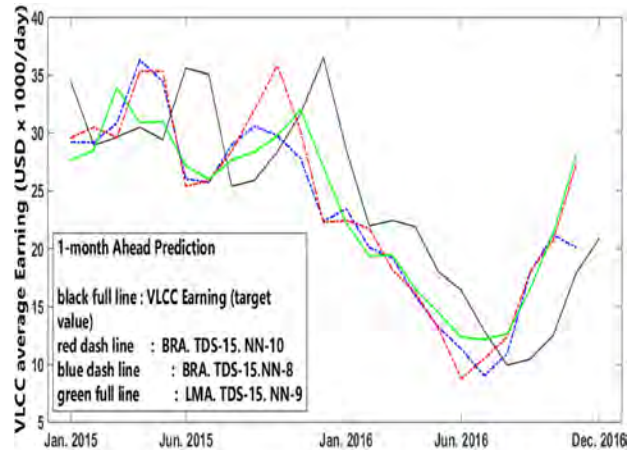


Figure 3: One-month-ahead prediction of VLCC

Table 3: Comparison for 3-months-ahead prediction of VLCC

[ 3-months-ahead Prediction ]	ANN Architecture		
	BRA .TDS-15. NN-8	BRA .TDS-15. NN-10	LMA .TDS-15. NN-9
Epoch /computing time(sec)	370/6	707/13	12
$\mu$	-	-	0.1
Gradient	1.0	1.23	11.3
Effective number of Parameter(used/total)	114/177	121/221	-
Regression	0.97649	0.97605	0.93509
Mean Performance Error	8.6425	8.9288	23.8859
Train Performance Error	2.2021	1.9989	10.3963
Validation Performance Error	-	-	58.8623
Test Performance Error	46.6056	48.6599	54.7607

(Unit of Performance Error: x 1000 USD/Day)

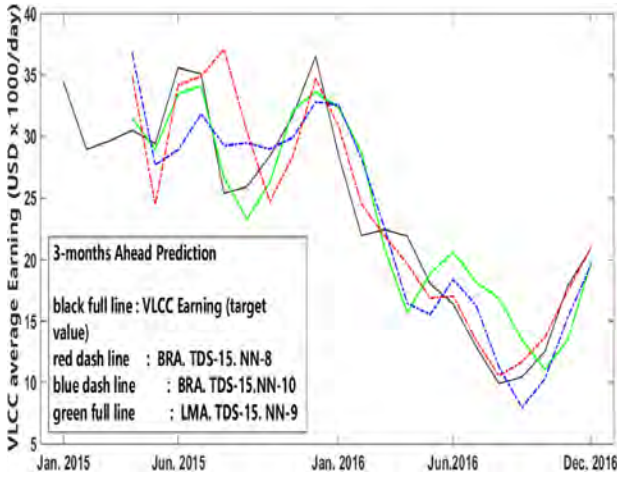


Figure 4: 3-months-ahead prediction of VLCC

Table 4: Comparison for 6-months-ahead prediction of VLCC

[ 6-months-ahead Prediction ]	ANN Architecture		
	BRA .TDS -15. NN-8	BRA .TDS -15. NN-10	LMA .TDS -15. NN-8
Epoch /computing time(sec)	347/6	707/15	17
$\mu$	-	-	0.001
Gradient	0.999	1.23	37.8
Effective number of Parameter(used/total)	110/177	121/221	-
Regression	0.97983	0.97605	0.92713
Mean Performance Error	7.8063	8.9288	27.5454
Train Performance Error	2.0636	1.9989	15.7874
Validation Performance Error	-	-	56.4943
Test Performance Error	40.7316	48.6599	54.2511

(Unit of Performance Error: x 1000 USD/Day)

