Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 21(1): 82-89, Mar. 2023 **Regular paper**

Performance Evaluation of a Feature-Importance-based Feature Selection Method for Time Series Prediction

Hyun Ahn[1]*, ***Member***, ***KIICE***

[1]Division of Software Convergence, Hanshin University, 18101, Republic of Korea

Abstract

Various machine-learning models may yield high predictive power for massive time series for time series prediction. However, these models are prone to instability in terms of computational cost because of the high dimensionality of the feature space and nonoptimized hyperparameter settings. Considering the potential risk that model training with a high-dimensional feature set can be time-consuming, we evaluate a feature-importance-based feature selection method to derive a tradeoff between predictive power and computational cost for time series prediction. We used two machine learning techniques for performance evaluation to generate prediction models from a retail sales dataset. First, we ranked the features using impurity- and Local Interpretable Model-agnostic Explanations (LIME) -based feature importance measures in the prediction models. Then, the recursive feature elimination method was applied to eliminate unimportant features sequentially. Consequently, we obtained a subset of features that could lead to reduced model training time while preserving acceptable model performance.

Index Terms: Feature importance, Feature selection, LIME, Time series prediction

I. INTRODUCTION

Over the past decade, we have witnessed a data explosion in terms of scale and variety, and advances in computing technology have taken such phenomena into account. In this regard, various prediction models for time series prediction that yield a high predictive power by learning patterns of vast and diverse time series have been proposed [1,2].

However, as these data-intensive techniques (e.g., machine learning and deep learning models) generally require massive amounts of computation, the challenge of determining a tradeoff between performance and time cost cannot be understated. For instance, in several state-of-the-art deep learning models, as the amount of data and feature space size increase, the learning process tends to cause nonlinearly increasing amounts of computation.

Hence, in this study, we experimentally validated a feature selection method that guarantees acceptable model performance based on feature importance and significantly reduces the training time. To this end, prediction models were created using a retail sales dataset [3], and features were ranked by applying two types of feature importance. Subsequently, using the recursive feature elimination (RFE) process, the model accuracy and training time for each feature subset were measured while sequentially removing features with lower ranks (i.e., less important features).

The remainder of this paper is organized as follows. Section II summarizes previous studies on feature selection methods based on feature importance for time series prediction. Section III describes the retail sales dataset used in the experiment. Section IV presents the feature selection method proposed herein. Section V provides the experimental results regarding the tradeoff between model performance and training time using feature importance. Finally, the conclusions and future directions are presented in Section VI.

Received 18 October 2022, **Revised** 2 November 2022, **Accepted** 2 November 2022
***Corresponding Author** Hyun Ahn (E-mail: hyunahn@hs.ac.kr)
Division of Software Convergence, Hanshin University, Osan 18101, Republic of Korea

Open Access **https://doi.org/10.56977/jicce.2023.21.1.82** print ISSN: 2234-8255 online ISSN: 2234-8883

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

Copyright ⓒ The Korea Institute of Information and Communication Engineering

II. RELATED WORKS

Sales prediction is one of the most critical issues in building retail businesses. Specifically, store operations can be optimized through short-term sales forecasting (e.g., day-ahead prediction). It is also feasible to forecast future sales trends by analyzing long-term time series data. As sales prediction is a time-series prediction problem, in several cases, raw data are given as multivariate rather than univariate. Therefore, computational approaches, such as machine learning techniques that can efficiently deal with it, are being actively applied [4,5].

For the Rossmann store sales dataset [3], a retail sales dataset provided by Kaggle [6], Kohli et al. [4] performed sales predictions for each drug store using linear regression and k-nearest neighbor (k-NN) regression models. The mean absolute percentage error (MAPE) was 22.1 and 31.4%, respectively. The experimental results suggest that the predictive power of the linear regression model for nonlinear problems is insufficient. The k-NN algorithm is based on feature similarity; however, it also shows limited performance because the importance of each feature is not considered sufficiently.

