Optimal hyperparameters for random forest to predict leakage current alarm on premises

Akihiro Yokoyama[1], and *Nobuyuki* Yamaguchi[2,*]

[1] Tokyo University of Science, Department of Engineering, 125-8585 Tokyo, Japan
[2] Tokyo University of Science, Graduate School of Engineering, 125-8585 Tokyo, Japan

Abstract. While the number of private electrical facilities is increasing, there are not enough security personnel to perform the security work. In this paper, we propose a random forest model for predicting leakage current alarms in order to improve the efficiency of electrical safety operations. A random forest was created using periodic inspection data, alarm data, and meteorological data as explanatory variables, and generalization performance was evaluated by OOB-based F-measure. In order to obtain the highest performance, a grid search was performed to optimize the hyperparameters. As a result, it was possible to achieve alarm prediction with a certain level of performance. In addition, the optimal hyperparameters were found by grid search, and the F-measure was improved.

1 Introduction

As the number of private electrical facilities increases, the maintenance has become important. The maintenance must be performed by qualified security personnel, but the amount of the workforce is gradually becoming insufficient. According to the Japanese Ministry of Economy, Trade and Industry, about 4,000 people are expected to be short of the expected demand of about 18,000 people in 2030. For this reason, it is important to increase the efficiency of electrical security operations.

In order to improve the efficiency of electrical security operations, we focused on *"false alarms"*. When the insulation monitoring device installed in each electrical facility detects a leakage current exceeding a certain value, an alarm is issued and the security personnel goes to the site for inspection. However, there are cases where the cause is unknown or only minor defects are found. This is "false alarm" and causes a reduction in the efficiency.

We aim to suppress unnecessary dispatch caused by *"false alarms"* by predicting alarms using Random Forest (RF). RF is one the most popular Machine Learning methods. The applications of RF, for examples, solar power forecast [1], electricity price forecast [2], building energy prediction [3, 4], and fault diagnosis for PV arrays [5]. The authors have conducted the prediction model with random forest [6, 7]. In this paper, we conducted a grid search to find the most suitable hyperparameters for accurate prediction.

2 Electrical security and insulation monitoring

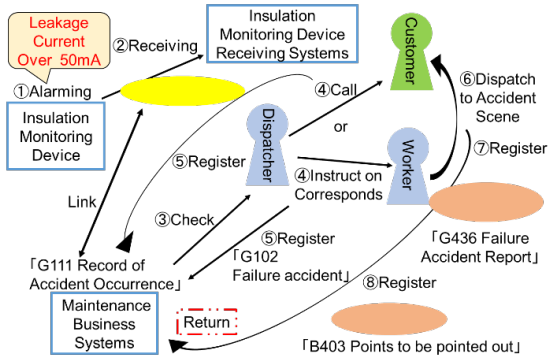


Fig. 1. Workflow and Insulation Monitoring Systems at the Accident.

2.1. Electrical security operations

Once a private electrical installation is installed, qualified security personnel must perform maintenance. The work of the security staff includes a monthly check once a month and an annual check once a year. In these periodic inspections, leakage current values are measured to check for abnormal insulation. In addition, a temporary inspection is conducted when an insulation alarm is triggered.

In this electrical accident response work, there is a case where a minor failure is detected or a failure is not found even though an alarm is received. The unnecessary dispatch caused by *"false alarm"* has reduced the efficiency of the electrical security service.

2.2. Insulation monitoring system

The electrical facility is monitored for insulation for 24 hours by an insulation monitoring device, which is

* Corresponding author: n-yama@rs.tus.ac.jp

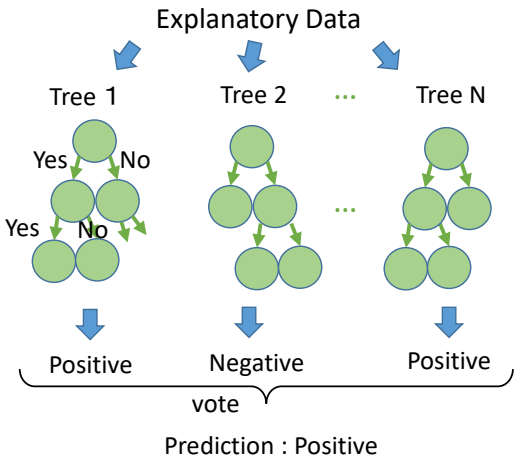


Fig. 2. Image of classification by random forest

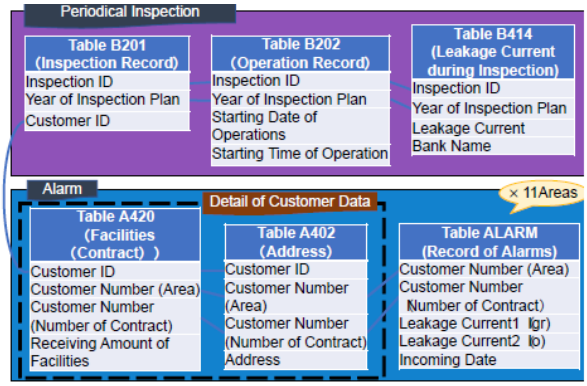


Fig. 3. Data tables for the proposed prediction models.

installed between the transformer and the electric facility. Fig. 1 shows an insulation alarms and the accident response work flow. The insulation monitoring device measures the current flowing into and out of the electric facility. Then, the leakage current is detected by calculating the difference. When the insulation monitoring device detects a leakage current exceeding 50 mA, an alarm is issued and the data is sent to the security service backbone system. Based on the data, security personnel go to the site and conduct a temporary inspection.

