Optimization of Artificial Neural Network for Forecasting Earning of Dirty Tanker Markets

By

Young-Jun Jung

Supervised by

Prof. Deog-Hee Doh, Ph.D.

A dissertation submitted for the degree of

Doctor of Philosophy

Division of Mechanical Engineering

Korea Maritime & Ocean University

February 2019

Approval Sheet

This dissertation, which is an original work undertaken by Young-Jun Jung in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Mechanical Engineering, is in accordance with the regulations governing the preparation and presentation of dissertations at the Graduate School in Korea Maritime & Ocean University, Republic of Korea.

Approved by the Dissertation Committee:

Prof. Ue-Kan Kim
Chairman ____________________

Prof. Dong-Hyuk Kim
Member ____________________

Prof. Sam-Sang You
Member ____________________

Prof. Jeong-Ryul Kim
Member ____________________

Prof. Deog-Hee Doh
Member ____________________

Division of Mechanical Engineering

Graduate School of Korea Maritime & Ocean University

December 20, 2018

Contents

List of Tables

List of Figures

List of Abbreviations

ADALINE	ADAptive Linear NEuron
AFRAMAX	Average Freight Rate Assessment MAXimum
ANN	Artificial Neural Network
ARCH	AutoRegressive Conditional Heteroskedasticity
ARIMA	Auto Regressive Integrated Moving Average
BRA	Bayesian Regularization Algorithm
CST	CentiSTokes
DWT	DeadWeight Tonnage
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
GDP	Gross Domestic Product
IEA	International Energy Agency
LMA	Levenberg Marquardt Algorithm
LMS	Least Mean Squared
MATLAB	MATrix LABoratory
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NARX	Non-linear AutoRegressive model and eXogenous input
NN	Number of Neurons of hidden layer
OPEC	Organization of the Petroleum Exporting Countries
SUEZMAX	SUEZ MAXimum
TDS	Test Data Set for full input data set

UNCTAD	United Nations Conference on Trade And Development
VAR	Vector AutoRegression
VLCC	Very Large Crude oil Carrier
WS	World Scale

유조선 시장의 수익 예측을 위한 인공신경망의 최적화

정 영 준

한국해양대학교 대학원, 냉동공조공학과

요 약

해운관련 기업들은 끊임 없이 변동하는 해운시항에 대하여 그 변동의 방향과 크기를 정확하게 예측 함으로서 손실을 극소화하거나 이윤을 극대화하려 한다. 따라서 보다 정확히 해운시황을 예측하기 위한 노력으로 해운서비스의 수요와 공급, 운임 등 다양한 변수간의 경제학적 모델을 수립하여 예측을 수행하여 왔으나 해운시황 결정요인이 매우 다양할 뿐 아니라 결정 메커니즘 또한 복잡하고 급격하게 변동하는 만큼 그 변동의 방향과 크기에 대한 정확한 예측은 여전히 어려운 과제로 남아있다.

이 연구는 인공신경망 (Artificial Neural Networks, ANN)을 이용하여 VLCC, SUEZMAX, AFRAMAX 유조선 시장에 대한 경기 예측을 수행하고 유조선 시장 예측을 위한 최적의 인공신경망 모델을 제시하고자 하였다. 이러한 인공신경망 예측에 사용된 데이터는 2000년부터 2016년까지의 월간 시계열 자료 204개를 사용하였으며, 인공신경망 학습 알고리즘으로 Levenberg-Marquardt algorithm과 Bayesian regularization algorithm의 두 가지 방법을 적용하여 1개월, 3개월, 6개월, 9개월, 12개월, 15개월 앞의 시황을 예측하였다. 이를 통하여 각 알고리즘 간의 예측 정확도를 비교 하였으며, 또한 인공신경망 구조의 Hidden layer의 수와 인공신경망의 학습에 사용되는 데이터의 크기를 변화시켜 예측결과를 비교 분석하고 최적의 인공신경망 모델을 찾고자 하였다. 나아가, 인공신경망을 이용한 예측에 있어 입력 변수의 크기가 변동할 때 입력변수와 목표 변수 간의

상관 관계의 세기가 인공신경망 예측에 미치는 영향을 평가하였으며, 유조선시장 시황 변동이 서로 다른 원유 운반선의 세가지 선종, VLCC, SUEZMAX, AFRAMAX에 대하여 인공신경망을 이용한 예측을 수행하여 각 선종에 대한 인공신경망 예측의 정확도 등을 평가하였다.

연구 수행 결과로, 유조선의 세가지 선종, VLCC, SUEZMAX, AFRAMAX에 대한 예측은 3가지 선종 모두에서 Bayesian regularization algorithm에 의한 예측이 Levenberg-Marquardt algorithm에 의한 예측보다 만족한 결과를 보이고 있다. 또한, 1개월, 3개월, 6개월 및 9개월 앞선 예측에서는 인공신경망의 입력변수 보다 적은 수의 Hidden layer Neuron을 갖는 인공신경망 구조에서, 입력변수 보다 큰 수의 Neuron을 갖는 구조보다 만족한 결과를 얻을 수 있었으며, 12개월과 15개월 앞선 예측에서는 입력변수보다 많은 수의 Neuron을 갖는 인공신경망 구조에서 입력변수보다 적은 수의 Neuron을 갖는 구조보다 만족한 결과를 얻을 수 있었다. 목표변수와 입력변수 간의 상관관계에 있어, 목표변수와 입력변수 간의 상관관계 세기가 강한 입력변수의 크기가 변동할 경우에는 변동 전과 예측결과의 오차 값에 큰 변화가 없었으나, 상관관계의 세기가 약한 입력변수의 크기가 변동할 경우 예측오차가 크게 발생하였다.

유조선 시장에 대한 수익 예측은 용선가격 협상이나 용선시기의 결정 및 신조선 건조 투자결정 등 선대운영의 최적화와 자금운영의 위험을 최소화 할 수 있을 것이며, 금융전략의 수립이나 리스크 평가 등에 있어 매우 실질적이고 유효한 수단으로 활용될 수 있다. 그러나 이와 같이 인공신경망을 이용한 예측 결과가 신뢰성 있는 정보로 활용되기 위하여는 예측하고자 하는 대상에 맞는 최적의 인공신경망 모델을 선정하는 것이 무엇보다 중요하다.

핵심어 : 인공신경망, 유조선시장 예측, VLCC, SUEZMAX, AFRAMAX

Abstract

Shipping companies are looking to minimize losses or maximize profits by accurately predicting the direction and the magnitude of the fluctuations in a constantly changing maritime situation. Therefore, in order to predict the maritime market more precisely, economic model between various variables such as demand, supply and freight rate etc. of shipping service has been established and forecasted. However, the determinants of the maritime markets are very diverse and volatile, and the decision mechanism is complex. The accurate prediction of the direction and the magnitude of the variation remains as a difficult challenge.

So, the purpose of this study is to propose an optimal Artificial Neural Network (ANN) model for dirty tanker markets forecasting through VLCC, SUEZMAX and AFRAMAX tanker market prediction using ANN. The data used in this ANN forecasting are 204 monthly time series data from 2000 to 2016. The ANN training algorithm was applied in two methods, Levenberg-Marquardt algorithm and Bayesian regularization algorithm, to forecast the tanker markets with the multi-step advanced time of one month, 3 months, 6 months, 9 months, 12 months and 15 months. And the performance accuracy of each algorithm was compared with.

In addition, the hidden layer size and the test data size of the Neural Network structure were changed and the predicted results were compared and evaluated to find an optimal ANN model for tanker market prediction. Furthermore, it was investigated the effect of the correlation between input and target variables on the ANN prediction when the size of each input variables change, and ANN forecasts were performed for three types of VLCC, SUEZMAX, AFRAMAX of dirty tankers which have different market fluctuations, and

evaluated the accuracy and propriety of the results.

As a result of the study, the predictions results for VLCC, SUEZMAX and AFRAMAX tanker by Bayesian regularization algorithm are more satisfactory than those predicted by the Levenberg-Marquardt algorithm. In the one month, 3 months, 6 months and 9 months ahead predictions, the ANN structure with less number of hidden layer neurons than the number of input variables is more satisfactory than the structure with a larger number of hidden layer neurons. In the 12 months and 15 months ahead predictions, satisfactory results are obtained in the ANN structure with a larger number of hidden layers than the input variables, rather than the structure with fewer neurons than the input variables.

In the correlation between the target variable and the input variable, when the magnitude of the input variable has a strong correlation intensity with the target variable is changed, there is no significant change in the prediction performance error. However, when the size of the input variable with weak correlation strength is changed, the prediction performance error varies greatly. Predictions for the dirty tanker markets using ANN will help to minimize the risk of financial and operational problems.

Also, the forecasting information can be used as a very practical and effective means for establishing financial strategy and risk assessment. However, in order to use the prediction results as reliable information, it is most important to select the optimal artificial neural network model for the object to be predicted.

Keywords : Artificial neural networks, Tanker market prediction, VLCC, SUEZMAX, AFRAMAX

Chapter 1 Introduction

1.1 Background

Without the heat and electricity from fuel combustion, economic activity would be limited and restrained. Modern society uses more and more energy for industry, services, homes and transport. Oils are an important source of energy for almost every human product used in human life and serve as a raw material for various products, which is a driving force of global economic growth. This is particularly true for oil, which has become the largest traded commodity in world wide.

The oil supply continues to grow in absolute terms, while total oil energy supply has been decreasing from 46.2% in 1973 to around 31.7% in 2016 with the growing importance of the environmental debate **[1]**. Furthermore, it is expected by some stakeholder that oil and natural gas will likely be about 60 percent of global supplies in 2040, while nuclear energy and renewables will grow about 50 percent and a 25 percent share of the world's energy mix **[2]**. The world crude oil production in 2016 was 4448 million-ton **[1]**. The total amount of crude oil transported by sea was 1949 million-ton **[3]**, which accounted for 43.8% of the total crude oil production. The crude tanker demand for world seaborne trade was 178.5 million dwt for VLCC (200,000 dwt plus) and 56.8 million dwt for SUEZMAX (125-199,999 dwt), and AFRAMAX (85-124,999 dwt) was 55.5 million dwt **[3]**.

However, the oil tanker markets, which accounts for a large portion of the world maritime transport, is highly influenced by the interaction of supply and demand for tanker transportation services, and is highly volatile. Furthermore, there is a lot of real-time data available in the markets due to the rapid fluctuation of the markets. In addition to the traditional stakeholders, there are huge surges in many speculative powers. Increasing risk due to changes in market price fluctuations makes decisions more complicated and difficult.

Therefore, the predictions of these market changes are very important for stakeholders of the tanker market demand and supply side and other stakeholders. Also, fast and accurate market predictions that can help financial and operational decision-making for the dirty tanker markets are highly needed.

1.2 Research purposes and scope

This paper focuses on the predictions of ANN for the Earnings for VLCC, SUEZMAX, and AFRAMAX which are in responsible for a major role in the marine transportation of crude oil. The dirty tanker Earning can be derived from the time charter rates or the time charter equivalent of spot rates when the vessel is operating in the spot market. Earnings are more representative of what a tanker operating produces **[4]**.

Meanwhile, the tanker contract freights in freight markets are calculated based on an international freight index called Worldscale (WS) **[5][6]**. The Worldscale index is essentially a measure of the breakeven rate of a standard tanker on a round trip between a loading port and a discharging port as a standard tanker route specific under certain assumptions, port charges, fuel prices and other factors **[7]**. Oil tanker spot (or voyage) freight rates that have been expressed, negotiated and agreed upon are reflected in the Worldscale index **[8]**. Therefore, when choosing the ANN prediction target for the oil tanker markets, the tanker Earning (USD/day) as the forecasting target, which is directly determined according to the markets condition, was selected instead of the oil tanker international freight index.

And the effect of correlation coefficient between dirty tanker Earnings and multiple input variables of the ANN model were assessed. An optimal ANN prediction model has been developed and proposed by using the Levenberg-Marquardt algorithm and Bayesian regularization algorithm for ANN training for each ship type of VLCC, SUEZMAX and AFRAMAX. For dirty tanker markets so far, ANN predictions have been proved to be better by a number of researchers, through comparing ANN predictions with traditional statistical predictions.

As a results of this study, the optimal ANN architecture can be used for dirty tanker markets forecasting with alternative training algorithms. Furthermore the results of this study can help to improve the prediction accuracy of the market changes, so that it can provide information for more accurate judgment to many stakeholders. Also it can be used as an information for establishing financial and operational strategies by predicting income forecasts and risks as well as determining the optimal timing for ship charter in / out, determination of fleet size, charter period and charter rate.

This study also makes sense for the first on the ANN forecasting for the SUEZMAX and AFRAMAX tanker market as well as VLCC market. And it can be deemed as a considerable contribution that this study prove that ANN forecast can be widely applied as a contributory information regardless of the ship type of maritime market.

1.3 Predictions on shipping markets

Demand and supply of international shipping services are influenced not only by various external factors such as economic growth, trade policies, changes in political conditions such as diplomatic relations, and changes in the natural environment, but also by factors such as production price factors and technology development, shipbuilding, steel, petroleum industry and port condition. The demand and supply change for the shipping service will be balanced through the freight rates determined in the shipping market. The freight rates in the markets will affect the newbuilding market, the charter market and the second hand ship market. Thereby forming a cyclic loop that affects an influence to the supply and demand of the global shipping market.

Therefore, in order to predict maritime market, many diverse economists have studied how to interact and construct linkages in the maritime market under the interaction of supply and demand variables. The factors influencing the demand for maritime transport are global economy, international maritime transport volume, profit margins, political events and transport costs.

On the supply side is world fleet and its productivity, shipbuilding, shipbreaking and freights **[9]**. As a result of economic studies on the tanker market, Zannetos **[10]** and D. Gren et al **[11]** presented a framework for understanding the relationship between spot rates and the long-term charter rates in the oil tanker market. Hawdon **[12]** derived an equations for the tanker freight rates under the hypothesis that the demand for oil freight services is a simple function of total world trade in oil, and Beenstock and Vergottis **[13]** established a theoretical model for the correlation between the freight markets and the ship market and applied it to the world tanker market.

For the prediction and analysis of various factors that constitute the shipping market, one of the typical methods used when forecasting maritime market conditions is a time series analysis that predicts future changes by identifying empirical laws required for forecasting using only the information contained in the observed historical data.

The time series analysis technique began to be widely used in the mid-1970s by G.E.P. Box and G.M. Since Jenkins proposed the Box-Jenkins model **[14]** which incorporates existing time series prediction theory. In addition, a variety of forecasting and analysis techniques have been developed and applied, such as ARIMA models, ARCH, GARCH, and VAR models for forecasting the shipping market.

In this study, Kavussanos **[15]** applied the GARCH model to investigate the volatility of freight rates in the spot and time charter markets of dry-bulk vessels. Kavussanos **[16]** has evaluated the relative risks involved in operating tanker vessels in world spot and time charter markets through the use of Co-integration Error Correlation ARCH models. Kagkarakis et al **[17]** used the VAR model to estimate the price in the ship-demolition markets.

The tanker freight market is characterized by the interaction between many determinants of supply and demand for tanker transportation services **[9][10][18]**. To forecast the dynamics and fluctuations of the freight rates in the tanker freight markets, many researches have been developed using univariate or multivariate time series analysis techniques **[16][19]-[21]**, and ANN models **[22]-[24]**.

Artificial Neural Networks (ANN) have powerful pattern classification and pattern recognition capabilities, and are being used for a wide variety of tasks in many different fields of business, industry and science. One major application area of ANN is forecasting **[25]**. ANN are inspired by biological systems, particularly by research into the human brain, and be able to learn from and generalize from experience. ANN is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. ANN are also parameterized computational nonlinear algorithms for numerical data, signal and image processing.

These algorithms are either implemented on a general-purpose computer or are built into a dedicated hardware **[26]**.

In order to verify the accuracy of this ANN, the researchers conducted a study comparing the prediction results by ANN with the prediction results by traditional time series techniques **[24][27]**. As a result of these studies, ANN can be more effective in forecasting by time series method in monthly and quarterly forecasting than annual time series forecasting, and ANN is more effective than statistical method in predicting three or more period horizon on the total forecasting horizon **[28]**.

The first ANN predictions for tanker freight rates were made by Li & Parsons **[22]**. In their study, three variables were used in predicting the dirty tanker spot freight rates (WS) using the tanker demand data and tanker supply data for 190 monthly time series data points from Jan. 1980 to Oct. 1995. And the ANN structure was consisted of one or three variables as inputs, one output and one hidden layer. The best number of neurons in the hidden layer was varied from case to case to obtain the best performance error tolerances for all the cases.

In subsequent studies, in order to more accurately predict freight rate fluctuations in tanker markets, forecasts using ANN were made by applying more large number of independent variables affecting the freight markets.

Lyridis et al **[23]** investigated the VLCC spot freight rates (WS) by using ANN with monthly time series data from Oct. 1979 to Dec. 2002 as independent variables; demand for oil transportation, active fleet, crude oil production, crude oil price, surplus as a percentage of active fleet. Their study attempted to uncover the benefits of using ANN in forecasting VLCC spot freight rates.

In order to obtain the best prediction result in ANN prediction, ANN forecasting was performed by changing the number of input variables for each prediction interval, and the number of optimal input parameters was observed by comparing the prediction results.

Santos et al **[24]** performed ANN forecasts for VLCC period charter rates (USD/Day) instead of spot freight rates (WS), which were previously predicted by researchers. And in applying ANN forecasting model, two different ANN model of multi-layer perceptron and radial basis functions were applied and the results were compared.

Eslami et al **[29]** developed a hybrid tanker spot freight rates (WS) prediction model based on an artificial neural networks and an adaptive genetic algorithm which searches a near-optimal combination of network parameters to improve the accuracy of ANN. And the prediction results are compared with those of previous researchers.

1.4 Structure of the paper

The paper is organized as follows.

In **Chapter 2**, it encompasses an essential and theoretical knowledge of ANN structure to design its architecture. The backpropagation algorithm introduced in this chapter is a major breakthrough in neural network research. And The Levenberg-Marquardt algorithm and Bayesian regularization algorithm that provide significant speedup and make the algorithm more practical are introduced to obtain reasonable results before ANN prediction execution.

Chapter 3 concentrate on practical aspects of the methodology used in this study including data collection, data normalization, and ANN architecture etc. for ANN prediction. In collecting data, the ship type is distinguished to VLCC, SUEZMAX, and AFRAMAX in accordance with the ship size, and the monthly data from January 2000 to December 2016 for each vessel type are obtained from resources of the reliable institutes. The Non-linear AutoRegressive model with eXogenous inputs (NARX network) is introduced as an ANN model which is applied in this study.

In **Chapter 4**, it is described the actual application process using the methodology mentioned in Chapter 4. And the schematic of ANN network for the tanker market prediction for applying Levenberg-Marquardt algorithm and Bayesian regularization algorithm is presented, and the details of the prediction calculation method and the verification method of the results to prove are presented. Also the predictive results are evaluated according to the training algorithms of the ANN architectures.

Finally, in **Chapter 5**, the thesis is wrapped up with conclusions including suggestions for tanker markets prediction by ANN.