Weng et al. [5] performed sales prediction on the Rossmann dataset by combining long short-term memory (LSTM [1]), which is a deep learning technique appropriate for nonlinear problems, and a light gradient boosting machine (LightGBM [7]), which has the capability to avoid overfitting. However, as the feature selection technique based on feature importance was not applied in this study, the high computational cost of the LSTM model, which processes sequences of multiple time steps as inputs, is inevitable. Therefore, concerning retail sales prediction, this study focuses on feature selection methods to explore the optimal solution for the tradeoff between the computational cost and model predictive power.

Feature selection strategies are broadly classified into wrapper, filter, and embedded methods [8]. Feature importance can be used as a criterion for scoring and ranking features and is mainly used as the basis for the operation of filter methods. For some machine learning models, the feature importance of a model is naturally quantified during the model training.

In [9,10], the Gini impurity, that indicates the effectiveness of splitting nodes, was employed as the feature importance for feature selection. Furthermore, in [10,11], the authors applied an recursive feature elimination (RFE) method to the features ranked by feature importance to obtain an optimal feature subset.

In [12], a hybrid RFE method combining different feature importance measures for support vector machine, random forest, and gradient boosting machine models was proposed to ensure the robustness of feature selection. Furthermore, to go beyond basic feature importance, the dynamic feature importance (DFI) index was proposed to readjust the measured feature importance according to the redundancy between features [13].

Taken together, a part of the machine learning models (particularly tree-based) intrinsically provides the measured feature importance, which can make filter-based feature selection easier. On the other hand, to obtain feature importance from deep learning models with difficulty of interpretation, it is necessary to apply other explanation means, such as explainable artificial intelligence (XAI) methods.

In [14], the authors adopted local-interpretable model explanations (LIME), an XAI method, to measure feature importance for identified clusters. Similarly, in [15], three feature importance criteria (mean decrease accuracy, Shapley value, and LIME) were compared in terms of the stability of feature selection.

In this study, we evaluated the effectiveness of a feature selection method that combines impurity- and LIME-based feature importance measures, considering the computational cost and model predictive power.

III. EXPLORATORY DATA ANALYSIS

This study uses the Rossmann store sales dataset [3] provided by Kaggle [6]. The dataset contains daily sales data for 1,115 drug stores in Europe. Table 1 represents the description of the dataset.

From the perspective of data quality, it is desirable to handle abnormal data using an appropriate method. We applied an interquartile range (IQR)-based outlier detection method to eliminate extreme outliers. The IQR is defined as $Q3 - Q1$, which represents the difference between the third quartile ($Q3$) and the first quartile ($Q1$). The upper fence (f_3) and lower fence (f_1) that distinguish extreme outliers from normal data are defined as follows:

$$f_3 = Q3 + 3 \cdot IQR\,, \tag{1}$$

$$f_1 = Q1 - 3 \cdot IQR\,. \tag{2}$$

After removing the outliers, the remaining data are rescaled through the min-max normalization to speed up and stabilize the training of the models.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

where x is the single value for a specific variable, x_{max} is the maximum value, and x_{min} is the minimum value. The normalized value x_{norm} is calculated according to the above equation.

The target variable we aim to predict is the daily retail sales, and the number of customers is an important variable

Table 1. Variables in the Rossmann dataset

Data source	Variable	Description
Store information	Store	A unique identifier for each store
	Store Type	Differentiates among 4 different store models: a, b, c, d
	Assortment	Describes an assortment level: a = basic, b = extra, c = extended
	Competition Distance	Distance in meters to the nearest competitor store
	Competition OpenSince	The approximated year and month of the time the nearest competitor was opened
	Promo2	A continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
	Promo2 Since	The year and calendar week when the store started participating in Promo2
	Promo Interval	The consecutive intervals Promo2 is started, naming the months the promotion is started anew. e.g., "Feb,May,Aug,Nov" indicates that each round starts in February, May, August, November of any given year for that store
Store sales data	Store	
	Day Of Week	The day of the week
	Data	A date formatted as YYYY-MM-DD
	Sales	The turnover for any given day (target variable)
	Customers	The number of customers on a given day
	Open	An indicator for whether the store was open: 0 = closed, 1 = open
	Promo	Indicates whether a store is running a promotion on that day
	State Holiday	Indicates a state holiday: a = public holiday, b = Easter holiday, c = Christmas, 0 = none
	School Holiday	Indicates a school holiday: 1 = school holiday, 0 = none