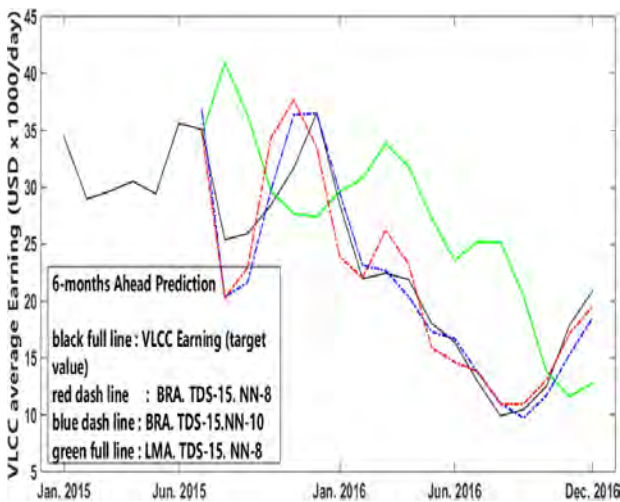


Figure 5: 6-months-ahead prediction of VLCC

Table 5: Comparison for 9-months-ahead prediction of VLCC

[ 9-months-ahead Prediction ]	ANN Architecture		
	BRA .TDS -15. NN-8	BRA .TDS -15. NN-10	LMA .TDS -15. NN-9
Epoch /computing time(sec)	279/4	247/5	15
$\mu$	-	-	0.01
Gradient	0.82	0.602	99.6
Effective number of Parameter(used/total)	120/177	136/221	-
Regression	0.9828	0.97937	0.94738
Mean Performance Error	6.7013	7.6663	19.2948
Train Performance Error	1.3874	0.6623	7.5927
Validation Performance Error	-	-	44.3444
Test Performance Error	37.168	47.8225	49.6353

(Unit of Performance Error: x 1000 USD/Day)

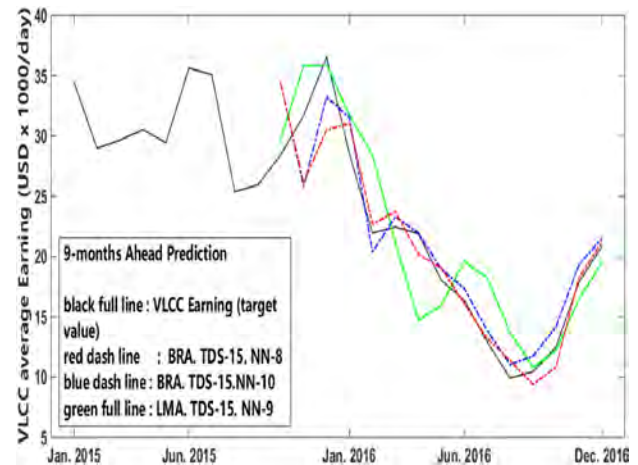


Figure 6: 9-months-ahead prediction of VLCC

Table 6: Comparison for 12-months-ahead prediction of VLCC

[ 12-months-ahead Prediction ]	ANN Architecture		
	BRA .TDS -15. NN-8	BRA .TDS -15. NN-10	LMA .TDS -15. NN-9
Epoch /computing time(sec)	310/5	639/13	15
$\mu$	-	-	0.1
Gradient	0.553	0.397	4.95
Effective number of Parameter(used/total)	124/177	146/221	-
Regression	0.97443	0.98681	0.95092
Mean Performance Error	9.3915	4.9389	17.8695
Train Performance Error	0.8942	0.3323	6.6971
Validation Performance Error	-	-	39.4219
Test Performance Error	58.1091	31.3500	49.1998

(Unit of Performance Error: x 1000 USD/Day)

