Journal of Advanced Marine Engineering and Technology, Vol. 44, No. 2, pp 174~180, 2020
J. Advanced Marine Engineering and Technology (JAMET)
https://doi.org/10.5916/jamet.2020.44.2.174

ISSN 2234-7925 (Print)
ISSN 2234-8352 (Online)
Original Paper

Study on prediction of ship′s power using light GBM and XGBoost

Ji-Yoon Kim[1] • Hun-Seok Lee[2] • Jin-Seok Oh[†]

(Received November 25, 2019 ; Revised February 20, 2020 ; Accepted March 4, 2020)

Abstract: Research on smart ship is becoming active. In smart ships, forecasting power demand is important. Through the power forecast, the power supply can be flexibly improved to maximize the power generation efficiency. Energy reduction can be achieved through increased efficiency. In this paper, we construct a ship power demand prediction model using XGBoost (eXtream Gradient Boost) and LGBM (Light Gradient Boosting Model). The hyperparameters of the boosting algorithm improve the accuracy of the prediction model and prevent overfitting. In addition, the verification functions of XGBoost and LGBM were constructed and verified, and the model suitability was compared. The verification function used Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Scaled Error (MASE). The XGBoost model showed similar performance compared to LGBM, but was slow in computation speed. In terms of operation speed, LGBM is recommended when constructing a power demand prediction model using boosting techniques in smart ships.

Keywords: Boosting algorithm, XGBoost, LGBM, Power demand, Smart ship

1. Introduction

A smart ship refers to a ship in which all the devices and conditions of the ship can be checked on land by using IoT, big data, and artificial intelligence, and also a ship with the capability of diagnosing or predicting device failure, autonomous operation, and eco-friendly operation. Recently, research on smart ships has been actively underway.

Smart ships should be able to predict power and change a supply method through big data. The supply method can be combined by using the number of parallel generators and batteries. We can improve the operating efficiency of generators through the combined method. We can reduce fuel consumption through the improved efficiency, which will affect the economy and environmental protection. Therefore, the accuracy of power prediction is critical.

The power devices used in a ship is limited, and the devices used are set in advance depending on the ship's operation mode. The shipyard analyzes power loads by using this. However, since power loads depend on the operation and situation of a ship, the method adopted by the shipyard is not accurate.

In this paper, we created power demand prediction models for actual measurement data of a ship using eXtreme Gradient Boost (XGBoost) and Light Gradient Boosting Model (LGBM), verified the models using Squared Root (SQRT), Mean Absolute Percentage Error (MAPE), and Mean Absolute Squared Error (MASE).

XGBoost is one of the ensemble algorithms which apply the gradient boosting technique to the tree model that creates the more powerful classifier by sequentially improving the weak classifier. It is very accurate thanks to the use of the boosting technique, and its calculation speed is fast thanks to parallel calculation.

LGBM is an algorithm for improving low efficiency and scalability when the data of the high dimensional variable of the conventional gradient boosting technique is large, and uses Gradient-based One-Side Sampling (GOSS) and a new algorithm called Exclusive Feature Bundling (EFB).

In this paper, we propose an optimal model by using a ship power demand prediction method through XGBoost and LGBM techniques and adjusting hyperparameters for each technique. In addition, we propose a fast prediction model relative to performance by measuring the prediction time of each

† Corresponding Author (ORCID: https://orcid.org/0000-0003-3627-476X): Professor, Division of Marine Engineering, Korea Maritime and Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: ojs@kmou.ac.kr, Tel: 051-410-4283
1 Director, Research Institue, Romantique, E-mail : rlawldbs2918@new.toto-romance.com, Tel : 051-242-8619
2 Senior Engineer, Naval System Team 3 of Naval R&D Center, Hanwha System, E-mail: hs.lee20@hanwha.com, Tel: 054-460-8726

prediction model. We use data measured at an interval of 10 minutes for one year from 2014.11 to 2015.12 to train the prediction models.