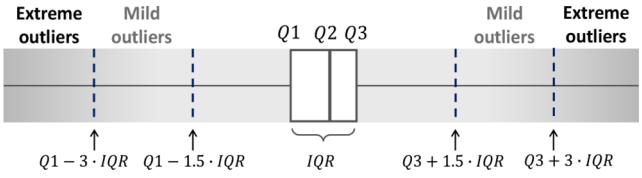

Fig. 1. IQR-based outlier detection.

required for the sales prediction. Table 2 presents the descriptive statistics for these two variables.

Table 2. Descriptive statistics of the sales and number of customers

Variable	Mean	Std. Dev.	Min	Max
Sales	6955.51	3104.21	0	41551
Customers	762.73	401.23	0	7388

Both the daily retail sales and the number of customers have weekly and yearly seasonal components, as shown in Fig. 2. These variables tended to peak at the end of each year.

Promo2, which indicates whether a store is participating in the promotion, is also an important variable to explore the significant difference from the sales data. As shown in Fig. 3, the group of stores that conduct promotions is more likely to achieve high sales.

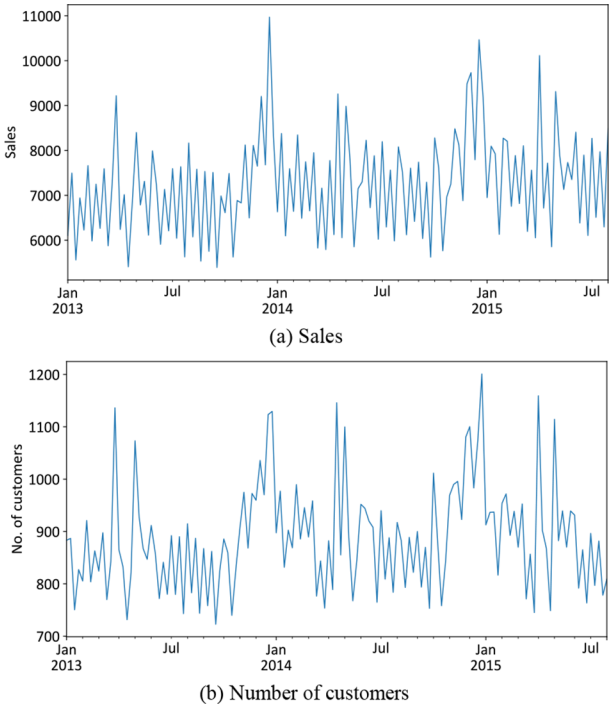

Fig. 2. Time series of two target variables. (a) Retail sales (b) Number of customers.

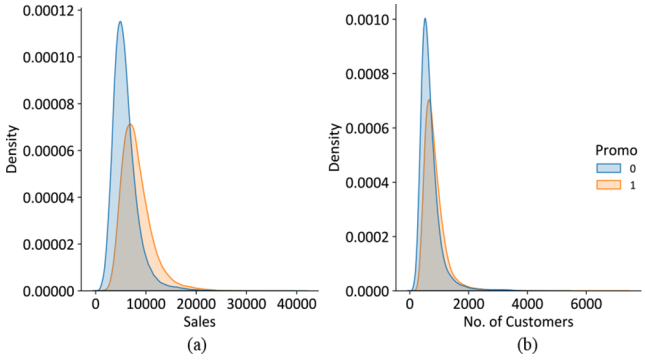

Fig. 3. Distributions by Promo2.

IV. FEATURE SELECTION METHOD

This section describes the primary concepts of the feature selection method investigated in this study. First, two types

of feature importance are briefly introduced: impurity-based and LIME-based feature importance. Subsequently, an RFE method with a threshold is described as the feature selection method.