Chapter 2 Artificial Neural Networks

2.1 An overview of Artificial Neural Networks (ANN)

The discipline of neural networks models human brains. The average human brain consists of nearly 10^{11} neurons of various types, with each neuron connecting to up to tens of thousands of synapses. As such, neural network models are also called connectionist models. Information processing is mainly in the cerebral cortex, the out layer of the brain. Cognitive functions, including language, abstract reasoning, and learning and memory, represent the most complex brain operation to define in the terms of neural mechanisms. **[30]**

Artificial Neural Networks (ANN) are composed of a number of highly interconnected simple processing elements called neurons or nodes. Each node receives an input signal which is the total "information" from other nodes or external stimuli, process it locally through an activation or transfer function and produces a transformed output signal to other nodes or external outputs. Although each individual neurons implements its function rather slowly and imperfectly, collectively a network can perform a surprising number of tasks quite efficiently. This information processing characteristics make ANN a powerful computational device and are able to learn from examples and then to generalize to examples never before seen **[25]**. A basic elements of an artificial neural network is depicted in **Figure 2.1 [31]**.

ANN can be treated as a general statistical tool for almost all disciplines of science and engineering. The applications can be in function approximation, classification, clustering and vector quantization, associative memory, optimization, feature extraction and information compression **[30]**.

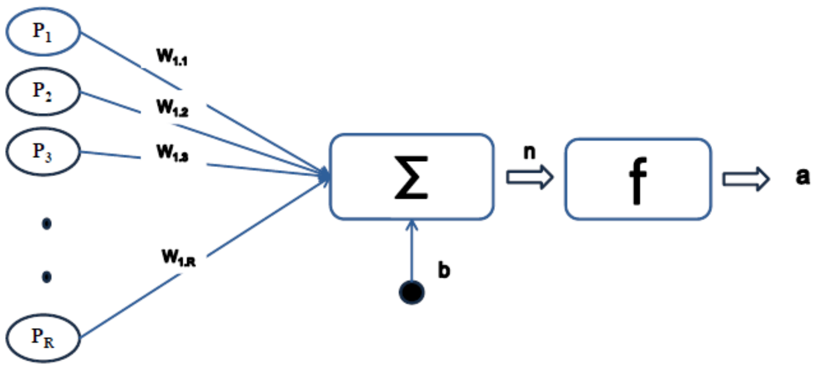

Figure 2.1 Multiple-input neuron

The individual inputs P_1, P_2, ⋯, P_R are each weighted by corresponding elements W_{11}, W_{12}, ⋯, W_{1R} of the weight matrix W. The neuron has a bias $\boldsymbol{b}$, which is summed with the weighted inputs to form the net input n:

$$\boldsymbol{n} = W_{11} P_1 + W_{12} P_2 + \cdots + W_{1R} P_R + \boldsymbol{b} \qquad (1)$$

The neuron output a of a transfer function can be written as

$$\boldsymbol{a} = f(WP + \boldsymbol{b}) \qquad (2)$$

Many different ANN models have been proposed since 1980s. One of the most influential models among them is the Multi-Layer Perceptron (MLP). The MLP networks are used in variety of problems especially in forecasting because of their inherent capability of arbitrary input-output mapping.

In feedforward MLP networks, the neurons are organized in the form of layers. The neurons in a layer get input from the previous layer or first layer, and feed their output to the next layer. The last layer or the highest layer of neurons is called the output layer and the one or more intermediate layers between the inputs and output layers are called the hidden layers. Artificial neuron mainly consists of weights, bias and activation function.

In **Figure 2.2 [31]**, third layer is called output layer and the other layers of the first and second layer are called hidden layers. There are R inputs, S^1, S^2 and S^3 neuron in the each layers. And different layers can have different numbers of neurons. Each layer has its own weight matrix W(for the first layer is written as W^1), its own bias vector $\boldsymbol{b}$, a net input vector $\boldsymbol{n}$ and an output vector $\boldsymbol{a}$.

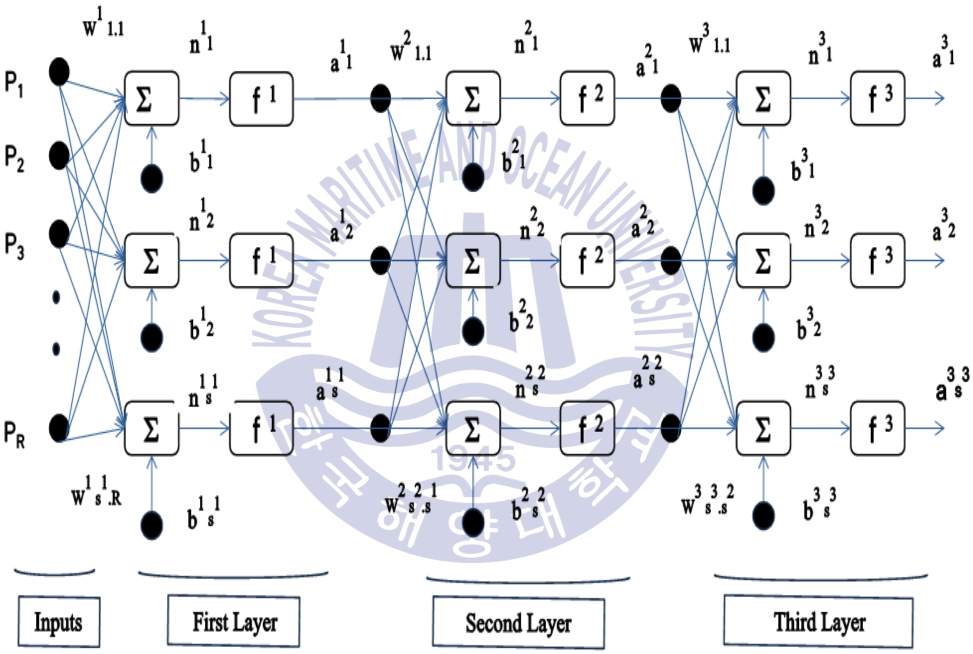

Figure 2.2 Multiple layers of neurons

2.2 Design of ANN model

2.2.1 Supervised learning

Learning (training) is a fundamental capability of neural networks. Learning rules are algorithm for finding suitable weights *W* and/or other network parameters. Learning of a neural network can be viewed as a nonlinear optimization problem for finding a set of network parameters. Learning methods are conventionally divided into supervised, unsupervised, and reinforcement learning **[30]**. Supervised learning adjusts network parameters by a direct comparison between the actual network output and the desired output. Supervised learning is a closed-loop feedback system, where the error is the feedback signal.

The error measure, which shows the difference between the network output and the output from the training samples, is used to guide the learning process. The error measure is usually defined by the mean squared error (MSE). To decrease error toward zero, a gradient descent procedure is usually applied. The gradient decent method always converges to a local minimum in a neighborhood of the initial solution of network parameters. The least mean squared algorithm (LMS) and backpropagation algorithms are two most popular gradient descent based algorithms **[30]**.

The most commonly used learning algorithm for a supervised neural network is a backpropagation algorithm which is proposed for the MLP model in 1986 by Rumelhart et al **[32]**. The goal of the backpropagation, as with most training algorithms, is to iteratively adjust the weights in the network to produce the desired output by minimizing output error.

The backpropagation is a gradient descent approach in that it uses the minimization of first order derivatives to find an optimal solution. It works with a training set of input vectors and target output vectors.

The training algorithm iteratively tries to force the generated output vectors to the desired output vector by adjusting the weights in the network through the use of a generalized delta rule **[33]**.

2.2.2 Mean squared error (MSE)

A quantitative measure of neural network performance is called the performance index, which is small when the network perform well and large when the network performs poorly. Many ANN researchers have investigated to develop algorithm to optimize a performance index. In other word, “optimize” means to find the value of the minimized performance index.

When the standard performance index ($F(\boldsymbol{x})$) is represented by Taylor series expansion and $\boldsymbol{x}$ is the scalar parameter at iteration κ, the general minimization algorithm is:

$$\Delta \boldsymbol{x}_k = (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) = \alpha_k \boldsymbol{P}_k \tag{3}$$

Where the vector $\boldsymbol{P}_k$ represents a search direction at iteration κ, and α_k is the learning rate. This equation is written in matrix form.

Three different categories of optimization algorithm to minimize the performance index through training of neural networks are the steepest descent, Newton’s method and conjugate gradient.

In the training algorithms of ANN, the Least Mean Squared (LMS) algorithm is an example of supervised training, in which the learning rule is provided with a set of examples of proper network behavior.

$$\{P_1, t_1\},\ \{P_2, t_2\},\ \{P_3, t_3\},\ \ldots\ldots,\ \{P_q, t_q\} \tag{4}$$

Where P_q is an input to the network, and t_q is the corresponding target output.

The LMS algorithm and the backpropagation algorithm for multilayer networks adjusts the weights and biases of the network in order to minimize the mean square error, where the error is the difference between the target output and the network output.

$$F(x) = \mathrm{E}[e^2] = \mathrm{E}[(t-a)^2)] \tag{5}$$

Where *F(x)* is the performance index of the neural network, E[ ] is denoted as an expected value and the expectation is taken over all sets of input/target pairs **[36]**. And the mean square error is expressed:

$$\mathrm{MSE} = \frac{1}{Q}\sum_{q=1}^{Q}(t_q - a_q)^2 \tag{6}$$

In the multilayer networks having multiple outputs, the performance index *F(**x**)* is expressed:

$$F(\boldsymbol{x}) = \sum_{q=1}^{Q}\left(t_q - a_q\right)^T (t_q - a^q) \tag{7}$$

Where a_q is the network output for input p_q. and t_q is the target output corresponding to the input p_q.

2.2.3 Least-mean squared (LMS) algorithm

The LMS algorithm achieves a robust separation between the patterns of different classes by minimizing the MSE rather than the number of misclassified patterns through the gradient-descent method **[30]**. The following two equations make up the LMS algorithm to minimize the MSE through adjusting the weights and biases of the ADALINE, and these equations which are referred to as the delta rule or the Widrow-Hoff learning algorithm can be written in matrix notation **[31]**.

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + 2\alpha\boldsymbol{e}(k)\boldsymbol{P}^T(k) \tag{8}$$

$$\boldsymbol{b}(k+1) = \boldsymbol{b}(k) + 2\alpha\boldsymbol{e}(k) \tag{9}$$

Where $\boldsymbol{W}(\kappa)$ is the weight at iteration κ, $\boldsymbol{e}(\kappa)$ and $\boldsymbol{b}(\kappa+1)$ are the error at iteration κ and the biases at iteration $\kappa+1$, α is the learning rate, and $\boldsymbol{P}^T(\kappa)$ is the input at iteration κ.

2.2.4 Backpropagation learning algorithm

Backpropagation learning is the most popular learning for performing supervised learning tasks **[32]**. The backpropagation algorithm is a generalization of the delta rule called the LMS algorithm. Thus it is also called the generalized delta rule. The backpropagation algorithm propagates to backward the error between the desired target and the network output through the network. After providing an input, the output of the network is then compared with a given target and the error of each output unit calculated.

This error is propagated backward, and a closed loop control system is established. The weights can be adjusted by a gradient-descent based algorithm **[30]**. The backpropagation can be used to train multilayer networks. As with the LMS learning law, the performance index of the backpropagation is mean squared error.

The difference between the LMS algorithm and backpropagation is only in the way in which the derivatives are calculated. In order to calculate the derivatives, it is needed to use the chain rule of calculus **[31]**.

As shown in **Figure 2.2**, for multilayer networks the output of one layer becomes the input to the following layer.

This operation are described as follows :

$$\boldsymbol{a}^{m+1} = f^{m+1}(\boldsymbol{W}^{m+1}\boldsymbol{a}^{m} + \boldsymbol{b}^{m+1}) \quad \text{for } m = 0, 1, 2, \ldots, M-1 \tag{10}$$

Where M is the number of layers in the network.

For the multilayer network the error is an indirect function of the weights in the hidden layer, therefore the chain rule of calculus to calculate the derivatives are applied to. The approximate steepest descent algorithm using the chain rule of calculus can be written in matrix notation.

$$\boldsymbol{W}^{m}(k+1) = \boldsymbol{W}^{m}(k) - \alpha\boldsymbol{S}^{m}(\boldsymbol{a}^{m-1})^{T} \tag{11}$$

$$\boldsymbol{b}^{m}(k+1) = \boldsymbol{b}^{m}(k) - \alpha\boldsymbol{S}^{m} \tag{12}$$

Where $\boldsymbol{S}^{m}$ is the sensitivity at layer m.

2.2.5 Levenberg-Marquardt algorithm

The basic backpropagation algorithm is too slow for most practical application. This has encouraged considerable research on methods to accelerate the convergence of the algorithm. As a consequence, several variations of backpropagation to provide significant speedup and make the algorithm more practical have been developed using heuristic techniques **[37]** and numerical optimization techniques **[38]**.

The Levenberg-Marquardt algorithm, which is one of numerical optimization techniques, was designed for minimizing functions that are sums of squares of other nonlinear functions in neural network training where the performance index is the mean squared error.

When the performance index is $F(\boldsymbol{x})$, the Levenberg-Marquardt algorithm for optimizing the performance index $F(\boldsymbol{x})$ is represented as **[39]**:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\boldsymbol{J}^T(\boldsymbol{x}_k)\boldsymbol{J}(\boldsymbol{x}_k) + \mu_k]^{-1}\boldsymbol{J}^T(\boldsymbol{x}_k)\boldsymbol{V}(\boldsymbol{x}_k) \tag{13}$$

Here, as the changing value of μ_K, the performance index $F(\boldsymbol{x})$ of the network can be adjusted with the optimization algorithms in small learning rate. Where $\boldsymbol{J}(\boldsymbol{x}_k)$ and $\boldsymbol{V}(\boldsymbol{x}_k)$ are the matrix elements to compute the gradient.

2.2.6 Generalization and Bayesian regularization algorithm

In operation a multilayer network, if the number of neuron is too large, the network will over-fit the training data. This means that the error on the training data will be very small, but the network will fail to perform as well when presented with new data. A network that generalizes well will perform as well on new data as it does on the training data. The complexity of neural network is determined by the number of free parameters that weights and biases, which is determined by the number of neurons. If network is too complex for a given data set, then it is likely to over-fit and to have a poor generalization **[31]**.

There are two approaches to improve the generalization capability of neural network: restricting the number of weights or restricting the magnitude of the weights called the regularization. The simplest method for improving generalization is early stopping **[40]**.

In a multilayer network, after removing the test data set from the input data set, the available data is divided into two parts: a training data set and validation data set. The training data set is used to determine the weight update at each iteration. The validation data set is an indicator of what is happening to the network function in between the training points, and the error of the validation data set is monitored during the training process. When the error on the validation data set goes up for several iteration, the training stopped, and the weights that produced the minimum error on the validation data set are used as the final trained network weights. The test data set is used to calculate its error. The error of the test data set, which is a measure of the generalization capability of the network, will give an indication of how the network will perform in the future.

This method to stop the training is called cross-validation **[41].** And another method for generalization is called regularization **[42]**. This regularization can be written as the sum of squares of the network weights, as follows:

$$F(\boldsymbol{x}) = \beta E_D + \alpha E_W = \beta \sum_{q=1}^{Q} (\boldsymbol{t}_q - \boldsymbol{a}_q)^T (\boldsymbol{t}_q - \boldsymbol{a}_q) + \alpha \sum_{i=1}^{n} \boldsymbol{X}_i^2 \qquad (14)$$

Where $F(\boldsymbol{x})$ is called the regularized performance index, and the ratio α/β control the effective complexity of the network solution. There are several technics for setting the regularization parameter. Bayesian Regularization among these methods is an automatic selection of the regularization parameter **[43]**.

$$\alpha^{MP} = \frac{\gamma}{2Ew(x^{MP})} \quad \text{and} \quad \beta^{MP} = \frac{N-\gamma}{2E_D(x^{MP})} \qquad (15)$$

Where $\gamma = \mathrm{n} - 2\alpha^{MP}\, tr(H^{MP})^{-1}$ is called the effective number of parameters, and n is the total number of parameters in the network. The γ is a measure of how many parameters (weights and biases) in the neural network are effectively used in reducing the error function.

Chapter 3　Methodology

3.1　Data and pre-processing

3.1.1　Data collection

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.

To collect the data sets for tanker markets prediction, firstly the ship type is distinguished to VLCC, SUEZMAX, and AFRAMAX tanker.　The monthly data sets from January 2000 to December 2016 for each tankers were obtained from Clarkson research services, the International Energy Agency (IEA), the Organization of the Petroleum Countries (OPEC) **[44]** and related organizations.

The two larger size tankers of VLCC and SUEZMAX are exclusively involved in crude oil transportation.　AFRAMAX vessels are also involved in transportation of crude oil, however, they contribute to product transportation time to time.　In this paper, the data set related to AFRAMAX is also restricted to the vessel transporting crude oil exclusively.

The purpose of this study is to provide an optimal ANN training architecture.　To do this, it was tried to derive the optimal ANN training architecture by comparing the results obtained from the Levenberg-Marquardt algorithm and Bayesian regularization algorithm by each ship type, and by changing the computing parameters of the ANN training algorithm for the actual prediction application and evaluating the results.

Therefore, when collecting data sets for each ship type, data samples such as unit and size were applied with the identical conditions **[45]**.

European brent spot oil prices is applied for crude oil prices, and the average prices for the new building prices, second hand ship prices and demolition prices are applied to. And the time charter rates and the targeting earnings for each ship type adapt its average values. Also, bunker prices are 180 CST rotterdam prices. World GDP time series data from UNCTAD **[46]** were converted from quarterly values to monthly using interpolation method. The aggregated data is made up of nine independent variables and one dependent variable, and each variable has 204 monthly observations from January 2000 to December 2016.

3.1.2 Data normalization

One of the most common tool to obtain better results of neural network is to utilize data normalization. Data normalization can also speed up training time by starting the training process for feature within the same scale.

Data normalization is especially useful for modeling applications where the inputs are generally on widely different scale. Data normalization is performed before the training process begins. Nonlinear activation functions are normalized to a value (0, 1) for logistic function or (-1, 1) for hyperbolic tangent function. When nonlinear transfer functions are used at the output nodes, the desired output values must be transformed to the range of the actual outputs of the network **[25]**. In this study, prediction is performed with a multilayer networks using sigmoid transfer functions in the first layer of hidden layer.

This sigmoid transfer functions are open used in the hidden layer. In the first layer, the net input is a products of the input times the weight plus the bias. If the input is very large, the weight must be small in order to prevent the transfer function from becoming saturated. In contrary, if the input values are very small, large weights are needed to produce a large net input. Thus, it is standard practice to normalize the inputs before applying them to the network. When the input values are normalized, the magnitudes of the weights have a consistent meaning in using regularization. The normalization step is applied to the input values and target values in the data sets **[31][33]**. The normalization can be done by the following equation:

$$\boldsymbol{p}_n = \frac{2\,(\boldsymbol{p} - \boldsymbol{p}_{min})}{\boldsymbol{p}_{max} - \boldsymbol{p}_{min}} - 1 \tag{16}$$

Where p_{min} is the vector containing the minimum values of each element of the input vectors in the data set, $\boldsymbol{p}_{max}$ contains the maximum values, and $\boldsymbol{p}_n$ (from -1 to 1) is the resulting normalized input vector.

3.2 Identification of ANN architecture

After collecting the data for forecasting of tanker markets, the type of ANN architecture is to be determined to solve the problem of tanker market prediction, and the specific details of how many neurons and layers it will be used in the network are to be decided. In ANN dynamic networks, the output depends not only on the current input to the network, but also current or previous inputs, outputs or states of the network. Tanker markets prediction is part of a time series analysis that predicts the future value of a time series. Therefore in this paper, the dynamic networks had been selected as an appropriate ANN model to forecast dirty tanker markets. The Non-linear Auto Regressive model with eXogenous inputs (NARX networks) **[30][31]**, a widely used network for applying predictions, is a recurrent dynamic network with feedback connections that encompass multiple layers of the network, as shown in **Figure 3.1 [31]**.