2. Ship data and algorithm

2.1 Ship specifications and data

The target ship is a 6,800TEU container, which is driven by one 69MW MAN B&W diesel engine and consists of four 3,000kW generators. The specifications of the target ship are shown in **Table 1**.

Table 1: Specification of target ship

kind of Ship	container
length over all	299m
Extreme Breadth	40m
Depth	13.5m
output of Main Engine	68,520kW
output of Generator Engine	3,000kW
Maximum Speed	25Knot
TEU	6732TEU

We used data measured at an interval of 10 minutes from 2014.11 to 2015.12. The data used is shown in **Table 2**.

Table 2: Acquisition data list

No.	Data	Unit
1	Wind Speed	m/s
2	Wind Angle	degree
3	Water Speed	m/s
4	Main Engine RPM	RPM
5	Ship Speed	knot
6	Ship State	-
7	Total Load	kW

The power consumption of a ship depends on the operation mode and environment of the ship. The operation mode of the ship is divided into voyage, entry and departure, and anchoring, and the equipment used in the ship differs according to each operation mode. For example, a large amount of power is consumed with the use of the bow thruster and winch for entry, and departure. Therefore, in this study, we divide the measurement data into the integer data of 0~2 depending on the operation mode of the ship. In addition, the external environment of the ship affects the main engine and how each device operates. This is because the parallel operation of equipment or cascade control under low loads is utilized as the environment outside the ship changes. In this study, we use the values such as the wind speed, wind angle, and water speed in **Table 2** as the external environmental data that is used as training data. Since the use of the main engine affects the use of the air compressor and auxiliary blower, this can represent the load of the main engine by utilizing the vessel speed and the Revolution Per Minute (RPM) of a main engine. Therefore, we use the operation mode, external environment, and the load of main engine of the ship as the XGBoost and LGBM models.

2.2 XGBoost

XGBoost applies the boosting technique to the decision tree model. The boosting technique is an algorithm that produces robust models with the capability of complex prediction by combining weak models suitable for simple classification. After training the given data through the weak classifier, we can reduce errors by training the errors present in the trained result through another weak classifier.

XGBoost gives high scores to the tree model of high importance by assigning different weights to each model when integrating the models through the boosting technique. The weight of the previous model depends on the current error. The objective function for training consists of the loss function and normalization term between the true and predicted values, and the model is trained by obtaining the weight that minimizes this objective function.

2.3 LGBM

LGBM is an algorithm for improving low efficiency and scalability when the data of the high dimensional variable of the conventional gradient boosting technique is large, and uses GOSS and a new algorithm called EFB. GOSS takes configuration that data entities with large gradients can play an essential role in calculating information acquisition by excluding data with small gradients among data entities. Therefore, GOSS can estimate information acquisition very accurately even with a small amount of data. EFB binds mutually exclusive variables through the greedy algorithm to reduce the number of variables. Therefore, it can effectively reduce the number of variables without significantly compromising the accuracy of split point determination.

3. Hyperparameter

3.1 Hyperparameter

A hyperparameter usually refers to a variable that is directly tuned by the user when training any arbitrary model in machine learning. The optimized hyperparameter will change depending on the given data. Therefore, it can be determined by experience or confirmed through various search methods.

Table 3: Important Hyper parameter of each model

XGBoost	LGBM
colsample_bytree	colsample_bytree
subsample	subsample
gamma	min_split_gain
max_depth	max_depth
n_estimators	n_estimators
-	num_leaves

Table 3 shows the major hyperparameters by model. In this study, we optimize the prediction models by adjusting the key hyperparameters in **Table 3**.

The following is the description of the hyperparameters in **Table 3**.