A. Measuring Feature Importance

1) Impurity-based Method

For tree-structured models, impurity is an indicator used to evaluate the quality of splits on the nodes. In the case of random forest, the impurity-based feature importance is measured according to the following: Given an element s of a set of nodes S in a tree t, $i(s)$ and $w(s)$ represent the impurity and fraction of a number of samples of s, respectively. For regression tasks, the impurity of a node was measured using a loss function. In this study, we applied the mean squared error (MSE) as the metric for $i(s)$.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_l)^2 , \tag{4}$$

where n is the number of samples, y_i is the ith observation, and $\hat{y}_i$ is the predicted value of y_i.

Assuming that t is binary, s has two children: s_l and s_r. The impurity reduction at s, denoted by $\Delta i(s)$, measures the impurity differences between s and its children, s_l and s_r. Therefore, a split at s is considered effective if the impurity reduction occurs to a significant extent.

$$\Delta i(s) = w(s)i(s) - w(s_l)i(s_l) - w(s_r)i(s_r) \tag{5}$$

Given a particular feature θ, $\mathcal{S}^\theta$ is defined as a subset of nodes that split the data on θ. Accordingly, the feature importance of θ for an individual tree t, denoted by $I(\theta)$, is measured using the following equation. Further, the final feature importance for a random forest is calculated by averaging the $I(\theta)$ values of the individual trees.

$$I(\theta) = \frac{\sum_{s \in \mathcal{S}^\theta}\Delta i(s)}{\sum_{s \in \mathcal{S}}\Delta i(s)} \tag{6}$$

2) Local Interpretable Model-agnostic Explanations

LIME [16] is an XAI technique used to understand and explain a complicated model for the instance level using surrogate models. As LIME focuses on local explanations, an explanation model that approximates the original model linearly is created, centered on a specific instance that needs to be explained. Explanations provided by an explanation model are referred to as interpretable visual representations, which vary depending on the type of data (text, image, and tabular data). Therefore, LIME users, including nonexperts, can easily understand the resultant explanations regardless of the actual features used by the model. LIME attempts to derive an explanation model that replicates the behaviors of the original model locally while having low model complexity.

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) , \tag{7}$$

where f is the original model and g is an explanation model as an element of a set of interpretable models G. x is an instance that needs to be explained. π_x measures the proximity between a perturbed instance z and x.

The term $\mathcal{L}(f, g, \pi_x)$ represents a locality-aware loss. It accumulates squared errors between the output of f for z, which is used as a label, and the output of g for z', which is a perturbed instance of reduced dimensionality. This term evaluates the local fidelity of g with respect to f.

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in Z}\pi_x(z)(f(z) - g(z'))^2 \tag{8}$$

$\Omega(g)$ represents the model complexity and is measured in different ways depending on the type of g. For example, if g is a tree-structured model (e.g., decision tree), $\Omega(g)$ represents the depth of the tree, whereas for a linear model, it corresponds to the number of nonzero coefficients of the model.

As our objective model is a prediction model for retail sales, the feature importance measured by LIME is a regression coefficient in a ridge regression model, which is a default explanation model for regression tasks.

B. Recursive Feature Elimination

RFE is the simplest form of the backward feature selection method. This method iteratively removes features ranked by importance one at a time. Let $\rho(\theta_t)$ be a ranking of θ by t, where $t \in \mathcal{T}$ is the type of feature importance. Then, the ranking score of features and the marginal sum of the ranking by the feature importance of individual features are obtained.

$$S(\theta) = \sum_{t=1}^{k}\rho(\theta_t) \tag{9}$$

As our method aims to determine a tradeoff between the predictive power and computational cost, we introduce the falling threshold β, limiting the lower prediction accuracy bound in the RFE method. For example, assuming that the predictive power of the baseline model $\phi(\mathcal{F})$ using the complete set of features $\forall\theta \in \mathcal{F}$ is 10 of mean absolute error (MAE), if β is 0.05 (= 5%), the lower predictive power bound is calculated as 10.5 according to the following equation:

$$\phi_\beta(\mathcal{F}) = \phi(\mathcal{F}) - (\phi(\mathcal{F})\beta) . \tag{10}$$