Until now, prediction using ANN for the VLCC tanker market has been done by the Levenberg-Marquardt training algorithm **[22]-[24][29]**, but no Bayesian regularization algorithm was used to predict the VLCC tanker market. Therefore this paper focuses on the prediction of ANN with the Levenberg-Marquardt algorithm and Bayesian regularization algorithm to evaluate the prediction accuracy of these two training algorithms.

After determining the network structure, the number of hidden layers in these two learning algorithms is decided to one to allow easy comparison of performance results and functions, and the ANN model implementing the backpropagation algorithm do not have too many layers, since the time for training of the network grows exponentially.

The number of neurons in the hidden layer are determined by the complexities of the function that is being approximated or the decision boundaries that are being implemented. Therefore, to determine the number of neurons in the hidden layer to find the best prediction performance for the VLCC tanker market, it is adjusted the number of neurons in the hidden layer of the ANN structure using the Levenberg-Marquardt algorithm to find the best performance without any overfitting. Also, in the prediction using Bayesian regularization algorithm, the ANN performs prediction using 8 and 10 neurons of the hidden layer based and evaluates the performance results of these two cases.

The inputs of the ANN model have 9 nodes for input signals. The hidden layer is made up the neurons with the tan-sigmoid transfer function selected as their activation function, and the output layer has the linear transfer function. The number of neurons in the output layer is the same size with the target.

For multi-step ahead prediction, the output is fed back to the input of the feedforward neural network as part of the standard network. The predicted value is fed back as input to network for next prediction and all other inputs are shifted to back ward one unit of time. It is called the closed loop which is useful for multi-step prediction.

For one step ahead predictions, the true output, which is available during the training of the network, is used instead of feeding back the estimated output. This means that the network is trained by ℓ-step ($\ell > 1$) apart differenced data as input for network. It is called the open loop and is useful for training. The typical workflow is to fully create the network in open loop, and only when it has been trained (which includes validation and testing steps), it is transformed to closed loop for multi-step ahead prediction **[22]**.

The prediction for tanker markets earning with the advanced time of one-step (month) ahead, 3-step, 6-step, 9-step, 12-step and 15-step are performed by MATLAB with the neural network toolbox.

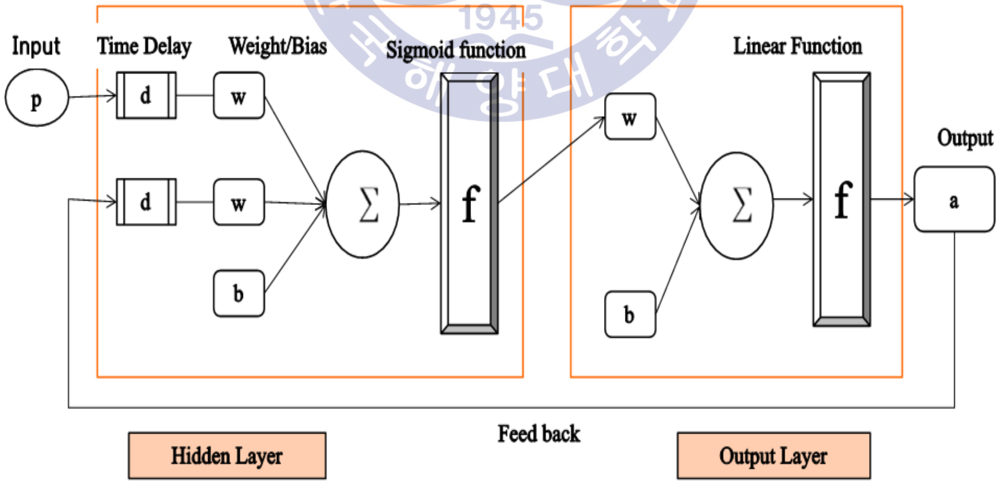

Figure 3.1 NARX network (Closed loop) for the tanker market prediction

3.3 Training and post training validation

For training the network, the Levenberg-Marquardt algorithm and the Bayesian regularization training algorithm are applied for VLCC tanker, and the Bayesian regularization training algorithm are only applied for SUEZMAX and AFRAMAX tanker. These algorithms are implemented batch learning scheme for weight updating. In batch learning scheme, the training samples are fed into the network and the change in all weights is computed from each sample. Then at the end it is updated the weights according to the sum of all updates.

For multilayer network, the weights and biases are generally set to small random values. In the case where the input are normalized to fall between -1 and 1, it is uniformly distributed between -0.5 and 0.5.

As an important tool for neural network validation, the regression coefficient between the network output and the target, known as the R value, should be close to 1 to ensure reliable ANN performance results.

And where applying the dynamic networks for prediction, such as the focused time-delay neural network, there are two important concepts when analyzing the trained prediction network. One is that the prediction errors should not be correlated in time. And another one is that the prediction errors should not be correlated with the input sequence.

In order to test the correlation of the prediction in time, the autocorrelation function is used:

$$R_e(\tau) = \frac{1}{Q-\tau} \sum_{t=1}^{Q-\tau} e(t)e(t+\tau) \qquad (17)$$

If the prediction errors are uncorrelated (white noise), it can be expected that $R_e(\tau)$ is close to zero, except when $\tau = 0$. To determine if $R_e(\tau)$ is close to zero, it can be defined by approximate 95% confidence interval **[14]** using the range:

$$-\frac{2R_e(0)}{\sqrt{Q}} < R_e(\tau) < \frac{2R_e(0)}{\sqrt{Q}} \tag{18}$$

To test the correlation between the prediction errors and the input sequence, the cross-correlation function is used:

$$R_{pe}(\tau) = \frac{1}{Q-\tau}\sum_{t=1}^{Q-\tau} p(t)e(t+\tau) \tag{19}$$

If there is no correlation between the prediction errors and the input sequence, it can be expected that $R_{pe}(\tau)$ is close to zero for all τ. To determine if $R_{pe}(\tau)$ is close to zero, it can be defined by approximate 95% confidence interval **[14]** using the range:

$$-\frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}} < R_{pe}(\tau) < \frac{2\sqrt{R_e(0)}\sqrt{R_p(0)}}{\sqrt{Q}} \tag{20}$$

Chapter 4 Implementation of Methodology

4.1 Implementation

4.1.1 Data processing

In order to predict the dirty tanker Earning as Target variable for VLCC, SUEZMAX and AFRAMAX, ANN prediction was performed using 9 Input variables- 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 as in the section 3.1.1 above.

Of the 204 monthly data from January 2000 to December 2016, the data used for ANN training were 180 monthly data from January 2000 to December 2014. The rest period from January 2015 to December 2016 was used as a multistep ahead forecasting, and the actual average Earing data of this period was used as target data for ANN supervised learning.

Total 180 data points of each variables ranged from January 2000 to December 2014 was randomly sampled and divided into three data sets during computation: training, validation and test data set. It is important that each of these data be representative of the full data set. In general, the validation and test data set cover the same region of the input space as the training data set. In the study by Jun li et al **[22]** and Lyridis et al **[23]** applying the Levenberg-Marquardt algorithm to VLCC, the test sequence was constructed as a continuous segment of the original data set with specifying the test data sampling range of the last 2 and 3 years monthly data of the full data set which applied.

Also, when applying the Levenberg-Marquardt algorithm to VLCC in this study, the training set made up about 70% of the full data set with about 15% for each validation and test set, during its iterative computing. And when applying the Bayesian regularization training technique to VLCC, SUEZMAX and AFRAMAX, the ANN prediction was performed in two cases: the data set for testing is 15% and the case is 20%, and the results of the two cases were compared. The reason for dividing the test data set into two types, 15% and 20%, is to broaden the test data and to compare the results by varying the width of the training data. The validation set is not necessary to be assigned to the Bayesian regularization algorithm.

4.1.2 ANN model for tanker markets prediction

The schematic diagram of ANN network for the tanker market prediction is shown below **Figure 4.1**.

When the Levenberg–Marquardt algorithm was applied to VLCC, 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.

And the number of neurons in the output layer with the linear function as its activation function is the same as the size of the target. The schematic diagram of ANN network for the tanker market prediction is presented in the bellow **Figure 4.1**.

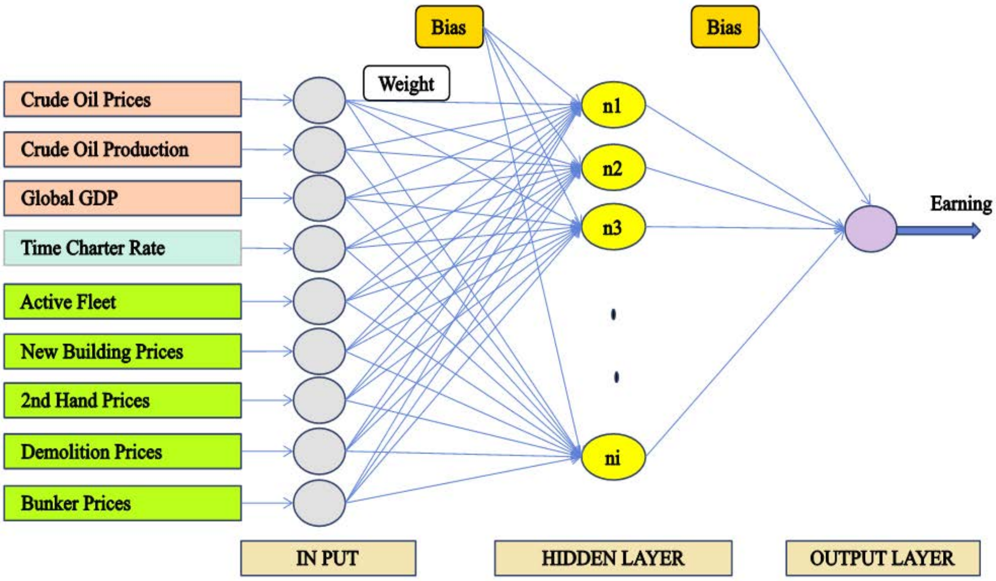

Figure 4.1 Schematic diagram of ANN network for the tanker market prediction

4.1.3 Computation

Experiments have been carried out to identify the optimal ANN architecture for the forecasting of the Earning of tanker markets with an advanced time of one-step (i.e. one month) ahead, 3-step ahead, 6-step ahead, 9-step ahead, 12-step ahead and 15-step ahead. The value of the tagged delay line as a time delay was 2 months without any changing during implementation.

Each implementation for the prediction was repeated several times to identify the optimal parameters and conditions of the network. When the results were dissatisfied with the network's performance on the target data, it was trained again or retrained after reinitializing the weights and bias. Each time a neural network is trained, can result in a different solution due to different initial weight and bias values and different data points into training, validation, and test sets. As a result, different neural networks trained on the same problem can give a different output for the same input.

When training with the Levenberg-Marquardt algorithm, the number of neurons can be adjusted to prevent overfitting or extrapolation. If the performance on the training set is good, but the validation and test performance is significantly worse, then can be reduced the number of neurons to improve the results. If training performance is poor, then it can be increased the number of neurons.

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 a similar characteristics
- Expert judgement considering various parameters and performance indices

When training with the Bayesian Regularization training technique, there are a total 221 parameters in the 9(input)-10(number of neurons of the hidden layer)-1(output) network, and a total 177 parameters in the 9-8-1 network. The effective number of parameters was about min. 56 and max. 171 during training the 9-10-1 network for VLCC. The training of the 9-10-1 network effectively used it less than 77.3% of the total number of the weights and biases. In 9-8-1 network, the effective number of parameters was about min. 51.9 and max. 133 and used it less than 75.1%.

The computational results of the Bayesian Regularization training algorithm were considered as reasonable, when the algorithm was fitted with the following considerations:

- The final mean performance index (MSE) was small

- The training error (training performance index (MSE)) was small
- Expert judgement considering various parameters and performance indices

The computer specification used for the calculation was Intel® Core ™ i5-5200U CPU @ 2.20GHz

4.1.4 Validation

The regression plots display the network outputs with respect to targets for training, validation and test sets. For a perfect fit, the data will fall along a 45 degree line, where the network outputs are equal to the targets. Regression is considered appropriate when the R value is at least 0.93. The R value is an indication of the relationship between the outputs and targets. If R=1, this indicates that there is a linear relationship between the outputs and targets. If R is close to 0, then there no linear relationship between the outputs and targets.

The autocorrelation function of the prediction error and the cross-correlation function to measure the correlation between the input and the prediction error were used for the ANN prediction model validation with the help of the graphics.

4.2 Prediction performance results

In this study, the multi-step ahead predictions were performed for VLCC, SUEZMAX and AFRAMAX tanker market from January 2015 to December 2016. Of the 204 monthly data from January 2000 to December 2016, the data used for ANN training were 180 monthly data from January 2000 to December 2014. The rest period from January 2015 to December 2016 was used as a multistep ahead predictions.

This actual average Earing data over the two years was used as supervised learning target data to compute the ANN predictive performance.

Figure 4.2 shows the average Earning trend for VLCC, SUEZMAX and AFRAMAX tanker markets by time series data from January 2000 to December 2016 **[3]**.

The Earning value of SUEZMAX is the largest among the three ship type, and the width of the change is also large. Also, the variation of this Earning is the largest SUEZMAX. For VLCC, the size of the change is relatively small compared to other tanker types, and for AFRAMAX, the middle fluctuation value between SUEZMAX and VLCC is displayed. However, it can be seen that the average earning of each type of tanker changes to bigger or smaller depending on the market condition of the tanker types.

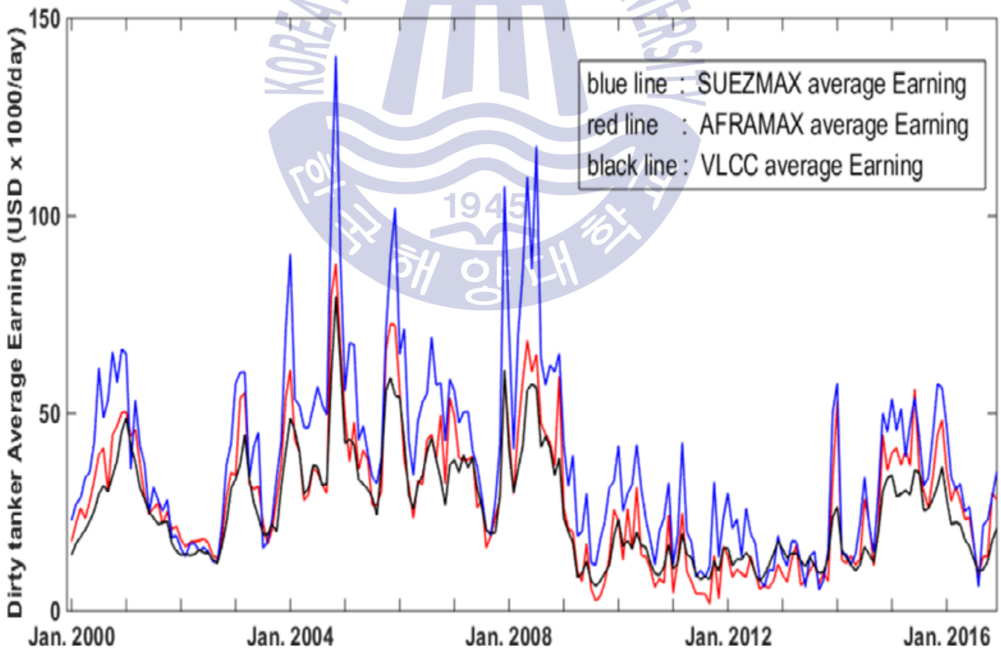

Figure 4.2 Dirty tanker average Earning in time series trend

4.2.1 Prediction performance results for VLCC

Table 4.1 - Table 4.6 and **Figure 4.3 - Figure 4.14** show the prediction results for VLCC Earning by the Levenberg-Marquardt algorithm and the Bayesian regularization algorithm. For the representation of the line in the figures, the observation value is represented by a solid black line, and the predicted result is also represented by the solid red line in case of the best performance. And the remained performance results are illustrated by the dashed line.

4.2.1.1 One-month ahead prediction

Table 4.1 shows the performance details of one-month ahead prediction using the Levenberg-Marquardt algorithm and Bayesian regularization algorithm.

In the Bayesian regularization algorithm, as the size of the hidden layer neuron increased, the iteration increased, the computing time increased, and the gradient value and train performance error (MSE) converged to a smaller value. Also, the effective number of parameters in 8 neurons or 10 neurons was about 60% - 70% of the total number of parameters. Increasing the test sampling data size from 15% to 20% did not show any significant change in performance results. In addition, the overall mean performance index was not improved. The train, validation and test performance errors by the Levenberg-Marquardt algorithm are comparable in size, indicating no overfitting or extrapolation.

In the LMA.TDS-15.NN-9 network, the performance error of the network is 12.22. In the BRA.TDS-15.NN-8 network, the results is 6.7013, and 6.7026 in the BRA.TDS-15.NN-10 network, 8.79 in the BRA.TDS-20.NN-10 network. The mean performance error of the BRA.TDS-15.NN-8 network and the BRA.TDS-15.NN-10 network are almost the same values, but the train performance error of the BRA.TDS-15.NN-10 network is 0.266, which is much lower than the BRA.TDS-15.NN-8 network of 1.3874 as shown in **Table 4.1**.

However, when comparing the ANN performance results of these two networks, in the case of that the mean performance error (MSE) of the two networks is very similar value but the train performance error (MSE) of one network is more converged to smaller value than the other, the researcher is an option to take a final choice of which the case of smaller training error is adapted as a final prediction results. As shown in **Figure 4.3**, the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value, and it shows a certain time lag.

Table 4.1 Comparison of ANN performance for one-month ahead prediction

VLCC [One-month ahead prediction]	ANN Architecture			
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10	BRA.TDS-20. NN-10	LMA.TDS-15. NN-9
Epoch	279	443	324	12
Computing Time (sec)	5	9	6	-
μ	-	-	-	0.1
Gradient	0.82	0.32	0.69	8.68
Effective number of Parameter (used/total)	120/177	147/221	130/221	-
Regression	0.98218	0.98175	0.97644	0.96639
Mean Performance Error	6.7013	6.7026	8.7922	12.2177
Train Performance Error	1.3874	0.2660	0.7113	9.2426
Validation performance Error	-	-	-	17.0812
Test Performance Error	37.1680	43.6063	41.5199	21.4366

LMA : Levenberg-Marquardt Algorithm

BRA : Bayesian regularization algorithm

TDS- : Test data set for full input data set (%)

NN- : Number of neurons of hidden layer

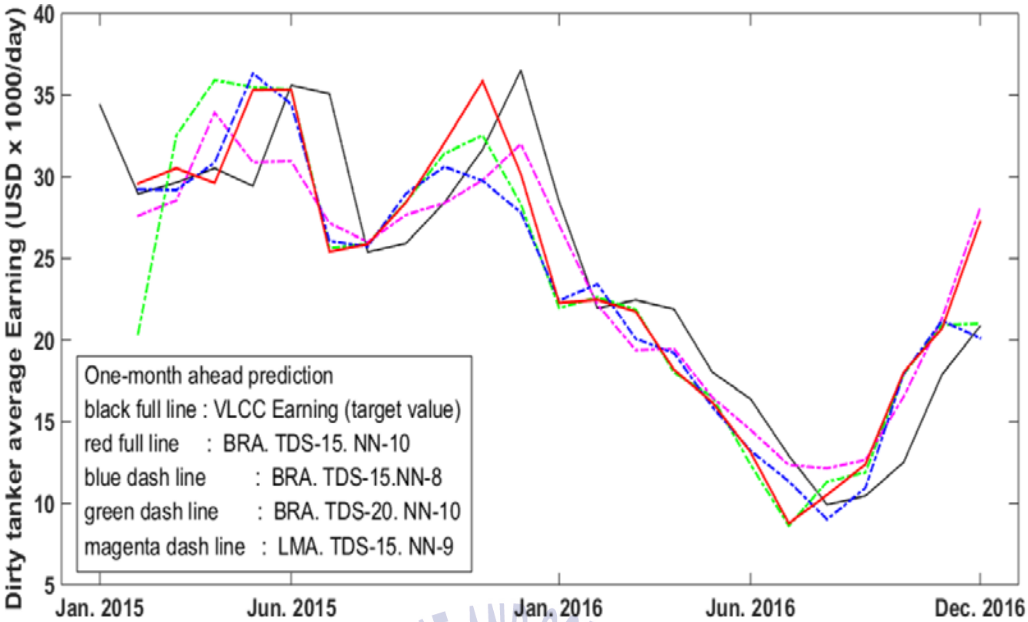

Figure 4.3 One-month ahead prediction

The correlation coefficient between the outputs and targets shows bellow **Figure 4.4** as R value of 0.98175. If R = 1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. The R value of 0.98175 shows that all the data does not fall exactly on the regression line, but the variation is pretty much small.