- colsample_bytree: When training a tree, we do not use all training data, but only some of the data randomly extracted based on the column. This improves performance by reducing overfitting.
- subsample: When training a tree, we do not use all training data, but only some of the data randomly extracted based on the row. This improves performance by reducing overfitting.
- max_depth: We specify the maximum tree depth in the process of forming the tree model. A small value causes underfitting while a large one causes overfitting.
- n_estimators: It refers to the number of trees. Since increasing the value adds more trees to the ensemble, the complexity of the model increases. Therefore, the probability of correcting errors in training goes up. However, the large value means more memory and longer training time.
- gamma: It is a hyperparameter used for XGBoost, which limits the complexity of the tree model. If the value is large, the tree model does not create many leaf nodes easily.
- min_split_gain: It is a hyperparameter used for LGBM, and plays the same role as the gamma of XGBoost.
- num_leaves: It is a hyperparameter used in LGBM. It refers to the number of leaves of the entire tree, and is the main parameter that limits the complexity of the tree model. The large value improves accuracy but can lead to overfitting.

3.2 The optimization method of hyperparameters

Hyperparameters are used to enhance prediction models, and is a variable that must be set directly by the user. The hyperparameters are not theoretically determined but must be set empirically. Finding optimal hyperparameter is essential for building high-performance prediction models.

There are three widely-known optimization methods of hyperparameters. That is, grid search, random search, and Bayesian optimization are typical methods. In this paper, we optimize hyperparameters using the grid search.

The grid search is a method of dividing hyperparameters into grids and then checking the model performance for all grid points. When searching hyperparameters in a grid form, a relatively even and global search is possible. However, if the gap is too tight or the number of hyperparameters increases, there is a drawback that the search time increases.

3.3 Optimization of hyperparameters of the XGBoost model

In this study, we use the grid search for hyperparameter optimization. To optimize hyperparameters, it is important to adjust the variables in **Table 3** in order. We adjust variables in the order of colsample_bytree, subsample, gamma, max_depth, n_estimators. After adjusting the variables, we select the variable with the smallest Root Mean Square Error (RMSE).

3.3.1 Validation function

In this paper, we use the k-fold cross-validation provided by the scikit-learn library to verify the hyperparameter values. Cross-validation randomly divides the training set into k subnets called a fold, and trains and evaluates the decision tree model by k times. Each time we select a different fold and used it for evaluation, and use the remaining k-1 folds for training. After validation, k evaluation scores are derived. In this paper, we configure the k evaluation scores to derive RMSE.

3.3.2 colsample_bytree, subsample

Of hyperparameters, colsample_bytree and subsample are used to prevent overfitting. They are initially set to 1. It the value is large, RMSE becomes small but can be overfitting. For the ensemble effect, after setting the two hyperparameters to 0.7 and verifying RMSE, we found that RMSE was 0.1102118.

3.3.3 gamma adjustment

gamma limits the complexity of the tree model, and is initially set to 0. While changing the gamma value from 0.4~0.8 by 0.1 with the colsample_bytree and subsample values fixed to 0.7, we verify RMSE using the validation function. **Figure 1** shows the validation results according to the change of the gamma value. The results show that RMSE was the lowest with 0.10986 at the gamma value of 0.7.

3.3.4 max_depth

max_depth specifies the maximum tree depth, and is initially set to -1. While changing the max_depth value from 5~9 by 1 with the colsample_bytree, subsample and gamma values fixed to 0.7, we verify RMSE using the validation function. **Figure 2** shows the validation results according to the change of the max_depth value. The results show that RMSE was the lowest with 0.1095133 at the max_depth value of 8.