For each iteration, RFE selects the least important feature θ^r from the current feature set $\mathcal{F}^s$. In other words, the model was trained on the feature set, except for the feature with the maximum of $S(\theta)$. If the predictive power was better than or equal to beta, the feature was removed.

$$\theta^r \leftarrow \underset{\theta \in \mathcal{F}^s}{\operatorname{argmax}} S(\theta) \tag{11}$$

Algorithm 1: Recursive Feature Elimination with β

Input $\mathcal{F}$: Set of features
$\mathcal{T}$: Set of classes of feature importance

begin
  for $\forall\theta \in \mathcal{F}$ **do**
    $S(\theta) \leftarrow \sum_{t=1}^{|\mathcal{T}|} \rho(\theta_t)$;
  end
  $\phi_\beta(\mathcal{F}) = \phi(\mathcal{F}) - (\phi(\mathcal{F})\beta)$;
  $\mathcal{F}^s \leftarrow \mathcal{F}$;
  $\theta^r \leftarrow \arg\max_{\theta \in \mathcal{F}^s} S(\theta)$;
  while $\phi(\mathcal{F}^s - \{\theta^r\}) \geq \phi_\beta(\mathcal{F})$ **do**
    $\mathcal{F}^s \leftarrow \mathcal{F}^s - \{\theta^r\}$;
    $\theta^r \leftarrow \arg\max_{\theta \in \mathcal{F}^s} S(\theta)$;
  end
  return $\mathcal{F}^s$;
end

V. EXPERIMENTAL RESULTS

To validate the feature selection method, we conducted an experiment involving ranking the features by measuring their importance and analyzing performance for each feature subset arranged by the RFE method with β. The specifications of the equipment and software libraries used in the experiments are listed in Table 3.

Table 3. Experimental settings

Attribute	Description
GPU	NVIDIA GeForce 2080 RTX Super (8GB)
CPU	Intel Core i9-9900
RAM	32GB
OS	Windows 10 Pro 64bit
Python Libraries	numpy (numerical operations) pandas (data I/O and manipulation) scikit-learn (feature engineering, model) lightgbm (model) bayesian-optimization (hyperparameter optimization) lime (feature importance) matplotlib (visualization)

A. Prediction Models

Various methods and models can be considered for retail sales prediction. As the feature selection method in this study combines impurity- and LIME-based feature importance, we use two machine learning models to which both of these feature importance methods are applicable: random forest regression (RFR) and light gradient boosting machine (LightGBM).

1) Random Forest Regressor

Random forest [17] is an ensemble machine learning model based on the bagging technique and has been widely used for classification and regression tasks. In this experiment, RFR was used, and its hyperparameters were configured as shown in Table 4.

Table 4. Hyperparameter settings for RFR

Hyperparameter	Default value	Search range	Description
n_estimators	100	[50, 1000]	Number of trees
max_depth	50	[30, 70]	Maximum tree depth
min_samples_split	2	[2, 100]	Minimum number of samples required to split
max_samples	1.0	[0.1, 1.0]	Fraction of the original dataset is given to any individual tree

2) Light Gradient Boosting Machine

LightGBM [7] is a computationally efficient gradient boosting algorithm that prevents overfitting. The hyperparameters of LightGBM were set as listed in Table 5.

Table 5. Hyperparameter settings for LightGBM

Hyperparameter	Default value	Search range	Description
n_estimators	200	[50, 1000]	Number of trees
max_depth	50	[30, 70]	Maximum tree depth
num_leaves	100	[2, 512]	Maximum tree leaves
subsample	1.0	[0.1, 1.0]	Subsample ratio of the training instance

B. Ranking the Features

The feature rankings for the two models were derived based on measurements of feature importance. Table 6 and 7 present the partial results (top-5 features). In the case of RFR, Promo was evaluated as the most important feature. In addition, DayOfWeek_1 (i.e., Monday) and Month_12 (i.e., December), the derived features for modeling weekly and annual cycles, were evaluated as high-ranking features.