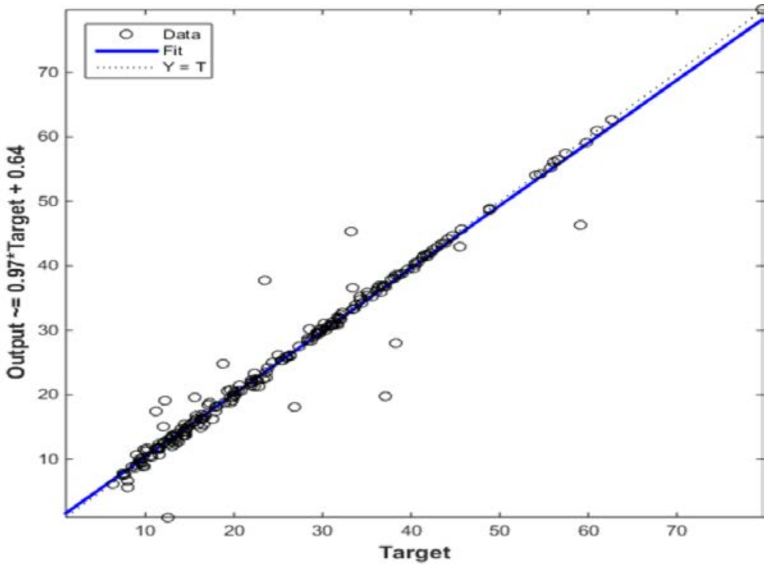

Figure 4.4 Regression between outputs and targets

For the prediction error to be uncorrelated, the autocorrelation function should be an impulse at $\tau = 0$, with all other values equal to 0. However in fact the values at $\tau \neq 0$ is never be exactly equal to zero because of white noise. The dashed red lines in **Figure 4.5** indicate the confidence bounds. The estimated autocorrelation function for the prediction errors falls outside these confidence bounds at a number of points.

This indicates that it may need to increase the length of the tapped delay line, which was set to 2. However, in this study, we applied the external adjustment variables as constant as possible to compare the results of the training algorithm according to ANN architecture. And the correlation between the prediction errors and the input sequence in **Figure 4.6** shows that it does not fall outside the confidence bounds at any points.

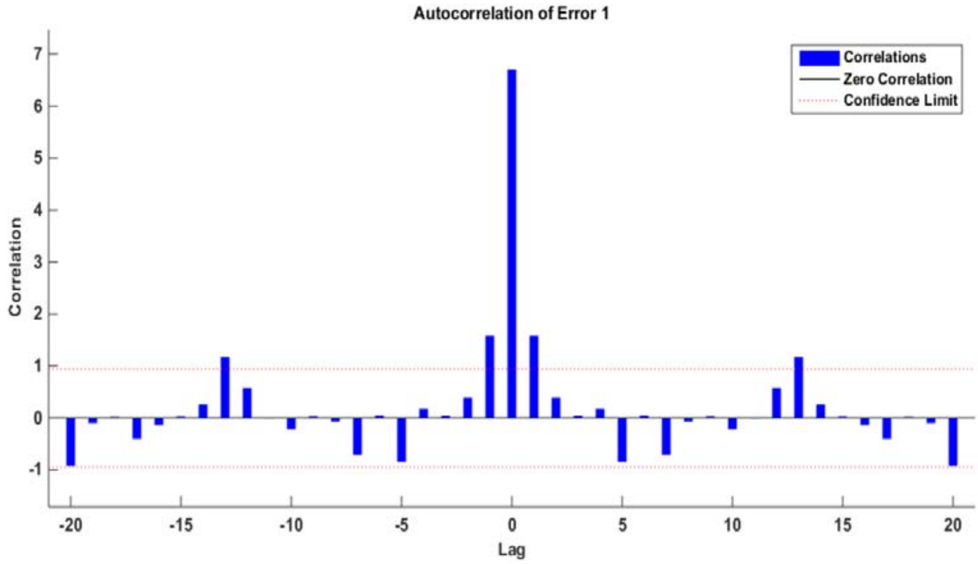

Figure 4.5　Autocorrelation of errors after 1-month prediction training

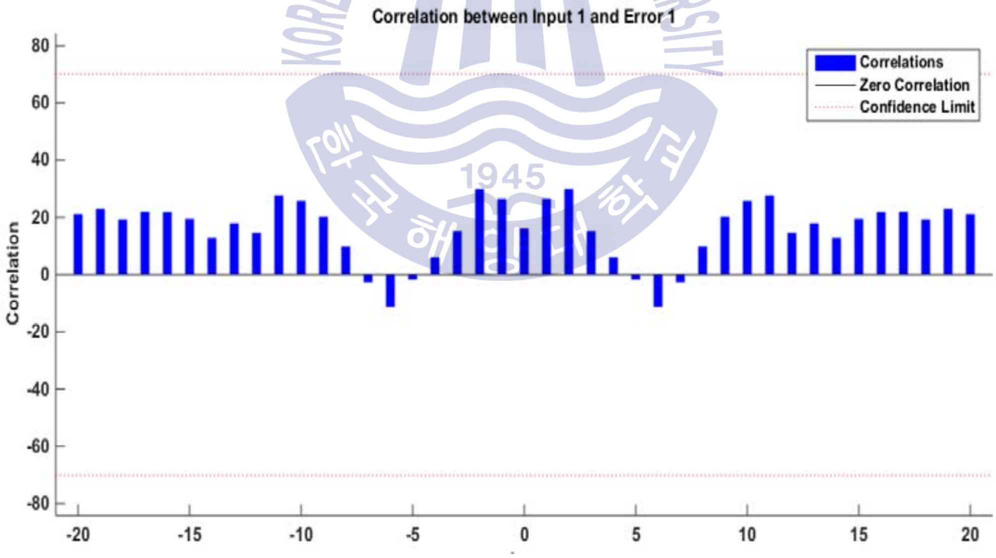

Figure 4.6　Correlation between input and errors

Figure 4.7 illustrates the training mean squared error versus iteration number which shows that the reduction in the training performance index per iteration remain almost constant for a number of iterations before stopping. The minimum training error occurred at iteration 442 and indicated by the circle. It can be verified as a stable convergence.

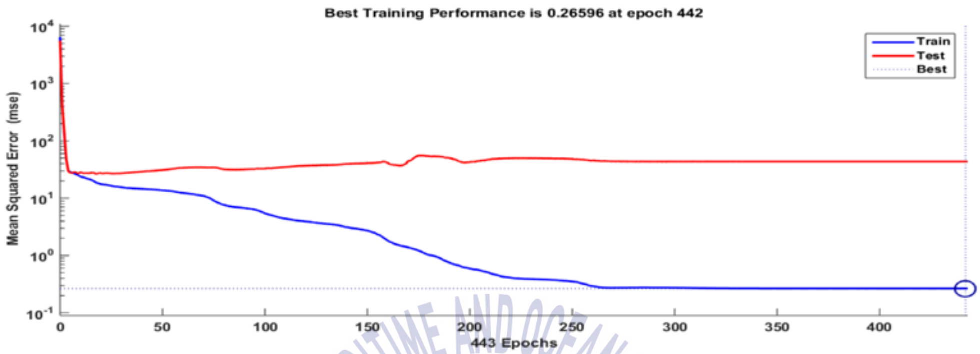

Figure 4.7 Training mean squared error vs. iteration number

The **Figure 4.8** shows the adjusted network parameters to optimize the performance of the network at the final stage of 442 iterations. It can be confident that a stable convergence is illustrated.

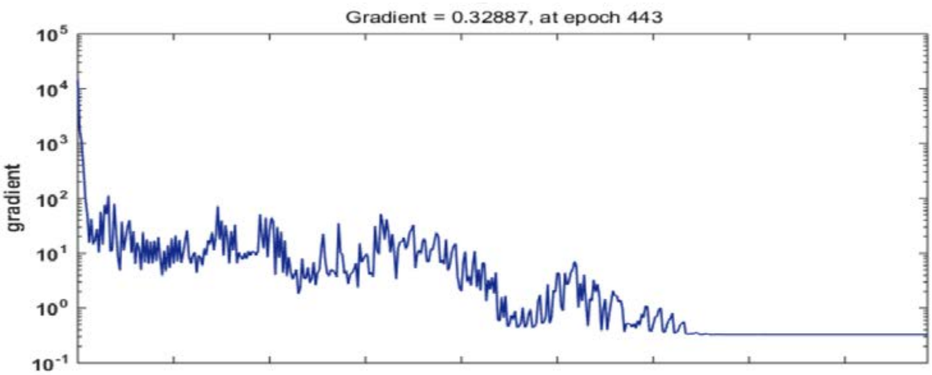

Figure 4.8 Conjugate gradient of parameters

The **Figure 4.9** shows the variation of the effective number of parameters and convergence to 146.8694 at the 442 iterations. The training algorithms have a total of 221 parameters in this 9 (input) -10 (number of neurons of the hidden layer) -1 (output) network, so this network was using about 66.5% of the weights and biases.

This training algorithm is insured that the number of parameters needed for the training performance were used effectively.

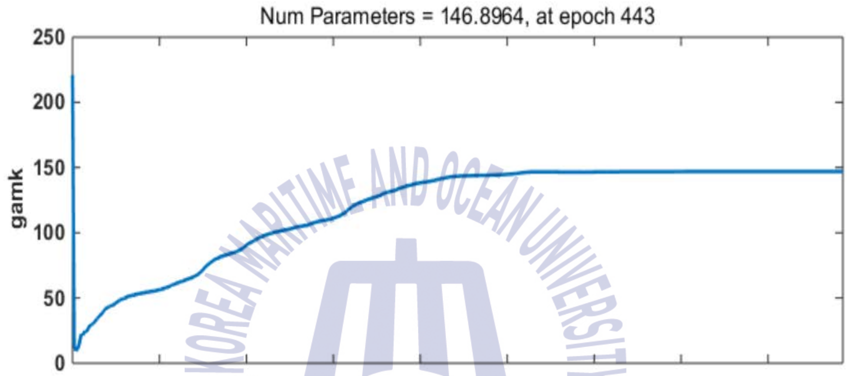

Figure 4.9 Effective number of parameters

4.2.1.2 3-months ahead prediction

Table 4.2 shows the performance details of 3-months ahead prediction using the Levenberg-Marquardt algorithm and Bayesian regularization algorithm.

In the Bayesian regularization algorithm, as the size of the hidden layer neuron increased, the iteration increased, the 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 error of 8 neurons as 8.6425 show better results than 8.9288 of 10 neurons in the same forecasting horizon. Also, the effective number of parameters in 8 neurons or 10 neurons was about 55% -65% of the total number of parameters.

When increasing the test sampling data size from 15% to 20% in the Bayesian regularization algorithm did not show any significant change in performance results. In addition, the overall mean performance index was not improved. The results of the correlation coefficient between the outputs and targets show that the test sampling data size 20% is worse than 15%.

The train, validation and test performance errors by the Levenberg-Marquardt algorithm show a significant error in size even though no indication overfitting or extrapolation.

In **Figure 4.10** of 3-month ahead prediction, the network of BRA.TDS-15.NN-8 has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value. Forecasts of market uptrends around June 2015 tend to be unstable but show better behavior during market recession.

Table 4.2 Comparison of ANN performance for 3-months ahead prediction

VLCC [3-months ahead prediction]	ANN Architecture			
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10	BRA.TDS-20. NN-10	LMA.TDS-15. NN-9
Epoch	370	707	507	12
Computing Time (sec)	6	13	11	-
μ	-	-	-	0.1
Gradient	1	1.23	0.69	11.30
Effective number of Parameter (used/total)	114/177	121/221	130/221	-
Regression	0.97649	0.97605	0.95460	0.93509
Mean Performance Error	8.6425	8.9288	11.8703	23.8859
Train Performance Error	2.2021	1.9989	0.0737	10.3963
Validation performance Error	-	-	-	56.8623
Test Performance Error	46.6056	48.6599	46.7973	54.7607

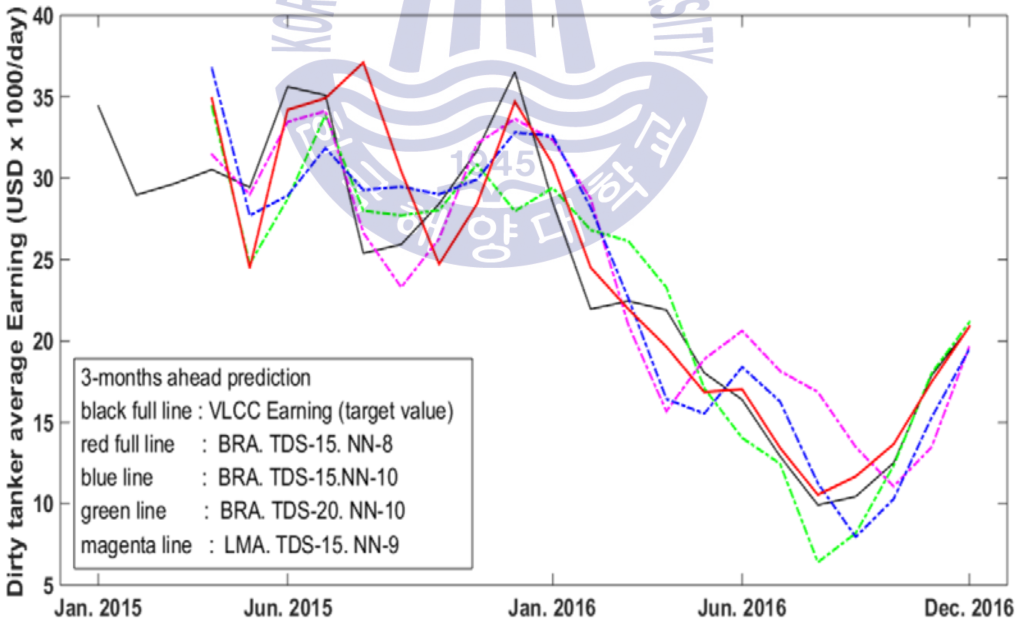

Figure 4.10 3-months ahead prediction

4.2.1.3 6-months ahead prediction

As shown in **Table 4.3**, as the hidden layer neurons are increased from 8 neurons to 10 neurons, the iteration was increased and the gradient value and the train performance error (MSE) converged to a smaller value. And also in the short term ahead prediction of six months, the mean performance error of 8 neurons as 7.8063 show better results than 8.9288 of 10 neurons and other training algorithm in the same forecasting horizon.

Also, the effective number of parameters in 8 neurons or 10 neurons was about 55% - 65% of the total number of parameters. As a singular point, in the case of the BRA.TDS-15.NN-10 3-months-ahead prediction and the BRA.TDS-15.NN-10 6-months-ahead prediction, all results were the same except for multistep ahead performance error (MSE). This seems to be due to the fact that the random sampling points were almost identical.

When increasing the test sampling data size from 15% to 20% in the Bayesian regularization algorithm did not show any significant change in performance results. In addition, the results of the correlation coefficient between the outputs and targets show that the test sampling data size 20% is worse than 15%. The train, validation and test performance errors by the Levenberg-Marquardt algorithm show a significant error in size even though no indication overfitting or extrapolation.

In **Figure 4.11** of 6-months ahead prediction, the network of BRA.TDS-15.NN-8 has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value. However, In the case of sudden up and down changes of market, unstable prediction is shown.

Table 4.3 Comparison of ANN performance for 6-months ahead prediction

VLCC [6-months ahead prediction]	ANN Architecture			
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10	BRA.TDS-20. NN-10	LMA.TDS-15. NN-8
Epoch	347	707	662	17
Computing Time (sec)	6	15	13	-
μ	-	-	-	0.001
Gradient	0.99	1.23	0.27	37.80
Effective number of Parameter (used/total)	110/177	121/221	143/221	-
Regression	0.97983	0.97605	0.95778	0.92713
Mean Performance Error	7.8063	8.9288	15.4979	27.5454
Train Performance Error	2.0636	1.9989	0.1451	15.7874
Validation performance Error	-	-	-	56.4943
Test Performance Error	40.7316	48.6599	77.6767	54.2511

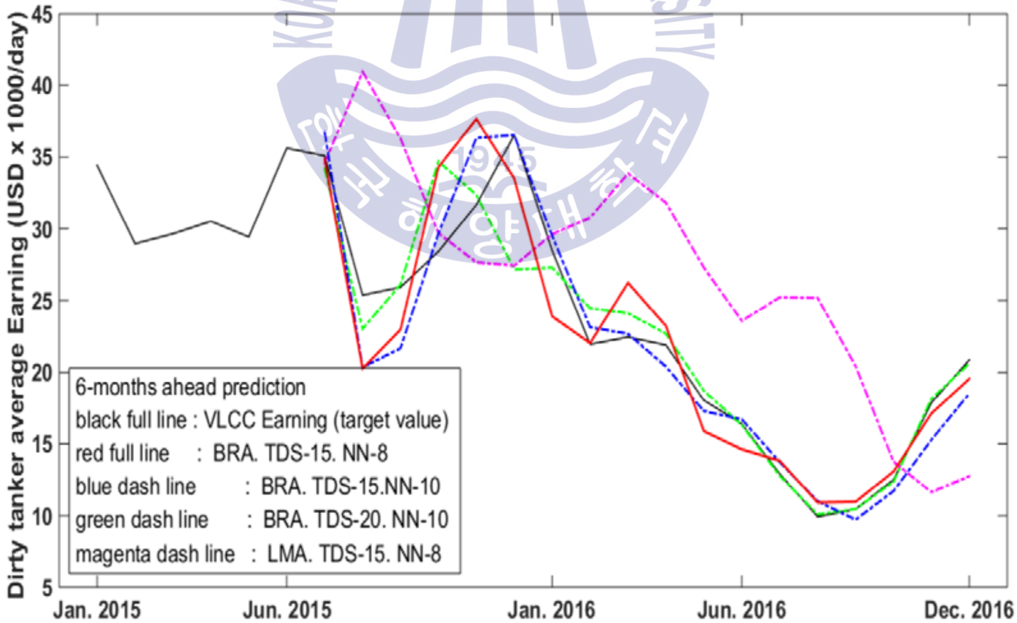

Figure 4.11 6-months ahead prediction

4.2.1.4 9-months ahead prediction

As shown in **Table 4.4**, as the hidden layer neurons were increased from 8 neurons to 10 neurons, the iteration was increased, and the gradient value and the train performance error (MSE) converged to a smaller value. And also the size of the hidden layer neurons increased in the short term ahead prediction, the mean performance error of 8 neurons as 6.7013 showed better results than 7.6663 of 10 neurons and other training algorithm in the same forecasting horizon.