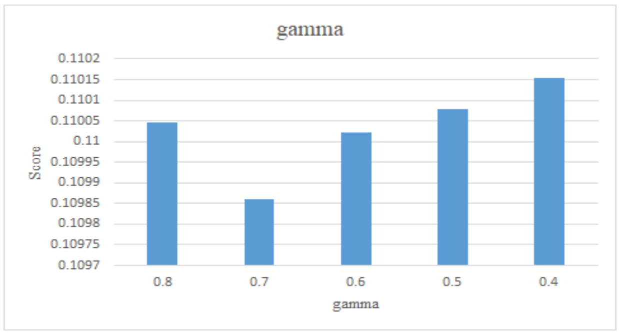

Figure 1: score of gamma in XGBoost

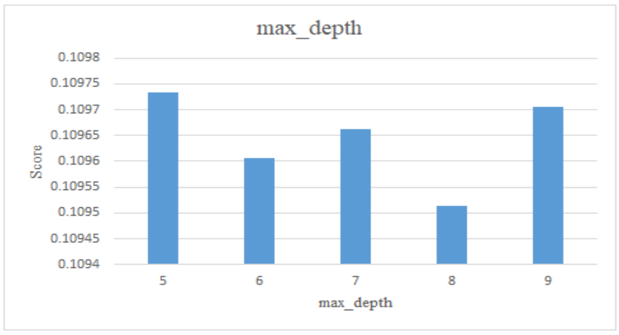

Figure 2: score of max_depth in XGBoost

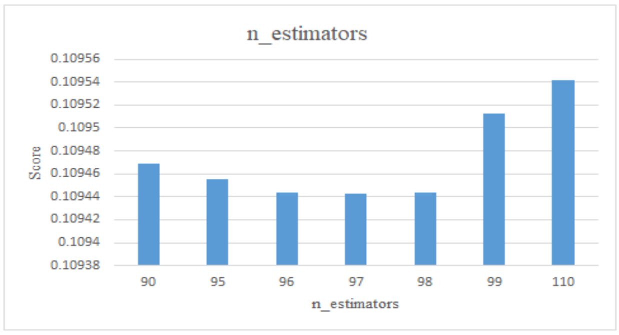

Figure 3: score of n_estimators in XGBoost

3.3.5 n_estimators

n_estimators refers to the number of trees, and is initially set to 100. While changing the n_estimators value from 95~99 by 1 with the colsample_bytree, subsample and gamma values fixed to 0.7 and the max_depth value to 8, we verify RMSE using the validation function. **Figure 3** shows the validation results according to the change of the n_estimators value. The results show that RMSE was the lowest with 0.1094432 at the n_estimators value of 97.

3.4 Optimization of hyperparameters of the LGBM model

LGBM also uses the grid search method just like the optimization of XGBoost. colsample_bytree, subsample, max_depth, and n_estimators are the same as XGBoost but min_split_gain and num_leaves are different. Thus, the grid search order of LGBM is different from that of XGBoost. The variable optimization order of LGBM is colsample_bytree, subsample, min_split_gain, max_depth, n_estimators, and num_leaves. Just like XGBoost, the validation function used for optimizing variables uses the k-fold cross-validation provided by the scikit-learn library.

3.4.1 colsample_bytree, subsample

Of hyperparameters, colsample_bytree and subsample are used to prevent overfitting. They are initially set to 1. For the ensemble effect, we use 0.7 that was used in XGBoost. The results of RMSE using the verification function was 0.1118904 after setting them to 0.7.

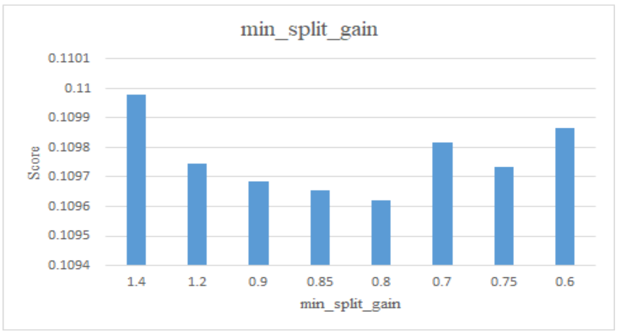

Figure 4: score of min_split_gain in LGBM

3.4.2 min_split_gain

min_split_gain plays the same role as the gamma of XGBoost, and limits the complexity of the tree model. It is initially set to 0. While changing the min_split_gain value with the colsample_bytree and subsample values fixed to 0.7, we verify RMSE. **Figure 4** shows the results of the validation function according to the change of the min_split_gain value. The results show that RMSE was the lowest with 0.1096191 at the min_split_gain value of 0.8.

3.4.3 num_leaves

num_leaves adjusts the number of leaves in the entire tree, and limits the complexity of the tree model. It is initially set to 31. While changing the num_leaves value from 7~11 by 1 with the colsample_bytree and subsample values fixed to 0.7 and the min_split_gain value to 0.8, we verify RMSE. **Figure 5** shows the validation function results according to the change of the num_leaves value. The results show that RMSE was the lowest with 0.1095569 at the num_leaves value of 9.