Table 6. Partial results of measuring feature importance and overall rankings for RFR

Feature	Importance (impurity)	Importance (lime)	Ranking
Promo	0.1367 (2)	829.5015 (1)	1
DayOfWeek_1	0.0335 (5)	225.5958 (2)	2
Month_12	0.0223 (7)	109.6942 (4)	3
CompetitionOpenElapsedDays	0.1306 (3)	59.6449 (9)	4
Promo2ElapsedDays	0.0416 (4)	47.9985 (11)	5

However, the distance from the competing drug store (i.e., CompetitionDistance) is considered the most important feature of the LightGBM model.

Table 7. Partial results of measuring feature importance and overall rankings for LightGBM

Feature	Importance (impurity)	Importance (lime)	Ranking
CompetitionDistance	43182 (1)	709.5813 (3)	1
Promo	13054 (4)	886.1849 (1)	2
DayOfWeek_1	5119 (7)	748.4422 (2)	3
CompetitionOpenElapsedDays	33031 (2)	98.6852 (12)	4
Month_12	4025 (11)	463.3038 (5)	5

Note that the ranking results of the two separately applied feature importance methods differ significantly, which supports the idea that the hybrid approach aggregating multiple feature importance is appropriate as a feature selection method.

C. Performance Analysis

In the performance analysis experiment, the lowest-ranking features were removed individually using the RFE method. For each iteration, the weighted mean absolute percentage error (WMAPE) value and training time were measured to derive a tradeoff according to the falling threshold β. Therefore, the β values that determine the tradeoff were divided into 0.05, 0.1, and 0.2 and applied to each experiment.

The performance analysis involves the basic performance measurements of models and additional performance measurements with Bayesian optimization (BO), which is widely used in model optimization. The search ranges of the hyperparameters in BO are presented in Tables 4 and 5.

To evaluate the accuracy of the prediction models, the WMAPE was used as a scale-independent metric.

$$WMAPE = \frac{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\frac{1}{n}\sum_{i=1}^{n} y_i}, \quad (12)$$

where n is the number of instances, and y_i and y_i are the observation and prediction in the ith instance, respectively.

Fig. 4 shows the performance measured by the RFR and the tradeoffs for each β value (5, 10, and 20%). In RFR, the training time was dramatically reduced as the number of features decreased. In contrast, when the number of features was less than 11, the WMAPE increased steeply because features of high predictive power were eliminated.

On the other hand, LightGBM trains the model hundreds of times faster than RFR because it performs pruning, which significantly reduces the amount of computation. However, in general, higher WMAPE values than those of RFR are obtained. A small feature subset leads to a reduction in training time. However, the effect on the overall performance is trivial.

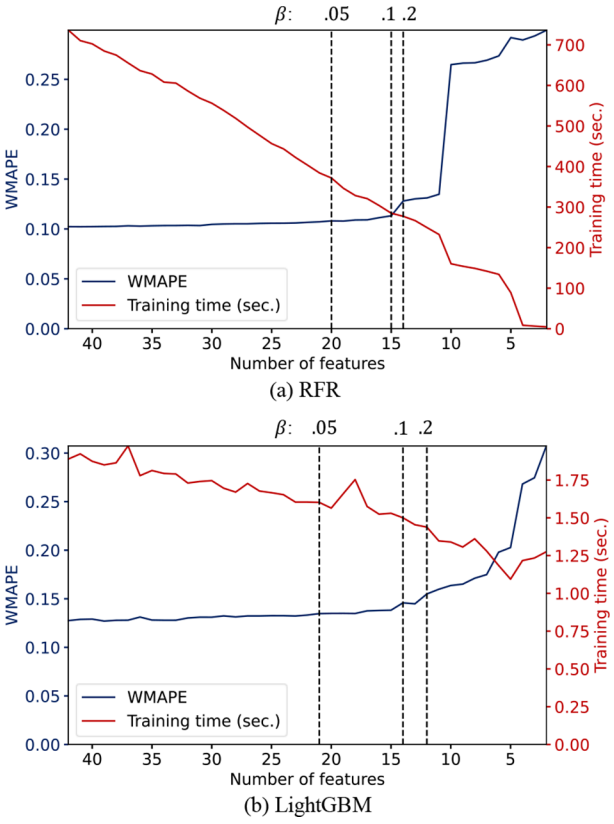

Fig. 4. Performance results obtained using RFE.