Also, the effective number of parameters in 8 neurons or 10 neurons was about 59% - 68% of the total number of parameters. When increasing the test sampling data size from 15% to 20% in BRA did not show any significant change in performance results. In addition, the results of the correlation coefficient between the outputs and targets show that the test sampling data size 20% is worse than 15%. The train, validation and test performance errors by the Levenberg-Marquardt algorithm show a significant error in size even though no indication overfitting or extrapolation.

In **Figure 4.12** of 9-months ahead prediction, the network of BRA.TDS-15.NN-8 has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value. However, In the case of sudden up and down changes of market, unstable prediction is shown also with the 6-months ahead prediction.

Table 4.4 Comparison of ANN performance for 9-months ahead prediction

VLCC [9-months ahead prediction]	ANN Architecture			
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10	BRA.TDS-20. NN-10	LMA.TDS-15. NN-8
Epoch	279	247	332	15
Computing Time (sec)	4	5	6	-
μ	-	-	-	0.01
Gradient	0.82	0.60	0.59	99.60
Effective number of Parameter (used/total)	120/177	136/221	130/221	-
Regression	0.9828	0.97937	0.97663	0.94738
Mean Performance Error	6.7013	7.6663	8.5568	19.2948
Train Performance Error	1.3874	0.6623	0.5784	7.5927
Validation performance Error	-	-	-	44.3444
Test Performance Error	37.1680	47.8225	40.8693	49.6353

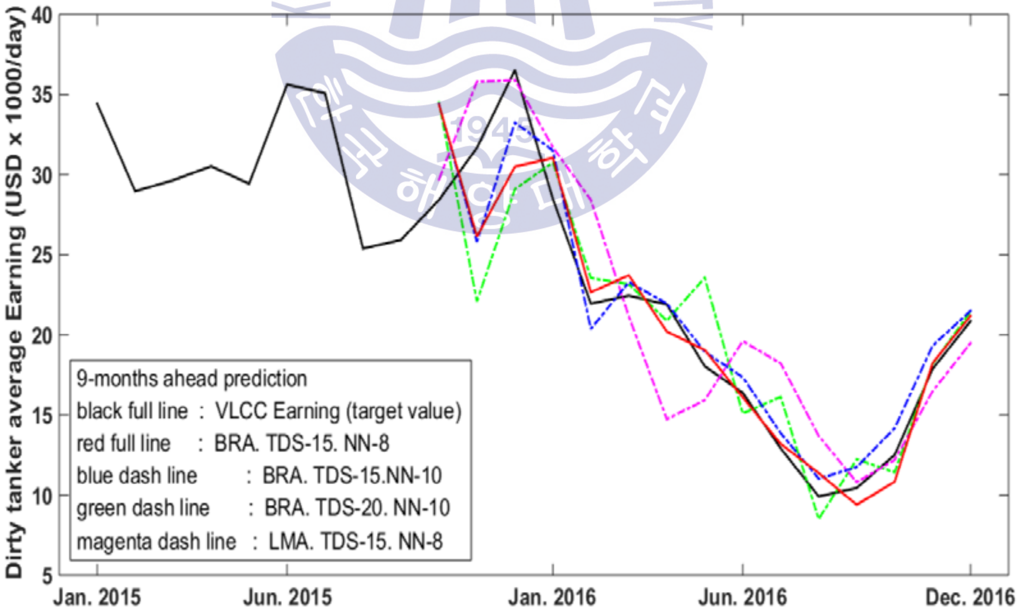

Figure 4.12 9-months ahead prediction

4.2.1.5 12-months ahead forecast

As shown in **Table 4.5**, as the hidden layer neurons were increased from 8 neurons to 10 neurons, the iteration was increased, and the gradient value and the train performance error (MSE) converged to a smaller value. The mean performance error of 10 neurons in 12-months as 4.9389 showed better results than 9.3915 of 8 neurons and other training algorithm in the same forecasting horizon. As shown here, in the long term ahead forecasting such as 12-months ahead prediction, the training algorithm which has a larger size of the hidden layer neuron exhibits better forecasting performance than the smaller.

Also the effective number of parameters in 8 neurons or 10 neurons was about 33% - 70% of the total number of parameters, and BRA.TDS-20.NN-20 network showed that the number of parameters used was drastically reduced to 33%. When increasing the test sampling data size from 15% to 20% did not show any significant change in performance results. In addition, the results of the correlation coefficient between the outputs and targets show that the test sampling data size 20% is worse than 15%. The train, validation and test performance errors by the Levenberg-Marquardt algorithm show a significant error in size even though no indication overfitting or extrapolation.

In the **Figure 4.13** of 12-months ahead prediction, the network of BRA.TDS-15.NN-10 has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value. However, In the case of sudden up and down changes of market, unstable prediction is shown also with the 3, 6, 9-months ahead prediction.

Table 4.5 Comparison of ANN performance for 12-months ahead prediction

VLCC [12-months ahead prediction]	ANN Architecture			
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10	BRA.TDS-20. NN-10	LMA.TDS-15. NN-9
Epoch	310	639	622	15
Computing Time (sec)	5	13	13	-
μ	-	-	-	0.1
Gradient	0.55	0.39	2.91	4.95
Effective number of Parameter (used/total)	124/177	146/221	73.3/221	-
Regression	0.97443	0.98681	0.9599	0.95092
Mean Performance Error	9.3915	4.9389	14.5221	17.8695
Train Performance Error	0.8942	0.3323	8.3127	6.6971
Validation performance Error	-	-	-	39.4219
Test Performance Error	58.1091	31.3500	39.6702	49.1998

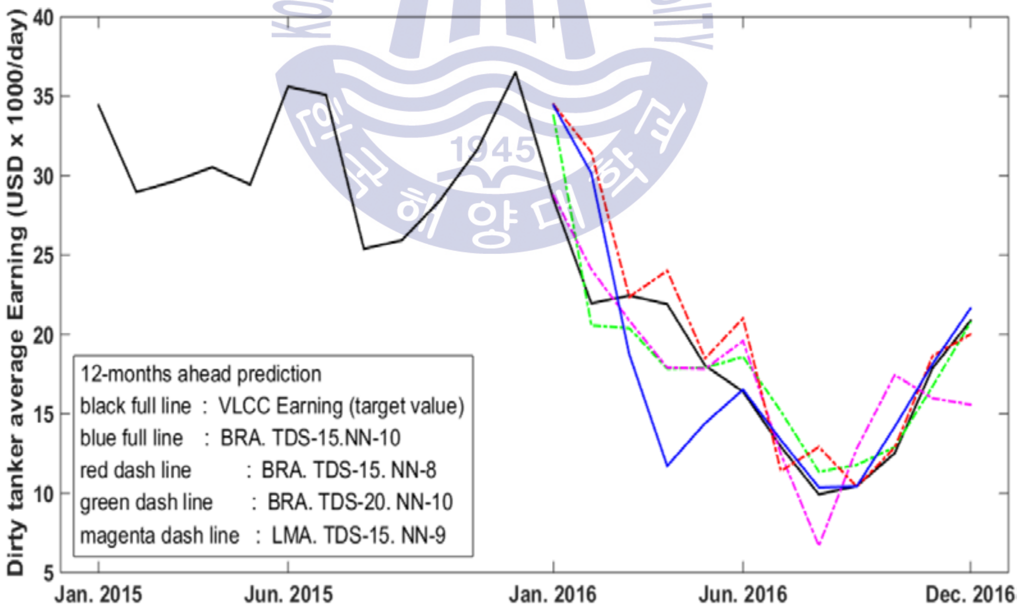

Figure 4.13 12-months ahead prediction

4.2.1.6 15-months ahead prediction

As shown in **Table 4.6**, as the hidden layer neurons were increased from 8 neurons to 10 neurons, the iteration was increased, and the gradient value and the train performance error (MSE) converged to a smaller value. The mean performance error of 10 neurons in 15-months as 8.243 showed better results than 10.5656 of 8 neurons and other training algorithm in the same forecasting horizon. As shown here, there is a similar performance results with the 12-month ahead prediction such as long-term ahead forecasting, the training algorithm which has a larger size of the neuron exhibits better forecasting performance than the smaller.

Also, the effective number of parameters in 8 neurons or 10 neurons was about 23% - 67% of the total number of parameters and BRA.TDS-20.NN-20 network showed that the number of parameters used was drastically reduced to 23%. When increasing the test sampling data size from 15% to 20% did not show any significant change in performance results. In addition, the results of the correlation coefficient between the outputs and targets show that the test sampling data size 20% is worse than 15%.

The train, validation and test performance errors by the Levenberg-Marquardt algorithm show a significant error in size even though no indication overfitting or extrapolation.

In the **Figure 4.14** of 15-months ahead prediction, BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the recession tendency of the observation value.

Table 4.6 Comparison of ANN performance for 15-months ahead prediction

VLCC [15-months ahead prediction]	ANN Architecture			
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10	BRA.TDS-20. NN-10	LMA.TDS-15. NN-6
Epoch	229	341	1000	15
Computing Time (sec)	4	7	24	-
μ	-	-	-	0.01
Gradient	0.83	0.29	3.32	30.30
Effective number of Parameter (used/total)	116/177	149/221	53/221	-
Regression	0.97111	0.97762	0.95311	0.94432
Mean Performance Error	10.5656	8.2430	16.9515	20.1285
Train Performance Error	1.5528	0.2142	12.2908	11.0865
Validation performance Error	-	-	-	42.1246
Test Performance Error	62.2390	54.2748	35.8274	40.9310

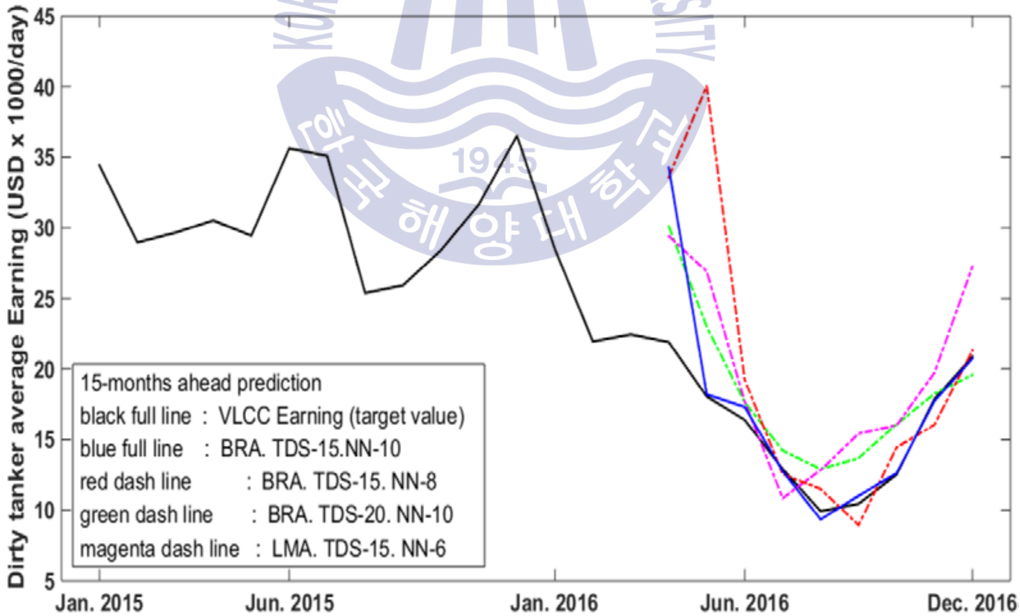

Figure 4.14 15-months ahead prediction

4.2.2 Prediction performance results for SUEZMAX

Table 4.7 - Table 4.12 and **Figure 4.15 - Figure 4.20** shows the prediction results for SUEZMAX Earning by the Bayesian regularization algorithm. For the representation of the line in the figures, the observation value is represented by a solid black line, and the predicted result is also represented by the solid red line in case of the best performance. And the remained performance results are illustrated by the dashed line.

4.2.2.1 One-month ahead prediction

In **Table 4.7** of one-month ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 31.1102 in 10 neurons in the hidden layer and 34.3612 in the BRA.TDS-15.NN-8 network. In the one-month ahead prediction of SUEZMAX, which has the greatest market fluctuation among three tanker types, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. As the number of the hidden layer neuron increased, the gradient value and train performance error (MSE) converged to a smaller value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.9722 in the 10 neurons hidden layer which is better performance than 8 neurons of 0.96816. In the case of 10 neurons in the hidden layer, the effective number of parameters was 127, which was only 55.9% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 110, 62.1% of total training algorithm parameters of 177.

As shown in **Figure 4.15** of 1-month ahead prediction, the performance results of the network of BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value, and it shows a certain time lag.

Table 4.7 Comparison of ANN performance for One-month ahead Prediction

SUEZMAX [One-month Ahead Prediction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	443/9	331/6
Gradient	7.97	4.97
Effective number of Parameter (used/total)	110/177	127/221
Regression	0.96816	0.97220
Mean Performance Error	34.3612	31.1102
Train Performance Error	17.4391	7.4387
Test Performance Error	131.3814	166.8265

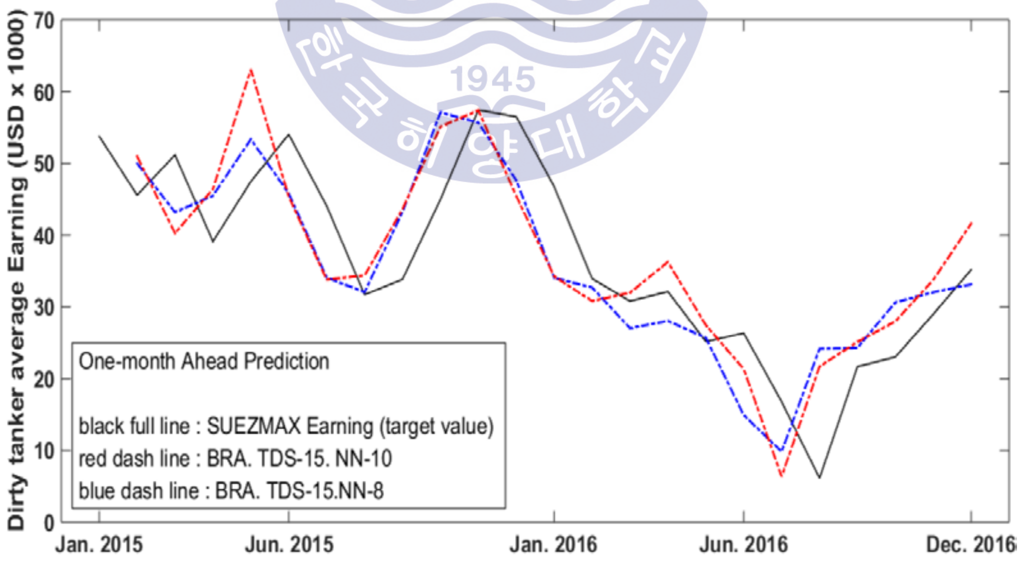

Figure 4.15 One-month ahead prediction

4.2.2.2 3-months ahead prediction

In **Table 4.8** of 3-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 35.5070 in 10 neurons in the hidden layer and 38.9666 in the BRA.TDS-15.NN-8 network. In the 3-months ahead prediction of SUEZMAX, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. As the number of the hidden layer neuron increased, the iteration and computing time increased, and the gradient value and train performance error (MSE) converged to a smaller value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.96815 in the 10 neurons hidden layer which is better performance than 8 neurons of 0.96387.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 121, which was only 54.8% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 104, 58.8% of total training algorithm parameters of 177.

As shown in **Figure 4.16** of 3-month ahead prediction, the performance results of the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value, and it shows good tendency for the overall prediction horizons.

Table 4.8 Comparison of ANN performance for 3-months ahead prediction

SUEZMAX [3-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	500/9	572/11
Gradient	8.78	6.49
Effective number of Parameter (used/total)	104/177	121/221
Regression	0.96387	0.96815
Mean Performance Error	38.9666	35.5070
Train Performance Error	20.7935	10.9874
Test Performance Error	143.1587	176.0862

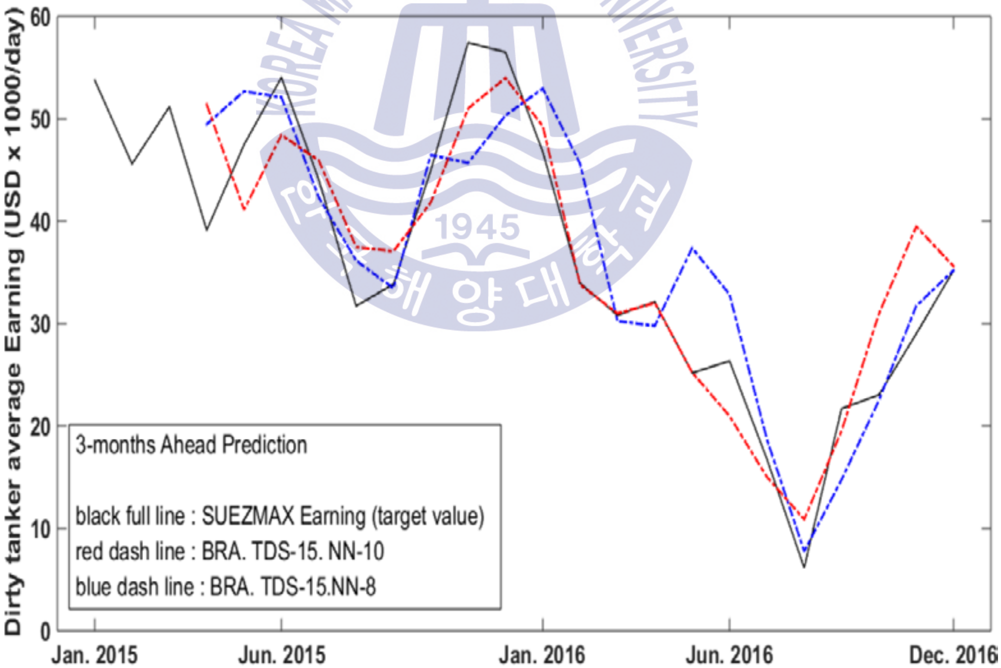

Figure 4.16 3-months ahead prediction

4.2.2.3 6-months ahead prediction

In **Table 4.9** of 6-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 34.3609 in 10 neurons in the hidden layer and 38.8077 in the BRA.TDS-15.NN-8 network. In the 6-months ahead prediction, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. As the number of the hidden layer neuron increased, the gradient value and train performance error (MSE) converged to a smaller value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.96870 in the 10 neurons hidden layer which was better performance than 8 neurons of 0.96419.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 120, which was only 54.3% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 106, 59.9% of total training algorithm parameters of 177.

As shown in the **Figure 4.15** of 6-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value.