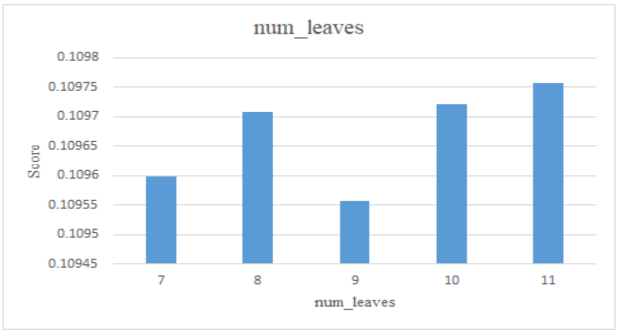

Figure 5: score of num_leaves in LGBM

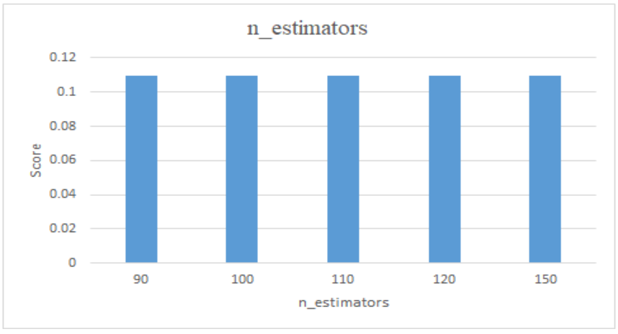

Figure 6: score of n_estimators in LGBM

3.4.4 n_estimators

n_estimators refers to the number of trees, and is initially set to 100. While changing the n_estimators value with the colsample_bytree, subsample values fixed to 0.7, the min_split _gain value to 0.8, and the num_leaves value to 9, we verify RMSE using the validation function. **Figure 6** shows the validation function results according to the change of the n_estimators value. The results show that RMSE was the constant with 0.1095569. Therefore, we use the initial setting value of 100.

3.4.5 max_depth

max_depth specifies the maximum tree depth, and is initially set to -1. While changing the max_depth value from 1~5 by 1 with the colsample_bytree and subsample values fixed to 0.7, the min_split _gain to 0.8, the num_leaves to 9, and the n_estimators to 100, we verify RMSE using the validation function. **Figure 7** shows the validation function results according to the change of the max_depth value. The results show that RMSE was the lowest with 0.1093323 at the max_depth value of 4.

4. The analysis and evaluation of ship power demand prediction

4.1 Validation method

To validate the models with optimized hyperparameters, we use RMSE that represents the difference between the predicted and actual values of the models, MAPE that can solve the drawbacks of size-dependent errors, and MASE that can represent the difference between the predicted and actual values as the mean variation.

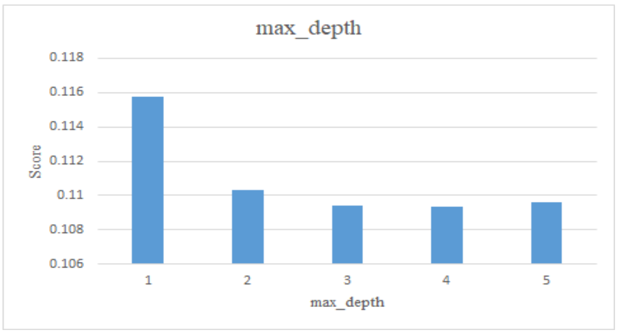

Figure 7: score of max_depth in LGBM

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{2}$$

$$MASE = \frac{\sum_{j=1}^{J}|e_j|}{\frac{J}{T-1}\sum_{i=2}^{T}|Y_t - Y_{t-1}|} \tag{3}$$

In **Equation (1)**, $\hat{y}_i$ refers to a predicted value, y_i to an actual value, and n to the number of data.

In **Equation (2)**, F_t refers to a predicted value, A_t to an actual value, and n to the number of data.