As both models are based on the ensemble approach, they naturally exhibit randomness and are sensitive to hyperparameter settings, resulting in a potentially volatile performance. In the experiment, we attempted to achieve a stable performance by applying BO to each basic model, and the results are shown in Fig. 5.

For both models, the results suggest that the degree of reduction in training time by feature elimination is inconsistent. In LightGBM, similar to Fig. 4, model training is completed much faster than in RFR. Therefore, the reduction in training time on a smaller feature subset is negligible. In the case of RFR, although significant training time is required owing to BO, the training time is considerably reduced while RFE is performed until near the beta value, whereas the predictive power of the model remains at an acceptable level.

The results of the overall performance analysis are presented in Table 8. In summary, the high computational cost of the RFR can be significantly reduced by RFE with an

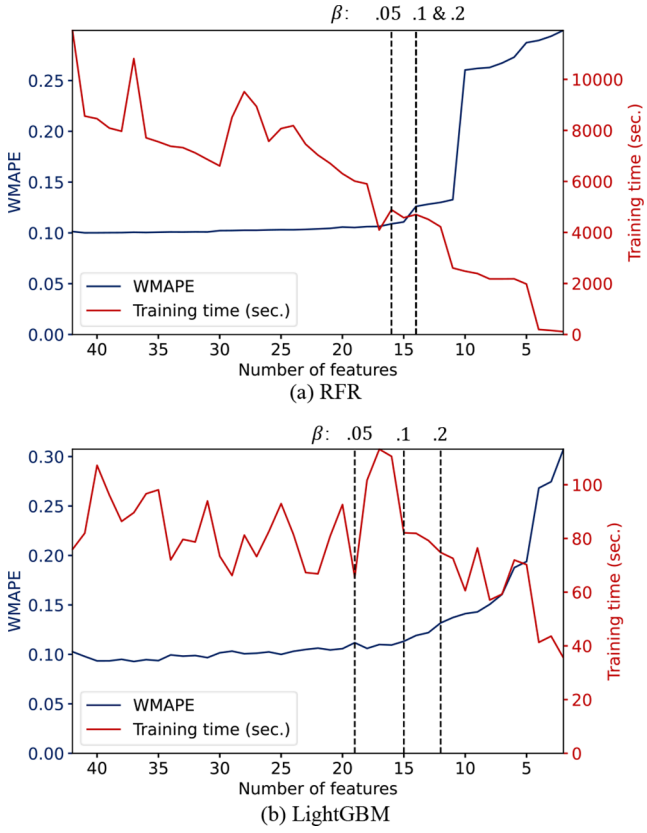

Fig. 5. Performance results obtained using RFE with BO.

appropriate beta. In particular, our feature selection method leads to a considerable reduction in training time in cases involving BO. This advantage allows us to derive tradeoffs that ensure reduced computational cost and acceptable model predictive power. In the case of LightGBM, as it is a computationally efficient algorithm, the validity of the tradeoffs discovered by the RFE is evaluated to be relatively less important.

VI. CONCLUSIONS

Data-intensive techniques, such as machine learning and deep learning models, which generally require massive amounts of computations, have been widely applied for time series prediction problems. In this regard, to find an appropriate tradeoff between computational cost and model predictive power, the effectiveness of the feature selection method using impurity- and XAI-based feature importance was evaluated in this study. The performance analysis was benchmarked against two machine learning models: RFR and LightGBM. In the case of RFR, which has a relatively high computational cost, it was confirmed that the feature-importance-based feature selection method affects the overall model performance more positively.

In the future, we plan to study an effective feature selection method that considers deep learning models with poor interpretability and high computational cost.