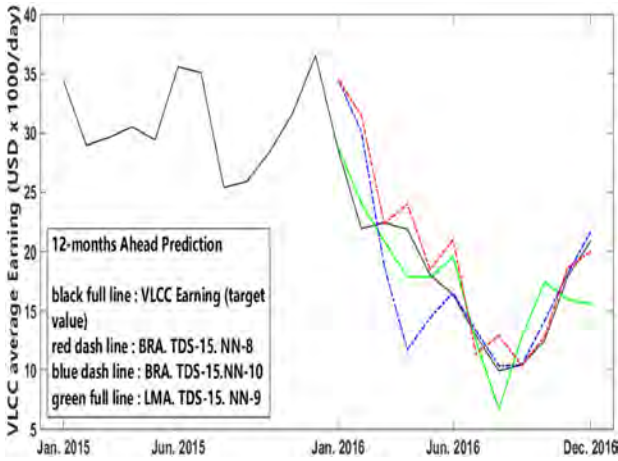


Figure 7: 12-months-ahead prediction of VLCC

Table 7: Comparison for 15-months-ahead prediction of VLCC

[ 15-months-ahead Prediction ]	ANN Architecture		
	BRA .TDS-15. NN-8	BRA .TDS-15. NN-10	LMA .TDS-15. NN-6
Epoch /computing time(sec)	229/4	341/7	15
$\mu$	-	-	0.01
Gradient	0.839	0.298	30.3
Effective number of Parameter(used/total)	116/177	149/221	-
Regression	0.97111	0.97762	0.94432
Mean Performance Error	10.5656	8.243	20.1285
Train Performance Error	1.5528	0.2142	11.0865
Validation Performance Error	-	-	42.1246
Test Performance Error	62.239	54.2748	40.9310

(Unit of Performance Error:  $\times 1000$  USD/Day)

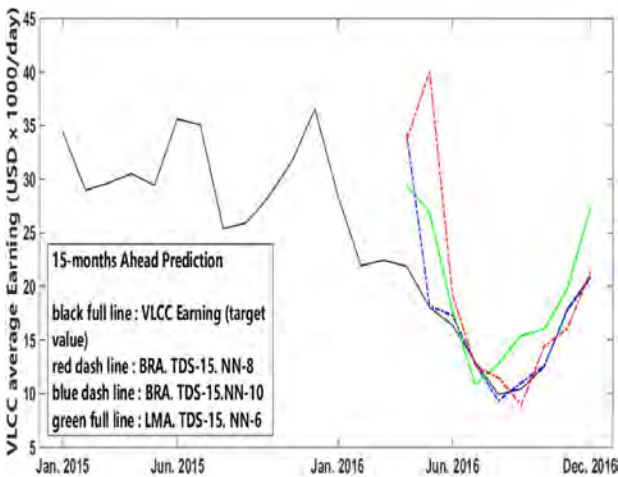


Figure 8: 15-months-ahead prediction of VLCC

where:

- LMA : Levenberg-Marquardt Algorithm
- BRA : Bayesian Regularization Algorithm
- TDS- : Test data set for full input data set (%)
- NN- : Number of neurons of hidden layer

### 5.1.7 Comparison of performance index with the training algorithm

Figure 9 shows that the Bayesian regularization algorithm produced satisfactory results for all prediction horizons as compared to the Levenberg-Marquardt algorithm. In addition, in both training algorithms, the prediction results were generally satisfactory when the number of neurons in the hidden layer was similar to the number of input variables. The exception was 12-months-ahead prediction with the Levenberg-Marquardt algorithm. Particularly in the case of the Bayesian regularization algorithm, up to 9-months-ahead prediction showed more satisfactory prediction results with smaller neurons in the hidden layer as compared to the number of input variables. However, in 12- and 15-months-ahead predictions, results that are more satisfactory were obtained with greater numbers of neurons in the hidden layer as compared to the number of input variables.

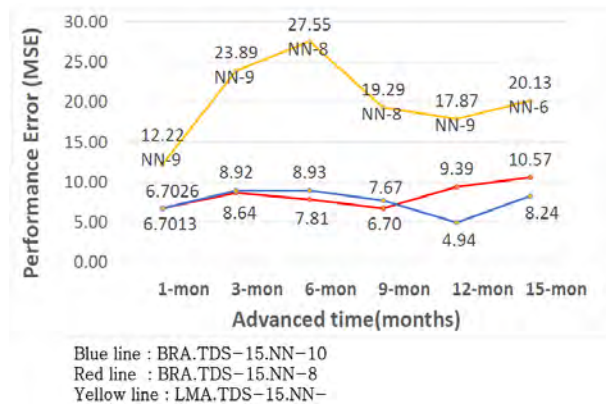


Figure 9: Comparison of performance index for training algorithm

## 6. Conclusion

In this study, we proposed alternatives to the ANN training algorithm to address the problem of multi-step forecasting. We used 204 monthly time-series data from 2000 to 2016 for dirty tankers of VLCC. Training algorithms used for the neural networks were the Levenberg-Marquardt and Bayesian regularization algorithms. A summary of this study's findings are as follows.