Table 4.9　Comparison of ANN performance for 6-months ahead prediction

SUEZMAX [6-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	567/10	493/10
Gradient	8.02	6.63
Effective number of Parameter (used/total)	106/177	120/221
Regression	0.96419	0.96870
Mean Performance Error	38.8077	34.3609
Train Performance Error	18.1697	10.7994
Test Performance Error	157.1321	169.4464

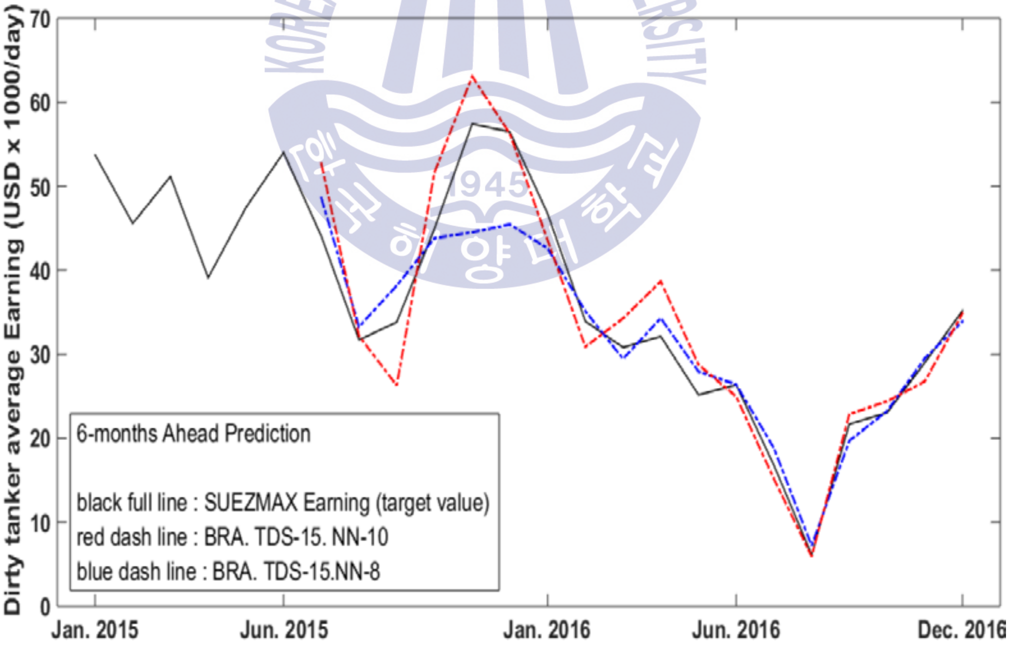

Figure 4.17　6-months ahead prediction

4.2.2.4 9-months ahead prediction

In the **Table 4.10** of 9-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 43.6329 and 65.7947 in the BRA.TDS-15.NN-8 network. In the 9-months ahead prediction, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. However, in the case of the BRA.TDS-15.NN-10 network with 10 number of neurons in the hidden layer, the prediction performance results was more satisfactory than the BRA.TDS-15.NN-8 network with the 8 number of neurons in the hidden layer, but the effective number of parameters was small, and the gradient value and the train performance error (MSE) showed a larger value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.96040 in the 10 neurons hidden layer which was better performance than 8 neurons of 0.94311.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 116, which was only 52.5% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 122, 68.9% of total training algorithm parameters of 177.

As shown in the **Figure 4.18** of 9-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value.

Table 4.10 Comparison of ANN performance for 9-months ahead prediction

SUEZMAX [9-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	585/10	537/10
Gradient	3.93	6.95
Effective number of Parameter (used/total)	122/177	116/221
Regression	0.94311	0.96040
Mean Performance Error	65.7947	43.6329
Train Performance Error	7.2434	12.6922
Test Performance Error	401.4888	221.0258

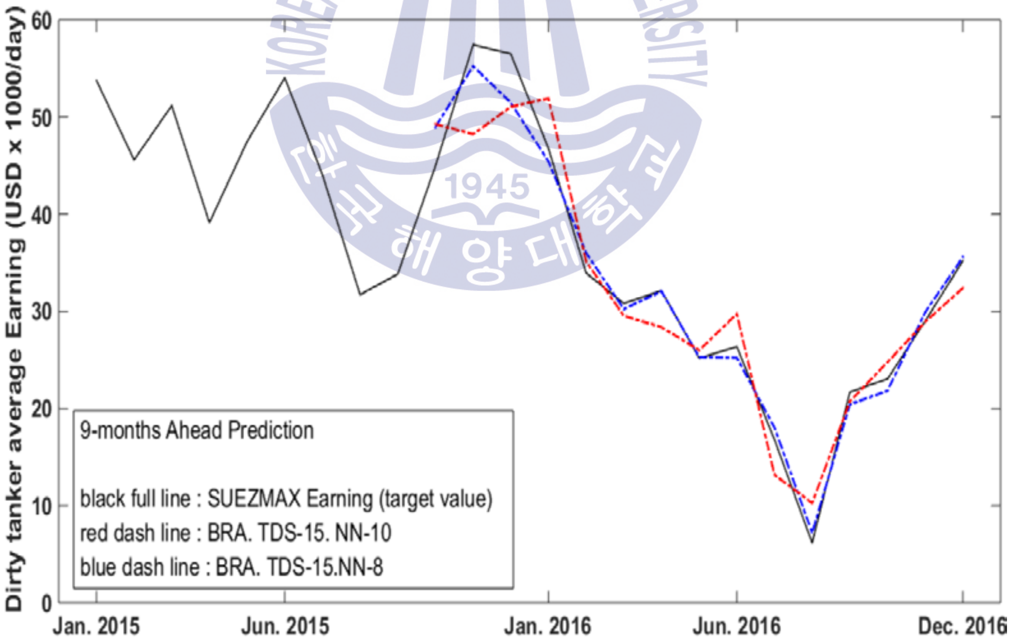

Figure 4.18 9-months ahead prediction

4.2.2.5 12-months ahead prediction

In **Table 4.11** of 12-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 43.5288 and 57.5943 in the BRA.TDS-15.NN-8 network. In the 12-months ahead prediction, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. As the number of the hidden layer neuron increased, the gradient value and train performance error (MSE) converged to a smaller value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.96127 in the 10 neurons hidden layer which was better performance than 8 neurons of 0.94603.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 119, which was only 53.4% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 110, 62.1% of total training algorithm parameters of 177.

As shown in **Figure 4.19** of 12-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value.

Table 4.11 Comparison of ANN performance for 12-months ahead prediction

SUEZMAX [12-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	606/10	440/9
Gradient	6.64	5.82
Effective number of Parameter (used/total)	110	119
Regression	0.94603	0.96127
Mean Performance Error	57.5943	43.5288
Train Performance Error	14.4853	10.0323
Test Performance Error	304.7520	234.9019

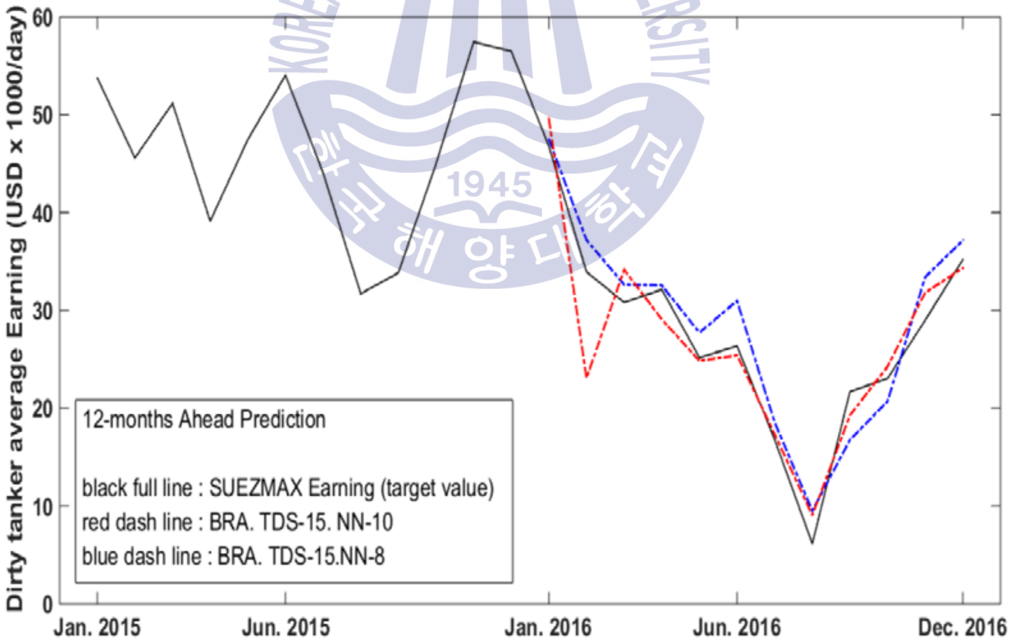

Figure 4.19 12-months ahead prediction

4.2.2.6 15-months ahead prediction

In **Table 4.12** of 15-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 27.3112 and 32.3071 in the BRA.TDS-15.NN-8 network. In the 15-months ahead prediction, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. As the number of the hidden layer neuron increased, the gradient value and train performance error (MSE) converged to a smaller value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.97508 in the 10 neurons hidden layer which was better performance than 8 neurons of 0.97035.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 127, which was only 57.5% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 111, 62.7% of total training algorithm parameters of 177.

As shown in **Figure 4.20** of 15-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value.

Table 4.12 Comparison of ANN performance for 15-months ahead prediction

SUEZMAX [15-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	273/5	361/7
Gradient	6.53	5.83
Effective number of Parameter (used/total)	111/177	127/221
Regression	0.97035	0.97508
Mean Performance Error	32.3070	27.3112
Train Performance Error	14.1581	8.7129
Test Performance Error	136.3610	133.9412

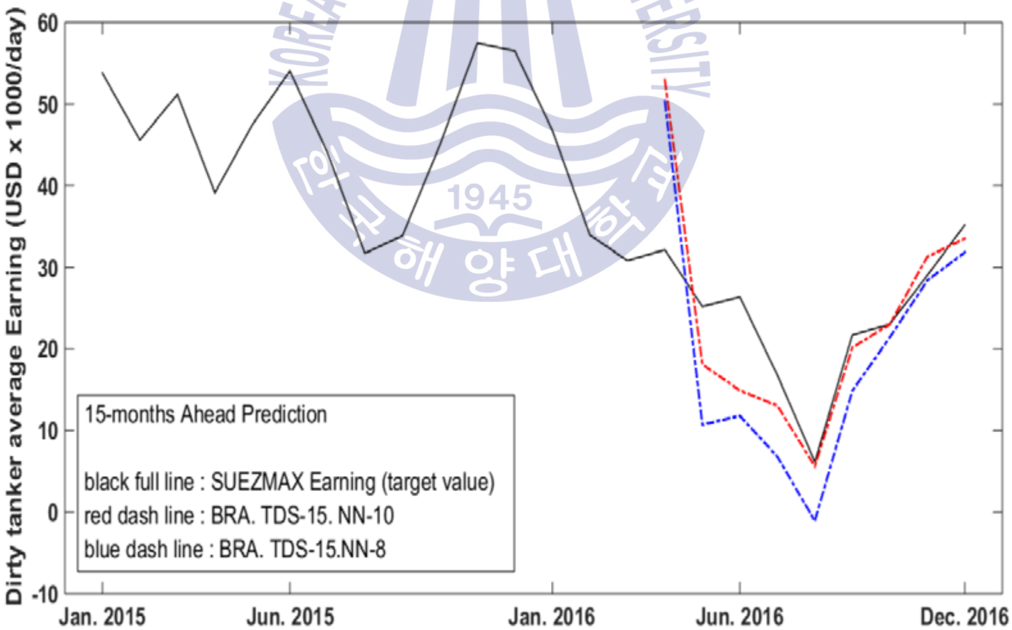

Figure 4.20 15-months ahead prediction

4.2.3 Prediction performance results for AFRAMAX

Table 4.13 - Table 4.18 and **Figure 4.21 - Figure 4.26** shows the prediction results for AFRAMAX Earning by the Bayesian regularization algorithm. For the representation of the line in the figures, the observation value is represented by a solid black line, and the predicted result is also represented by the solid red line in case of the best performance. And the remained performance results are illustrated by the dashed line.

4.2.3.1 One-month ahead prediction

In **Table 4.13** of one-month ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 37.8294 and 31.0876 in the BRA.TDS-15.NN-8 network. In the one-month ahead prediction of AFRAMAX, which has a moderate market fluctuation among three tanker types, the BRA.TDS-15.NN-8 network showed more satisfactory prediction results than the BRA.TDS-15.NN-10 network. Also in the BRA.TDS-15.NN-8 network, the gradient value and the train performance error (MSE) showed a better value than BRA.TDS-15.NN-10 netwok.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.94664 in the 8 neurons hidden layer which was better performance than 10 neurons of 0.93433.

In the case of 8 neurons in the hidden layer, the effective number of parameters was 105, which was only 59.3% of total training algorithm parameters of 177. Also in the case of 10 neurons in the hidden layer, the effective number of parameters was 98.6, 44.6% of total training algorithm parameters of 221. As shown in **Figure 4.21** of 1-month ahead prediction, the performance results of the BRA.TDS-15.NN-8 network has appeared good convergence with the total course of the peaks and bottoms of the observation value, and it shows a certain time lag.

Table 4.13 Comparison of ANN performance for one-month ahead prediction

AFRAMAX [One-month Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	469/8	447/8
Gradient	3.23	5.3
Effective number of Parameter (used/total)	105/177	98.6/221
Regression	0.94664	0.93433
Mean Performance Error	31.0876	37.8294
Train Performance Error	7.8594	14.1181
Test Performance Error	164.2627	173.7742

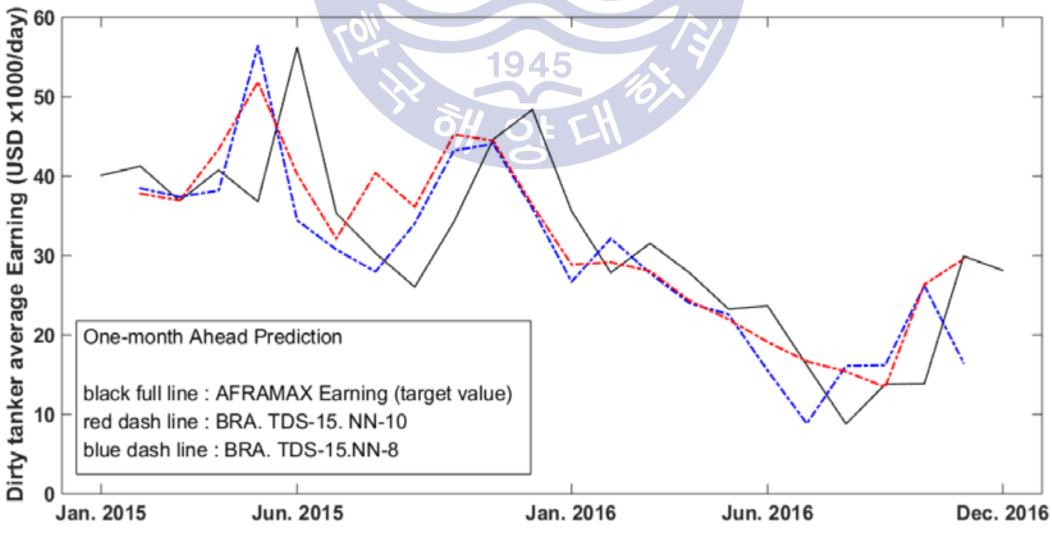

Figure 4.21 one-month ahead prediction

4.2.3.2 3-months ahead prediction

In **Table 4.14** of one-month ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 29.9537 and 27.7912 in the BRA.TDS-15.NN-8 network. In the one-month ahead prediction, the BRA.TDS-15.NN-8 network showed more satisfactory prediction results than the BRA.TDS-15.NN-10 network. Also in the BRA.TDS-15.NN-8 network, the gradient value and the train performance error (MSE) showed a better value than the BRA.TDS-15.NN-10 network.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.95312 in the 8 neurons hidden layer which was better performance than 10 neurons of 0.94814.

In the case of 8 neurons in the hidden layer, the effective number of parameters was 118, which was only 66.7% of total training algorithm parameters of 177. Also in the case of 10 neurons in the hidden layer, the effective number of parameters was 111, 50.2% of total training algorithm parameters of 221.

As shown in **Figure 4.22** of 3-months ahead prediction, the performance results of the BRA.TDS-15.NN-8 network has appeared good convergence with the total course of the peaks and bottoms of the observation value.

Table 4.14 Comparison of ANN performance for 3-months ahead prediction

AFRAMAX [3-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	563/9	908/18
Gradient	2.74	3.86
Effective number of Parameter (used/total)	118/177	111/221
Regression	0.95312	0.94814
Mean Performance Error	27.7912	29.9537
Train Performance Error	5.7924	7.7699
Test Performance Error	153.9224	157.1411

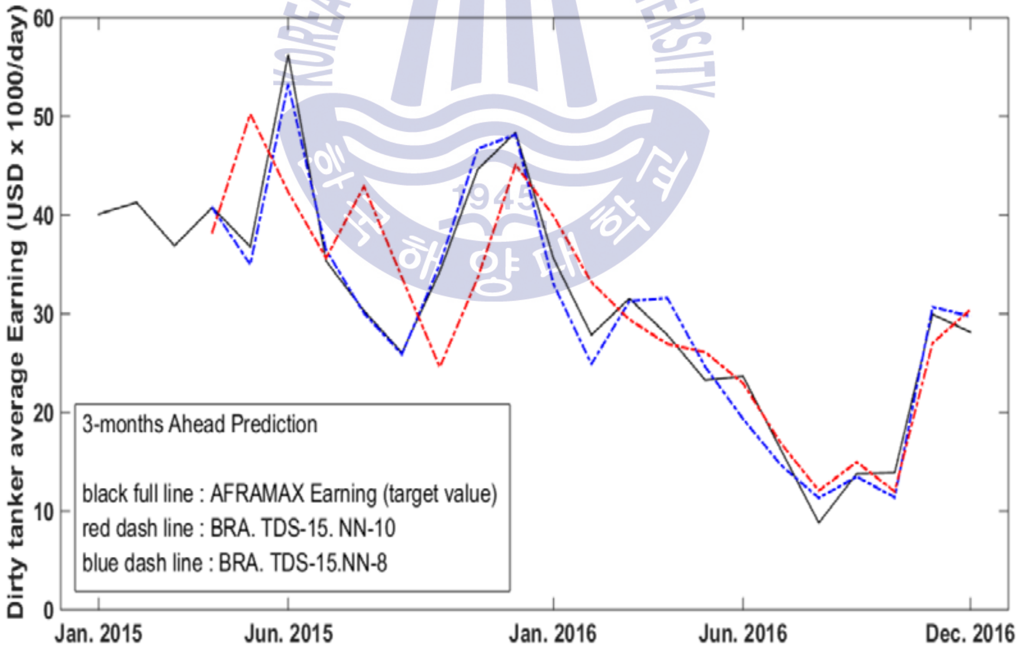

Figure 4.22 3-months ahead prediction

4.2.3.3 6-months ahead prediction

In **Table 4.15** of 6-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 29.3719 and 37.5551 in the BRA.TDS-15.NN-8 network. In the 6-months ahead prediction, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. When the number of hidden layer neurons increased from 8 to 10, the gradient value converged to a similar value in the two network and train performance error (MSE) converged to a smaller value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.94922 in the 10 neurons hidden layer which was better performance than 8 neurons of 0.93441.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 105, which was only 47.5% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 94.1, 53.2% of total training algorithm parameters of 177.