In **Equation (3)**, e_j refers to the j_{th} error value, J to the number of predictions, Y_t to an actual value, and T to the number of data.

Furthermore, in order to increase the reliability of the model, we configure it to verify the accuracy of the model according to the amount of the training data by changing the amount of the training data to 20%, 40%, 60%, 80%.

4.2 Model evaluation

4.2.1 LGBM

We can confirm that it shows high performance with RMSE showing 0.1 on average, MASE 0.07, and MAPE 1.

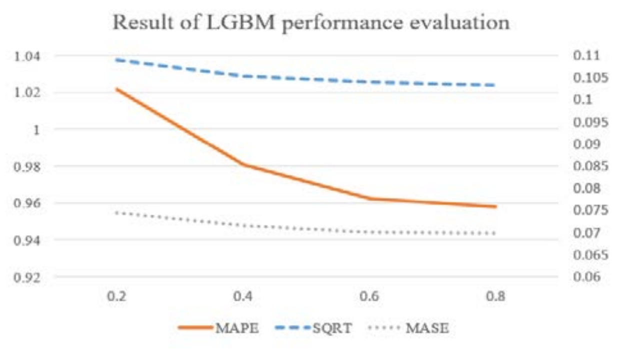

Figure 8: Result of LGBM performance evaluation

4.2.2 XGBoost

We can confirm that it shows high performance with RMSE showing 0.1 on average, MASE 0.06, and MAPE 0.9.

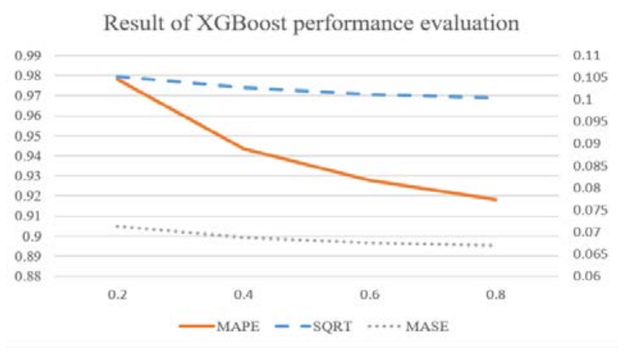

Figure 9: Result of XGBoost performance evaluation

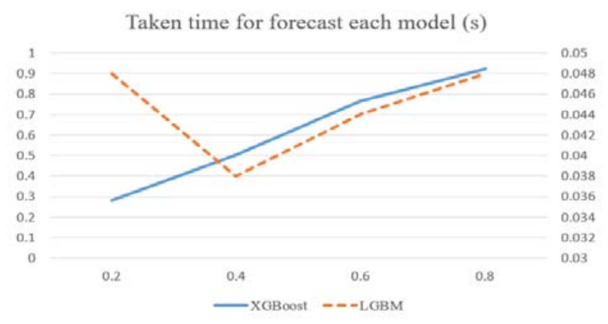

Figure 10: Taken time for forecast each model

4.2.3 Prediction time by model

We compare the prediction time of LGBM and XGBoost by training data. The number of training data consists of 20%, 40%, 60%, 80% of the total data. **Figure 10** shows the prediction time by training data.

In **Figure 10**, the left side represents the values of XGBoost and the right side does the values of LGBM. XGBoost took 0.6183 seconds on average and LGBM did 0.0444 seconds on average. This means approximately 14 times the difference in the calculation speed between them.

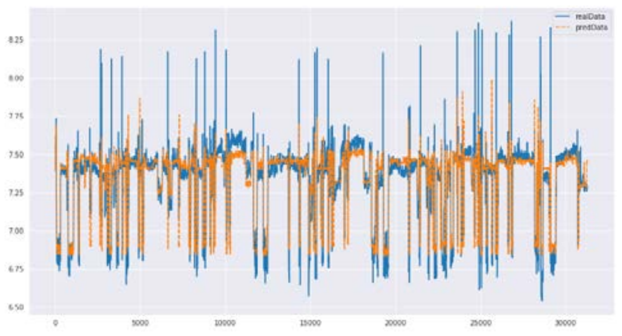

Figure 11: Comparison of data between real and predicted values (XGBoost)

4.3 Comparison of prediction models

Figure 11 and **12** show graphs comparing the training results using XGBoost and LGBM with the actual power data, respectively. As shown in the graphs and the evaluation of the models described in Section 4.2, the errors of the two models are not significantly different.