- The Bayesian regularization algorithm outperformed the Levenberg-Marquardt algorithm for all prediction horizons.
- The prediction performance results from both algorithms were generally satisfactory when the number of neurons in the hidden layer was similar to the number of input variables.

- In the Bayesian regularization algorithm, when the size of the hidden layer neuron increased, the number of iterations and computing time increased, and the gradient value and training performance error (MSE) converged to a smaller value.
- In the predictions within 9-months-ahead prediction, the ANN training architecture with smaller neurons in the hidden layer showed the better performance.
- In the predictions exceeding 9-months-ahead prediction, the ANN training architecture with larger neurons in the hidden layer showed the better performance.

This study showed that ANN can be used as a major tool in more accurately predicting market changes regardless of the magnitude of fluctuations. However, to improve predictive performance, designing an optimal ANN architecture for predicting targets is essential.

### Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea Government (No.2017R1A2B2010603). Further, this has been also supported by the program of the developments of convergence technology funded by TIPA/SMBA of Korea (S2356988) and by the World Class 300 R&D program (S2415805), the Special Program for Occupation (R0006323) and Business Cooperated R&D Program (R000626) of MOTIE of Korean Government.

### References

- [1] The International Energy Agency (IEA), <https://www.iea.org/statistics/kwes/>, Accessed October 20, 2017.
- [2] Clarksons Research Services, <https://sin.clarksons.net/Timeseries>, Accessed December 19, 2017.
- [3] D. Hawdon, "Tanker freight rates in the short and long run," *Applied Economics*, vol. 10, no. 3, pp. 203-217, 1978.
- [4] A. H. Alizadeh and N. K. Nomikos, "Trading strategies in the market for tankers," *Maritime Policy and Management*, vol. 33, no. 2, pp. 119-140, 2006.
- [5] K. L. Du and M. N. S. Swamy, *Neural Networks and Statistical Learning*, Springer London, 2014.
- [6] D. F. Shanno, *Recent advances in numerical techniques for large scale optimization*, Neural networks for control, Cambridge MA: MIT Press, 1990.
- [7] L. E. Scales, *Introduction to Non-Linear Optimization*, New York, Springer-Verlag, pp. 110-136, 1985.
- [8] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús, *Neural Network Design*, 2nd edition, 2014. [Online]. Available: <http://hagan.okstate.edu/NNDesign.pdf>.
- [9] A. N. Tikhonov, "On the solution of ill-posed problems and the regularization method," *Dokl. Acad. Nauk USSR*, vol. 151, no. 3, pp. 501-504, 1963.
- [10] M. Beenstock and A. R. Vergottis, "An econometric model of the world tanker market," *Journal of Transport Economics and Policy*, vol. 23, no. 2, pp. 263-280, 1989.
- [11] M. G. Kavussanos, "Time varying risks among segments of the tanker freight markets," *Maritime Economics and Logistics*, vol. 5, no. 3, pp. 227-250, 2003.
- [12] J. Li and M. G. Parsons, "Forecasting tanker freight rate using neural networks," *Maritime Policy and Management*, vol. 24, no. 1, pp. 9-30, 1997.
- [13] P. Eslami, K. H. Jung, D. W. Lee, and A. Tjolleng, "Predicting tanker freight rates using parsimonious variables and a hybrid artificial neural network with an adaptive genetic algorithm," *Maritime Economics & Logistics*, pp. 1-13, 2016.
- [14] D. V. Lyridis, P. Zacharioudakis, P. Mitrou, and A. Mylonas, "Forecasting tanker market using artificial neural network," *Maritime Economics and Logistics*, vol. 6, no. 2, pp. 93-108, 2004.
- [15] A. Santos, L. N. Junkes, and F. C. M. Pires, Jr, "Forecasting period charter rates of VLCC tankers through neural networks: A comparison of alternative approaches," *Maritime Economics & Logistics*, vol. 16, no. 1, pp. 72-91, 2013.