As shown in **Figure 4.23** of 6-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks an

Table 4.15 Comparison of ANN performance for 6-months ahead prediction

AFRAMAX [6-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	449/7	484/9
Gradient	4.61	4.83
Effective number of Parameter (used/total)	94.1/177	105/221
Regression	0.93441	0.94922
Mean Performance Error	37.5550	29.3719
Train Performance Error	15.5525	11.2500
Test Performance Error	180.9028	133.2707

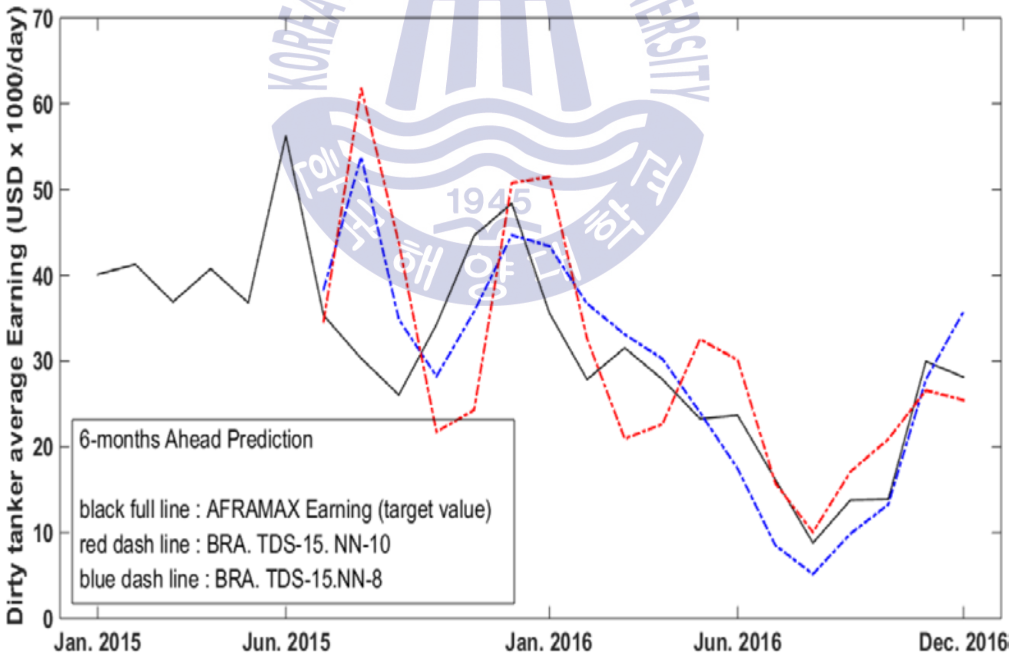

Figure 4.23 6-months ahead prediction

4.2.3.4 9-months ahead prediction

In **Table 4.16** of 9-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 32.6568 and 33.5015 in the BRA.TDS-15.NN-8 network. In the 9-months ahead prediction, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. However, even though the number of hidden layer neurons increased from 8 to 10, the gradient value and train performance error (MSE) converged to a similar value in the two network.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.94326 in the 10 neurons hidden layer which was better performance than 8 neurons of 0.94177.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 82.8, which is only 37.5% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 82.3, 46.5% of total training algorithm parameters of 177.

As shown in the **Figure 4.24** of 9-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network has appeared relatively good convergence with the total course of the peaks and bottoms of the observation value.

Table 4.16　Comparison of ANN performance for 9-months ahead prediction

AFRAMAX [9-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	533/9	1000/27
Gradient	6.42	6.64
Effective number of Parameter (used/total)	82.3/177	82.8/221
Regression	0.94177	0.94326
Mean Performance Error	33.5015	32.6568
Train Performance Error	20.7368	21.4463
Test Performance Error	106.6858	96.9302

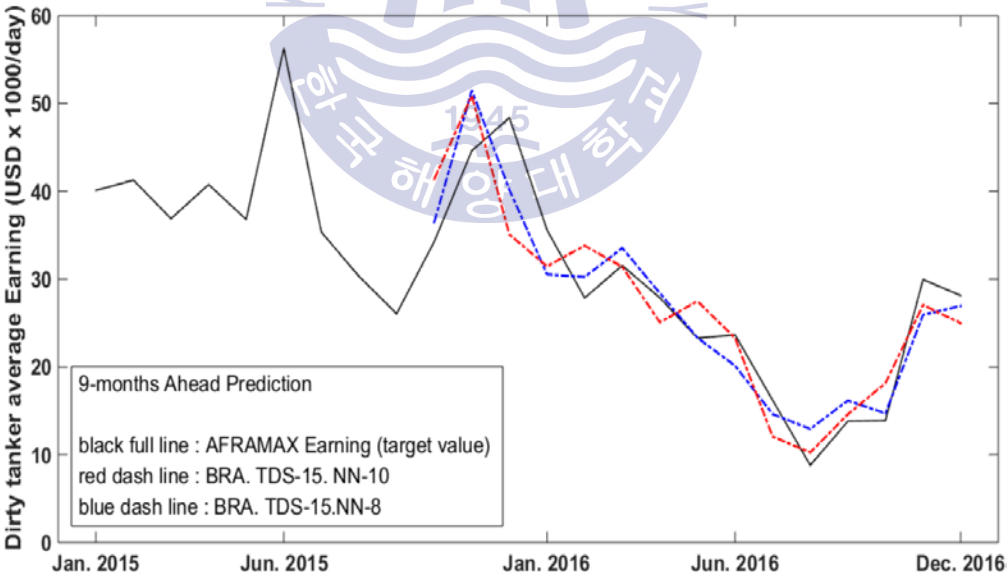

Figure 4.24　9-months ahead prediction

4.2.3.5 12-months ahead prediction

In **Table 4.17** of 12-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 30.9197 and 43.4286 in the BRA.TDS-15.NN-8 network. In the 12-months ahead prediction, the BRA.TDS-15.NN-10 network showed more satisfactory prediction results than the BRA.TDS-15.NN-8 network. As the number of hidden layer neurons increased from 8 to 10, the gradient value and train performance error (MSE) converged to a smaller value.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.94668 in the 10 neurons hidden layer which was better performance than 8 neurons of 0.93399.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 90.8, which is only 41.1% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 70.9, 40.1% of total training algorithm parameters of 177.

As shown in **Figure 4.25** of 12-months ahead prediction, The performance results of the "BRA.TDS-15.NN-10" shows good tendency with the total course of the peaks and bottoms of the observation values.

Table 4.17 Comparison of ANN performance for 12-months ahead prediction

AFRAMAX [12-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	725/15	1000/21
Gradient	7.44	6.14
Effective number of Parameter (used/total)	70.9/177	90.8/221
Regression	0.93399	0.94668
Mean Performance Error	43.4286	30.9197
Train Performance Error	29.3808	17.7166
Test Performance Error	123.9697	106.6175

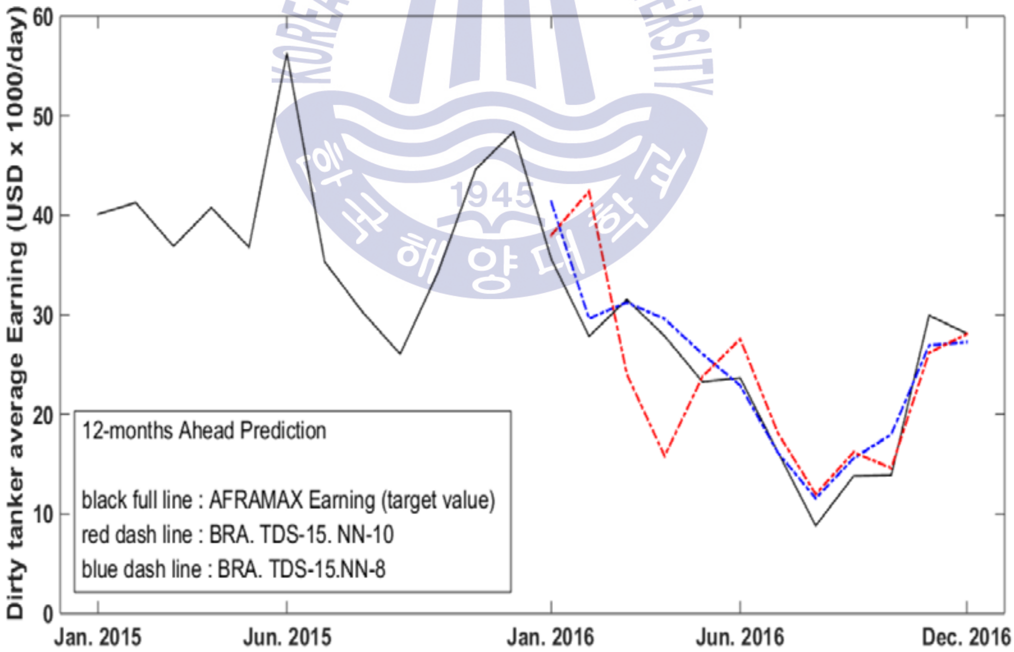

Figure 4.25 12-months ahead prediction

4.2.3.6 15-months ahead prediction

In **Table 4.18** of 15-months ahead prediction, the performance results of the BRA.TDS-15.NN-10 network was 31.2048 and 29.0555 in the BRA.TDS-15.NN-8 network. In the 15-months ahead prediction, the BRA.TDS-15.NN-8 network showed more satisfactory prediction results than the BRA.TDS-15.NN-10 network. When the number of hidden layer neurons increased from 8 to 10, the gradient value and the train performance error (MSE) converged to a smaller value, but the performance error increased.

The R value for correlation between the outputs and targets was in the acceptable boundary of 0.94947 in the 8 neurons hidden layer which was better performance than 10 neurons of 0.94556.

In the case of 10 neurons in the hidden layer, the effective number of parameters was 114, which was only 51.6% of total training algorithm parameters of 221. Also in the case of 8 neurons in the hidden layer, the effective number of parameters was 91.8, 51.9% of total training algorithm parameters of 177.

As shown in **Figure 4.26** of 15-months ahead prediction, The performance results of the BRA.TDS-15.NN-8 network shows good tendency with the total course of the peaks and bottoms of the observation values.

Table 4.18 Comparison of ANN performance for 15-months ahead prediction

AFRAMAX [15-months Ahead Perdiction]	ANN Architecture	
	BRA.TDS-15. NN-8	BRA.TDS-15. NN-10
Epoch/computing time (sec)	445/8	875/18
Gradient	6.05	4.07
Effective number of Parameter (used/total)	91.8/177	114/221
Regression	0.94947	0.94556
Mean Performance Error	29.0555	31.2048
Train Performance Error	17.9484	5.3082
Test Performance Error	92.7367	162.4785

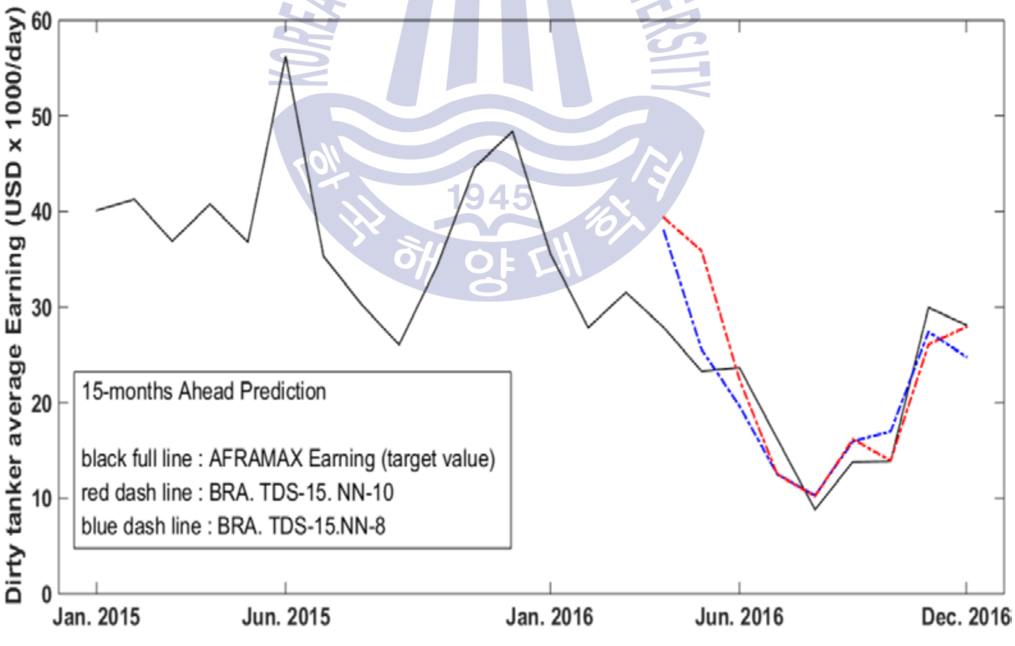

Figure 4.26 15-months ahead prediction

4.2.4 Comparison for different hidden layer size

In this study, we modified the hidden layer size with the Bayesian regularization algorithm and compared the effect of hidden layer size on network performance by implementing one-month ahead prediction. The performance results such as iterations, gradient, training parameters and performance errors are summarized in **Table 4.19** and **Figure 4.27** shows the prediction results according to the hidden layer size. Increasing the size of the hidden layer neurons increased iteration and increased computing time. The gradient value and the train performance error (MSE) converge to a smaller value. However, even with increasing neuron size, the R-values did not decrease and mean performance errors did not decrease.

The mean performance error (MSE) were minimized when the size of the neurons was 8 and 10, and those of the neurons were almost equal to the size of the input variables (i.e. 9). The effective number of parameters at this best performance was 67.8% and 66.5% of the total training algorithm parameters.

Table 4.19 Comparison for different hidden layer size

[One-month ahead prediction]	Hidden Layer Size			
	3 Neuron	8 Neuron	10 Neuron	15 Neuron
Epoch	98	279	443	1000
Computing Time (sec)	1	5	9	32
Gradient	2.62	0.82	0.329	0.178
Effective number of Parameter (used/total)	43.4/67	120/177	147/221	162/331
Regression	0.94886	0.98218	0.98175	0.97669
Mean Performance Error	18.4899	6.7013	6.7026	8.9125
Train Performance Error	13.6415	1.3874	0.2660	0.0515

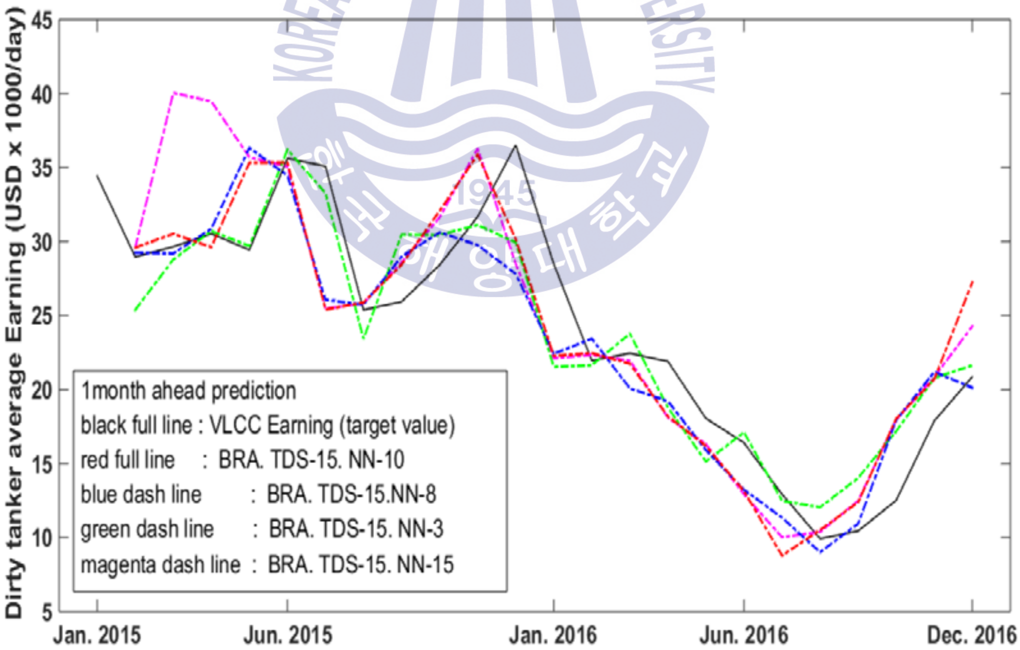

Figure 4.27 Comparison for different hidden layer size

4.2.5 Evaluation on ANN performance results for VLCC according to correlation coefficient between input variables and target variable

In this study, the effect of the correlation between input and target variables on the ANN prediction was investigated with changing the size of each input variables. The ANN model applied for this was the Bayesian regularization algorithm with 10 neurons of the hidden layer, and the test sampling data size was 15%, based on 3-months ahead prediction for VLCC. In addition, ANN prediction was performed for each case by amplifying the magnitudes of nine input variables (Time charter rates, Crude oil prices, World GDP, Global oil production, Active fleets, Bunker prices, Demolition prices, New building prices and second hand ship prices) by 1.1 and 1.3 times, respectively.

The correlation coefficients between Earning as prediction target variable and input variables, and the results of ANN prediction performance are shown in the following **Table 4.20.** The Correlation coefficient between tanker Earning and Time charter rates shows the strongest correlation with r = 0.8190, and followed by Second hand ship prices with r = 0.5825, New building prices with r = 0.3369. Also, in Demolition prices of r = -0.0341, the correlation coefficients between two variables of Earning and Demolition prices is reversed and the correlation strength is also weak.

In the case of the largest Correlation coefficient (r = 0.8190) of Time charter rates, the mean performance error (MSE) is 10.0411 at 1.1 times amplification and 10.8922 at 1.3 times amplification, which are slightly increased from 8.9288 of the section 4.2.1.2 above, but there is no significant difference against the magnitude of the variation of the variables size. In the case of Second ship hand prices (r = 0.5825), the mean performance error (MSE) is 10.1039 at 1.1 times amplification, 22.3375 (MSE) at 1.3 times amplification. The mean performance error (MSE) at 1.3 times amplification is showing significantly greater than

8.9288 of the section 4.2.1.2 above. This trend can be seen in New building prices and Global oil production. Further, the mean performance error (MSE) was significantly increased in both 1.1 and 1.3 times amplification in the case of the other input variables with low correlation strength.

In ANN forecasting, the ANN prediction performance error (MSE) when there is a large correlation between the input and target variable does not change much according to the change of the input variable size, and also in the case of the input variable with a small correlation coefficient, the prediction performance error (MSE) changes according to the change of the input variable size. In other words, it is found that the strength of the correlation between the input variables and the target variable in the ANN affects the accuracy of the ANN prediction performance.

Table 4.20 Comparison on ANN performance for VLCC according to correlation coefficient between input variables and target variable (Earning)

	Correlation coefficient with target variable (r)	Prediction performance error (MSE)	
		1.1 times amplicaton	1.3 times amplicaton
Time charter rate	0.8190	10.0411	10.8922
Crude oil prices	-0.2592	12.9335	19.5328
Global oil production	-0.1966	8.7641	17.2714
Active fleets	-0.4178	13.0031	19.3408
Bunker prices	-0.3616	25.5066	18.8246
Demolition prices	-0.0341	7.3997	5.9599
New building prices	0.3369	8.3411	12.8539
Second hand prices	0.5825	10.1309	22.3375
World GDP	-0.1946	17.2714	14.1946

4.2.6 Comparison of prediction performance error according to ship type

Table 4.21 - Table 4.23 show the prediction results for the three training models by the Bayesian normalization algorithm for dirty tanker Earning of VLCC, SUEZMAX and AFRAMA. And **Figures 4.28** – **Figure 4.30** show graphs of performance error variations.

As shown in **Figure 4.2**, the variation magnitude of the Earning on the ship type is the largest SUEZMAX, followed by AFRAMAX and VLCC. And also in **Table 4.21** – T**able 4.23**, the mean performance error (MSE) for each type of tanker is the largest of SUEZMAX regardless of the neuron size and the test data sampling size, and the AFRAMAX and VLCC are the least.