ACKNOWLEDGMENTS

This work was supported by Hanshin University Research Grant.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Dec. 1997. DOI: 10.1162/neco.1997.9.8.1735.
[2] Facebook, Prophet: Automatic forecasting procedure, [Online] Available: https://pypi.org/project/fbprophet.
[3] Kaggle, Rossmann store sales: Forecast sales using store, promotion, and competitor data, 2015. [Online] Available: https://www.kaggle.com/c/rossmann-store-sales.
[4] S. Kohli, G. T. Godwin, and S. Urolagin, "Sales prediction using linear and KNN regression," in *Advances in Machine Learning and Computational Intelligence*, Springer, Singapore, pp. 321-329, 2021. DOI: 10.1007/978-981-15-5243-4_29.
[5] T. Weng, W. Liu, and J. Xiao, "Supply chain sales forecasting based on light GBM and LSTM combination model," *Industrial Management & Data Systems*, vol. 120, no. 2, pp. 265-279, Sep. 2019. DOI: 10.1108/IMDS-03-2019-0170.
[6] Kaggle, [Online] Available: https://www.kaggle.com.
[7] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and TY. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, pp. 3149-3157, 2017.
[8] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1-45, Dec. 2017. DOI: 10.1145/3136625.

Table 8. Performance evaluation results

Models	WMAPE				# Features				Training time (s)			
β	Base	.05	.1	.2	Base	.05	.1	.2	Base	.05	.1	.2
RFR	.1023	.1072	.1113	.1132	42	21	16	15	736	384	304	284
LightGBM	.1275	.1332	.1382	.1448	42	22	15	13	1.89	1.60	1.53	1.45
RFR+BO	.1013	.1063	.1108	.1108	42	17	15	15	11,919	4,096	4,578	4,578
LightGBM+BO	.1027	.1058	.1095	.1220	42	20	16	13	76	93	111	79

[9] Y. Saeys, T. Abeel, and Y. V. Peer, "Robust feature selection using ensemble feature selection techniques," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Antwerp, Belgium, pp. 313-325, 2008. DOI: 10.1007/978-3-540-87481-2_21.

[10] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics*, vol. 10, no. 213, pp. 1-16, Jul. 2009. DOI: 10.1186/1471-2105-10-213.

[11] F. Pan, T. Converse, D. Ahn, F. Salvetti, and G. Donato, "Feature selection for ranking using boosted trees," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, Hong Kong, China, pp. 2025-2028, 2009. DOI: 10.1145/1645953.1646292.

[12] H. Jeon and S. Oh, "Hybrid-recursive feature elimination for efficient feature selection," *Applied Sciences*, vol. 10, no. 9, pp. 3211, May 2020. DOI: 10.3390/app10093211.

[13] L. Zhang and Q. Duan, "A feature selection method for multi-label text based on feature importance," *Applied Sciences*, vol. 9, no. 4, pp. 665, Feb. 2019. DOI: 10.3390/app9040665.

[14] E. Fezer, D. Raab, and A. Theissler, "XplainableClusterExplorer: a novel approach for interactive feature selection for clustering," in *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction*, Eindhoven, Netherlands, pp. 1-5, 2020. DOI: 10.1145/3430036.3430066.

[15] X. Man and E. P. Chan, "The best way to select features? comparing mda, lime, and shap," *The Journal of Financial Data Science*, vol. 3, no. 1, pp. 127-139, 2021. DOI: 10.3905/jfds.2020.1.047.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, pp. 1135-1144, 2016. DOI: 10.1145/2939672.2939778.

[17] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Montreal, Canada, vol. 1, pp. 278-282, 1995. DOI: 10.1109/ICDAR.1995.598994.

Hyun Ahn

received his B.S., M.S., and Ph.D. degrees in computer science from Kyonggi University in 2011, 2013, and 2017, respectively. He was a research professor at the Department of Computer Engineering, Kyonggi University from 2018 to 2021. He is an assistant professor at the Division of Software Convergence, Hanshin University. His research interests include process-aware information systems and data engineering.