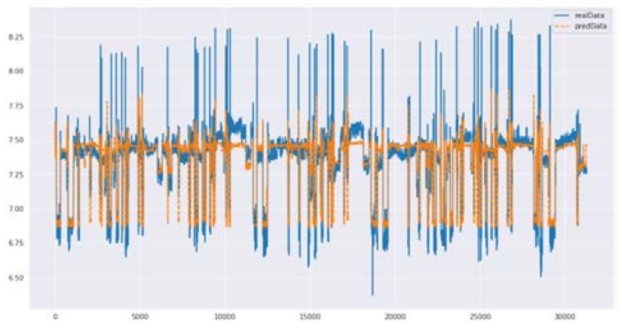

Figure 12: Comparison of data between real and predicted values (LGBM)

5. Conclusion

In this paper, we propose an actual ship power prediction method using XGBoost and LGBM models. We optimized the hyperparameters of the XGBoost and LGBM models using the grid search method by training them using the measurement values of the operation mode and external environment of a ship as input variables. We validated the accuracy of the models according to the optimization of hyperparameters, and also compared the calculation speed according to the number of data. In terms of accuracy, there is not much difference between XGBoost and LGBM. However, LGBM is 14 times faster than XGBoost in terms of computation speed. Therefore, we suggest using LGBM as a prediction model.

Acknowledgement

This study was conducted with the support of the Ministry of Education of Korea and National Research Foundation of Korea (NRF-2018R1D1A1B07049361) in 2018, and Korea Institute of Marine Science and Technology Promotion (Development of Korean style all-weather rescue boats for isolators in the coastal areas) funded by Korea Coast Guard in 2020.

Author Contributions

Conceptualization, J. Y. Kim; methodology, J. Y. Kim and H. S. Lee; Software, J. Y. Kim; Formal Analysis, H. S. Lee and J. Y. Kim; Investigation, H. S. Lee; Resources, J. S. Oh; Data curation J. Y. Kim and H. S. Lee; Writing-Original Draft Preparation, J. Y. Kim; Writing-Review & Editing, H. S. Lee.

References

[1] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boo sting System,” Proceedings of the 22nd ACM SIGKDD Int ernational Conference on Knowledge Discovery and Data Mining, pp 785-794, 2016. Available: https://doi.org/10.11 45/2939672.2939785.

[2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. -Y. Liu, “LightGBM: A highly efficient gradient boosting decision tree” NIPS Proceeding of 31st Conference, pp. 3146-3154, 2017

[3] Y. G. Kim, S. R. Lee, M. A. Jeong, B. M. Kim, and S. W. Min, “Efficient data transmission scheme with data fusion inside a smart vessel,” The Journal of Korean Institute of C ommunications and Information Sciences, vol. 39C, no. 11, pp. 1146-1150, 2014 (in Korean). Available: https://doi.or g/10.7840/kics.2014.39C.11.1146

[4] J. H. Park, J. Y. Son, G. I. Lee, "Trends in smart monitoring and failure handling systems for ships and offshore plants," The Korean Institute of Information Scientists and Engineers (in Korean), vol. 31, no. 1, pp. 46-54, 2013

[5] S. M. Jeon, Y. O. Cho, B. M. Kim, C. H. Kim, J. Y. Park, J. Y. Chu, and S. R. Lee, "Intelligent energy management system for smart ship", Conference Proceeding (summer) of Korea Institute of Communication Sciences, pp. 209-210, 2013 (in Korean).

[6] S. H. Jeong, J. H. Shim, and K. S. Choi, "The common platform technology of smart maritime autonomous surface ships," Proceedings of KIIT Conference, pp. 442-445, 2018 (in Korean).