In the case of VLCC with relatively small market fluctuations, the short term ahead prediction of 1, 3, 6, and 9-months shows satisfactory overall performance with the 8 neurons architecture in the hidden layer. And with the training algorithm with larger size of 10 neurons, better results are obtained in the 12, 15-months ahead predictions.

Also, in the prediction of the SUEZMAX where the market fluctuation is the most severe, the training algorithm with 10 neurons of hidden layer performs better than the 8 neuron algorithm for all forecast periods.

In the prediction of AFRAMAX with a large shipping market variance, the training algorithm with 8 neurons of hidden layer, which is smaller than the number of input variables, has better performance in 1-month and 3-months ahead prediction than the training algorithm of 10 neurons. And in the 6, 9, 12, 15-months ahead prediction, the larger network having 10 neurons has better performance than the smaller network of 8 neurons.

In the test data sampling size of 15%, the average value of the mean performance error of the 8 neurons in the hidden layer for VLCC is the smallest with 8.30, and SUEZMAX is 49.16, which is 5.9 times larger than that of VLCC. And 33.74 for AFRAMAX, which is 4.07 times larger than VLCC. In the case of neuron size 10 also, VLCC has the smallest error value of 7.57, and SUEZMAX is 33.81, which is 4.47 times larger than VLCC, and AFRAMAX is 31.99, which is 4.23 times larger than VLCC.

In the test data sampling size of 20%, VLCC has the smallest value of 12.70, SUEZMAX is 60.53, which is 4.77 times larger than VLCC, and AFRAMAX is 42.32, which is 3.33 times larger than VLCC. When the, the training algorithm with 10 neurons in the hidden layer of the 15% test data sampling model shows generally better results than the training algorithm with 8 neurons.

From these results, it can be seen that in the bigger variance of the time series data , the prediction error increases more, but the actual forecasting experiment applying the ANN forecasting models is not problematic at any cases. When the test data sampling size was increased from 15% to 20% of the total observation points, that is when the training data sampling size decreased from 85% to 80%, the number of iterations increased in all the forecasting horizons and the mean performance errors (MSE) also increased.

Table 4.21　Comparison of ANN performance error for VLCC

	BRA		
	TDS-15		TDS-20
Advanced period	NN-8	NN-10	NN-10
one-month	6.70	6.70	8.79
3-months	8.64	8.92	11.87
6-months	7.81	8.93	15.50
9-months	6.70	7.67	8.56
12-months	9.39	4.94	14.52
15-months	10.57	8.24	16.95
Average	8.30	7.57	12.70

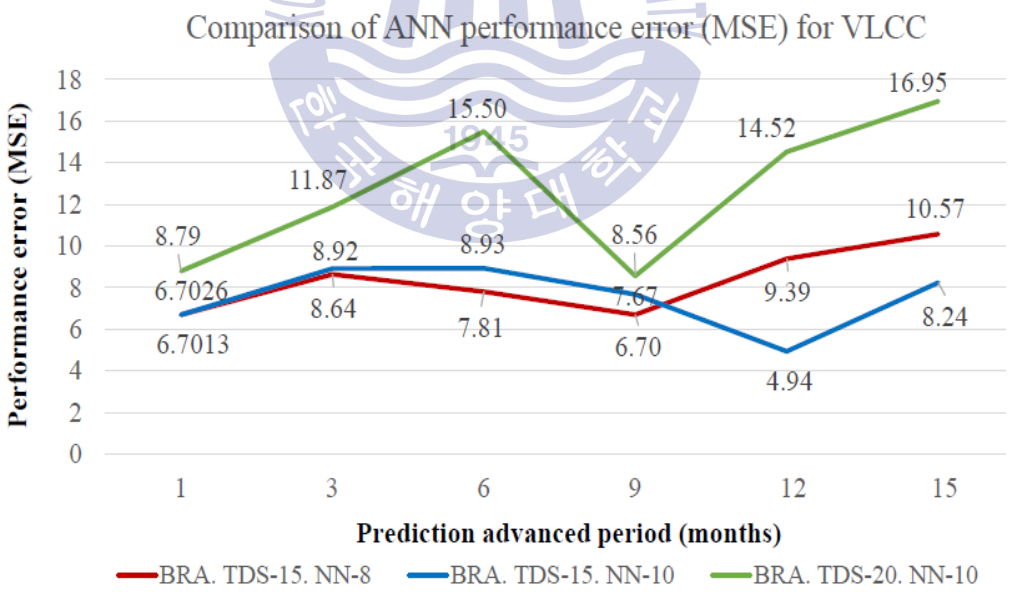

Figure 4.28　Comparison of performance error (MSE) for VLCC

Table 4.22 Comparison of ANN performance error for SUEZMAX

Advanced period	BRA		
	TDS-15		TDS-20
	NN-8	NN-10	NN-10
one-month	34.36	31.11	77.22
3-months	38.97	35.51	51.40
6-months	38.81	34.36	49.25
9-months	72.40	31.11	41.21
12-months	78.10	43.43	57.22
15-months	32.31	27.31	86.86
Average	49.16	33.81	60.53

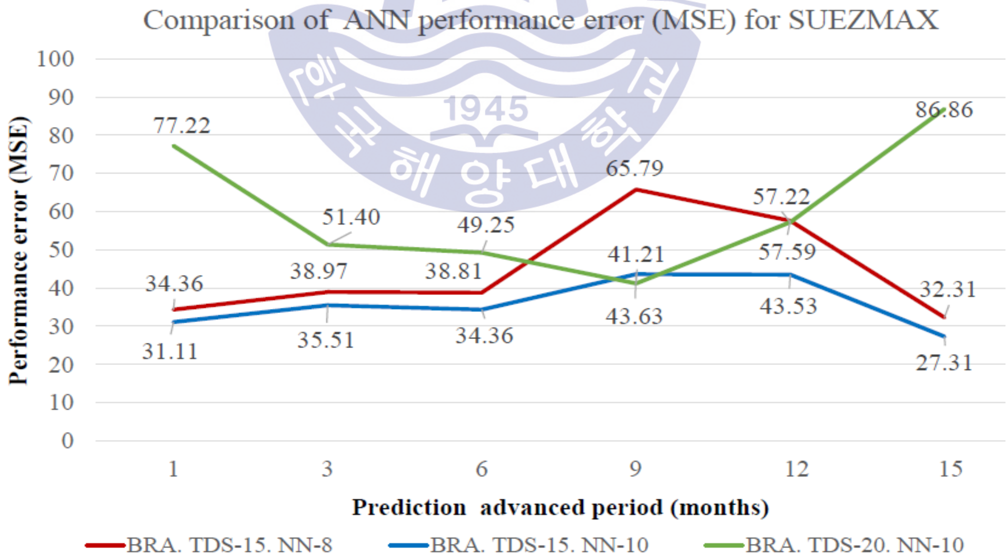

Figure 4.29 Comparison of performance error (MSE) for SUEZMAX

Table 4.23 Comparison of ANN performance error for AFRAMAX

	BRA		
	TDS-15		TDS-20
Advanced period	NN-8	NN-10	NN-10
one-month	31.09	37.83	50.49
3-months	27.79	29.95	35.57
6-months	37.56	29.37	35.57
9-months	33.50	32.66	46.22
12-months	43.43	30.92	50.49
15-months	29.06	31.20	35.57
Average	33.74	31.99	42.32

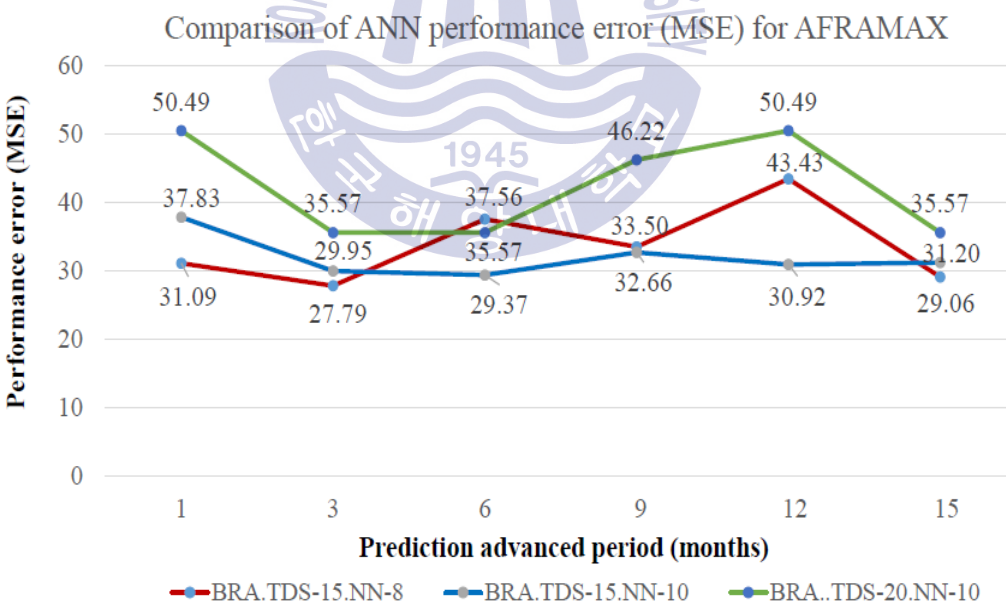

Figure 4.30 Comparison of performance error (MSE) for AFRAMAX

Chapter 5　Conclusions

In this paper, several alternatives to ANN training algorithms have been approached to solve multi-step ahead prediction problems using 204 monthly time series data from 2000 to 2016 for dirty tankers of VLCC, SUEZMAX and AFRAMAX.　The training algorithms of the neural networks used were the Levenberg-Marquardt algorithm and the Bayesian regularization algorithm, and the accuracy of the prediction performance was evaluated by applying the alternatives of the neural networks with changing parameters.　The findings of this study shown that when applying the adjustable parameters such as neuron size of hidden layer and test data sampling size, the Bayesian regularization algorithm has better performance specifications than the Levenberg-Marquardt algorithm in all prediction horizons of supervised prediction with an advanced time of one-month, 3-months, 6-months, 9-months, 12-months and 15-months.　The more detailed findings are as follows.

1.　In the Bayesian regularization algorithm, when the size of the hidden layer neuron increases, iteration and the computing time generally increase, and the gradient value and train performance error (MSE) converge to a smaller value.　However, in case of the considerably increasing or decreasing neuron size, the R-values does not improve and also prediction performance error does not decrease.　The mean performance error (MSE) are minimized when the size of the neurons is similar to the number of the input variables.

2.　In the short term ahead prediction within about 1 year, ANN training architecture with a smaller neuron size of hidden layer than the input variables has a best performance. In the long term ahead prediction about over 1 year, ANN training architecture with a larger neuron size of hidden layer than the input variables has a best performance.

3. Prediction of time series data with large fluctuation shows best forecasting specification in a training algorithm where the size of neurons in the hidden layer is larger than the input variables without relation to the forecasting horizons. Prediction of time series data with large fluctuation, the prediction error of training models increases more, but the actual forecasting experiment applying the ANN forecasting models is not problematic in terms of accuracy and tendency of prediction performance.

4. When there is a large correlation between the input and target variable, the ANN prediction performance error (MSE) does not change much with the change of the input variables size, and in the case of the input variable with a small correlation coefficient, the prediction performance error (MSE) changes according to the change of the input variable size. The strength of the correlation between the input variables and the target variable affects the accuracy of the ANN prediction performance.

5. When increasing the size of the test data set in the Bayesian regularization algorithm, the R value between outputs and targets and the mean performance errors show the worse than the smaller test data set. The reason seems to be that the training data set has been reduced.

This study concludes that in forecasting the dirty tanker markets, ANN can be used as substantial tool to more accurately predict market changes for various type of vessels, regardless of the magnitude of fluctuations. However, in order to improve predictive performance, it is important to design an optimal ANN architecture for the targets to be predicted.

References

[1] The international energy agency (IEA), https://www.iea.org/statistics/kwes/, Accessed October 20, 2017.

[2] Exxonmobil. https://corporate.exxonmobil.com/en/energy/energy-outlook/a-view-to-2040, Accessed October 29, 2017.

[3] Clarksons Research Services, https://sin.clarksons.net/Timeseries, Accessed October 24, 2017.

[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] Worldscale Association,

https://www.worldscale.co.uk/BookPage/2019/PreambleIndex.html?embed=True, Accessed May 31, 2016.

[6] G. Dimitriou, "Where is the Crude Oil Tanker Market Heading in the next ten years", MSc in Maritime Economics and Logistics. Erasmus University Rotterdam. 2015/2016.

[7] S. D. Toslakis, C. Cridland and H. E. Haralambides, "Econometric Modelling of Second-hand Ship Prices", Maritime Economics & Logistics, Vol.5, Issue 4, pp. 347-377, 2003.

[8] A. H. Alizadeh and W. K. Talley, "Vessel and voyage determinants of tanker", Transport Policy, Vol. 18, pp. 665-675, 2011.

[9] A. Jugovic, N. Komadina and A. P. Hadzic, "Factors influencing the formation of freight rates on maritime shipping markets", Scientific Journal of Maritime Research, Vol. 29, pp. 23-29, 2015.

[10] Z. Zannetos, The theory of oil tank ship rates, Cambridge, US. : The MIT press, 1966.

[11] D. Glen, M. Owen and R. van der Meer, “Spot and time charter rates for tankers, 1970-77”, Journal of transport economics and policy, Vol. 15, No. 1, pp. 45-58, 1981.

[12] D. Hawdon, “Tanker freight rates in the short and long run”, Applied Economics, Vol. 10, Issue 3, pp. 203-217, 1978.

[13] M. Beenstock and A. R. Vergottis, “An econometric model of the world market for dry cargo freight and shipping”, Applied Economics, Vol. 21, Issue 3, pp. 339-356, 1989.

[14] G. Box and G. Jenkins, TIME SERIES ANALYSIS forecasting and control, San Francisco, CA: Holden-Day, 1976.

[15] M. G. Kavussanos, “Comparison of volatility in the dry-cargo ship sector”, Journal of Transport Economics and Policy, Vol. 30, No. 1, pp. 67-82, 1996.

[16] 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.

[17] M. D. Kagkarakis, A. G. Merikas and A. Merina, “Modelling and forecasting the demolition market in shipping”, Maritime Policy & Management. Vol. 43, No. 8, pp. 1021-1035, 2016.

[18] 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.

[19] A. W. Veenstra and P. H. Franses, “A co-integration approach to forecasting freight rates in the dry bulk shipping sector”, Transportation Research – A, Vol. 31, No. 6, pp. 447–458, 1997.

[20] M. G. Kavussanos and A. Alizadeh, "The expectations hypothesis of the term structure and risk premia in dry bulk shipping freight markets: An EGARCH-M approach", Journal of Transport Economics and Policy, Vol. 36, No. 2, pp. 267–304, 2002.

[21] R. Adland and K. Cullinane, "A time-varying risk premium in the term structure of bulk shipping freight rates", Journal of Transport Economics and Policy, Vol. 39, No. 2, pp. 191–208, 2005.

[22] 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.

[23] 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.

[24] 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, Issue 1, pp. 72–91, 2013.

[25] G. Zhang, B. E. Patuwo and Y. H. Hu, "Forecasting with artificial neural networks: The state of art", International Journal of Forecasting, Vol. 14, pp. 35-62, 1998.

[26] A. Paplinski, Artificial Neural Networks. Course, Monash University, Australia, 2004. [Online]. Available: https://research.monash.edu/en/publications/ 2004.

[27] S. Fan, T. Ji, W. Gordon and B. Rickard, "Forecasting Baltic Dirty tanker Index by Applying Wavelet Neural Networks", Journal of Transportation Technologies, Vol. 3, pp. 68-87, 2013.

[28] W. Remus and M. O'Connor, Neural networks for time-series forecasting, Kluwer Academic Publishers, 2001.

[29] 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.

[30] K. L. Du and M. N. S. Swamy, Neural Networks and Statistical Learning, Springer London, 2014.

[31] M. T. Hagan, H. B. Demuth, M. H. Beale and O. D. Jesὐs, Neural Network Design, 2nd edition, 2014. [Online]. Avaliable : http://hagan.okstate.edu/NNDesign.pdf.

[32] D. E. Rumelhart, G. E. Hinton and R. J. Williams, “Learning representations by back-propagating errors”, Nature, Vol. 323, pp. 533-536, 1986.

[33] K. L. Priddy and P. E. Keller, Artificial Neural Networks an introduction, Vol. TT68, Bellingham, USA, 2005.

[34] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain”, Psychological Review, Vol. 65, pp. 386-408, 1958.

[35] B. Widrow and M. E. Hoff, Adaptive switching circuits. Technical Report No. 1553-1, Solid-State Electronics Laboratory, Stanford Electronics Laboratories and Stanford University, USA.

[36] B. Widrow and S. D. Stearns, Adaptive Signal Processing, Englewood Cliffs, N.J.07632: Prentice-Hall, 1985.

[37] R. A. Jacobs, Increased rates of convergence through learning rate adaption, Neural Networks, Vol. 1, Issue 4, University of Massachusetts, USA, pp. 295-308, 1988.

[38] D. F. Shanno, Recent advances in numerical techniques for large scale optimization, Neural networks for control, Cambridge MA: MIT Press, 1990.

[39] L. E. Scales, Introduction to Non-Linear Optimization, New York. Springer-Verlag, 1985.

[40] C. Wang, S. S. Vencatesh and J. S. Judd, “Optimal stopping and Effective Machine Complexity in Learning”, Advances in Neural Information Processing Systems, Vol. 6, pp. 303-310, 1994.

[41] W. S. Sarle, “Stopped training and other remedies for over fitting”, The 27th Symposium on Interface, 1995.

[42] A. N. Tikhonov, “On the solution of ill-posed problems and the regularization method”, Dokl. Acad. Nauk SSSR, Vol. 151, No. 3. pp. 501-504, 1963.

[43] D. J. C. MacKay, “Bayesian Interpolation”, Neural Computation, Vol. 4, Issue 3, pp. 415-447, 1992.

[44] Organization of the Petroleum Exporting Countries (OPEC), https://www.opec.org/opec_web/en/publications/3407.htm, Accessed October 20, 2017.

[45] Y. J. Jung, S. H. Kang, and D. H. Doh, “Forecasting Earning of VLCC tankers using artificial neural networks”, Journal of the Korean Society of Marine Engineering, Vol. 42, No. 10, pp. 851-858, 2018.

[46] United Nations Conference on Trade and Development (UNCTAD), https://unctad.org/en/Pages/statistics.aspx, Accessed October 20, 2017.

Acknowledgements

I would like to express my sincere appreciation to a lot of respectful people who helped me to finish my doctoral dissertation through this writes.

First of all, I am especially indebted to Dr. Deog-Hee Doh who encouraged me to draw pictures for my dissertation and presented me the directions of my research. And I would like to thank Dr. Dong-Hyuk Kim and Dr. Sam-Sang You who made great efforts in reviewing and adding valuable advice on my thesis during the busy time. Also I would like to thank Dr. Jeong-Ryul Kim for practical advice on academic activities for industrial development after graduation. In particular, I would like to appreciate to Dr. Ue-Kan Kim, who gave detailed instructions on all aspects of school life and research, I was able to finish my thesis with pleasure under his enduring encouragement.

When I think about the past time, I think I have done my best even though I have traveled a long distance between Daejeon and Pusan for many years. However I think it's thanks to the support and encouragement of my wife and colleagues. I give all the honor of my doctorate to my wife who had to take care of her job and housework for a long time alone. And I would like to express my heartfelt gratitude to everyone who cared for and encouraged me.