3 Random forest model

3.1. Random forest

Fig. 2 shows Random Forest consists of a large number of classification trees that are created with sets recursively applying two-conditional branching rules. The prediction is determined by classifying the data with the classification trees and taking the majority of them.

In this paper, the random forest model was implemented in Python scikit-learn package [8]. We used the periodic inspection data and alarm data provided by Kanto Electrical Safety Inspection Association (Fig 3) and the weather data released by the Japan Meteorological Agency. The target period is two years from April 2016 to March 2018, and the target areas are Ibaraki, Chiba, Tochigi, Gunma, Yamanashi, Saitama, Tokyo, Kanagawa and Eastern Shizuoka. Table 1 shows the objective variables and explanatory variables. These data were compiled and data containing defects were removed. As a result, the total was 338,933 data.

Table 1. Objective variable and explanatory variables

Objective variable	Explanatory variable
Flag represents alarm on the day	• Location • Leakage current at periodical inspection • Contracted demand • Alarm time • Flag represents alarm on the previous day • Customer ID • Hourly temperature • Hourly humidity • Hourly Precipitation

Table 2. Hyper parameters experimented in Grid Search.

Hyper Parameter	Number
n_estimators	100(default), 500, 550, 600
max_depth	2,5,10,20,None(default)
max_leaf_nodes	100,500,1000, 2000,5000,None(default)
max_features	"sqrt"(default), None

3.2. Grid search

To find the optimal hyperparameters for the random forest, a grid search was performed. The hyperparameters targeted for grid search are shown below [8].

(a) *n_estimators*

The number of decision trees contained in a random forest. The default is *100*.

(b) *max_depth*

The maximum depth of each decision tree. The default is *None* which means that classification is done until all nodes are pure or the number of samples in a node is a fixed number.

(c) *max_leaf_nodes*

The maximum number of leaf nodes created as a result of decision tree classification. The default is *None* which means that there is no limit on the number of leaf nodes.

(d) *max_features*

The maximum number of features used when creating each split. The default is *"auto"* which means that if the total number of the features is N, the maximum number of features used is $\sqrt{N}$. Other options include *"sqrt"* ($\sqrt{N}$, same as *"auto"*), *"log2"* ($\log_2 N$), and *None* (N).

Table 3. Comparison between the best result and the worst.

(a) The best result

n_estimators=550, *max_depth*=10, *max_leaf_nodes*=1000, *max_features="sqrt"*

		Prediction		Summation
		Negative	Positive	
Actual alarm	Negative	157520	47539	205059
	Positive	51592	82282	133874
Summation		209112	129821	338933
Accuracy = 0.70752		Precision = 0.63381		
Recall = 0.61462		F-measure = 0.62407		

(b) The worst result

n_estimators=500, *max_depth*=2, *max_leaf_nodes=None*, *max_features="sqrt"*

		Prediction		Summation
		Negative	Positive	
Actual alarm	Negative	162690	42369	205059
	Positive	61446	72428	133874
Summation		224136	114797	338933
Accuracy = 0.69370		Precision = 0.63092		
Recall = 0.54102		F-measure = 0.58252		

Table 2 shows the setting values of each hyper parameter for grid search.

3.3. Evaluation index

In order to evaluate the generalization performance of random forest, OOB verification was performed. When creating each decision tree in a random forest, the training instance is sampled using bootstrap. This means that about 37% of training instances are not sampled in each decision tree. These instances are called OOB instances. The performance of each decision tree is evaluated by OOB instances, and the generalization performance of the random forest is evaluated by averaging them [9].

In this study, we evaluated the accuracy, precision, recall, and F-measure obtained by OOB verification. First, the alarm prediction results are counted according to the following classification.

TP : Positive in prediction, Positive in actual
FP : Positive in prediction, Negative in actual
TN : Negative in prediction, Negative in actual
FN : Negative in prediction, Positive in actual

The accuracy rate is an index that shows how well the overall prediction results match the actual values, and is calculated by the following equation.

$$Accuracy = \frac{TP + TN}{TP + RN + FP + FN} \quad (1)$$

The precision rate is an index that shows how well the samples predicted to be positive match the actual value.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

The recall rate is an index that shows how well the samples that are actually positive match the predicted value.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F-measure is the harmonic mean of precision and recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

4 Results

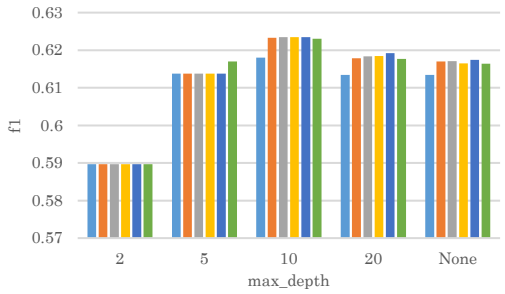
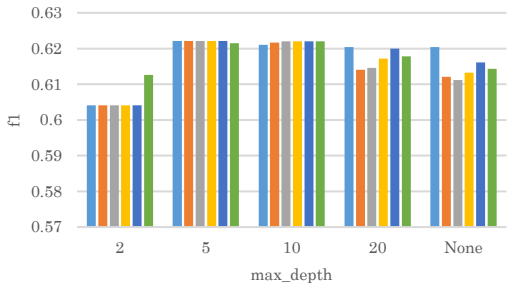
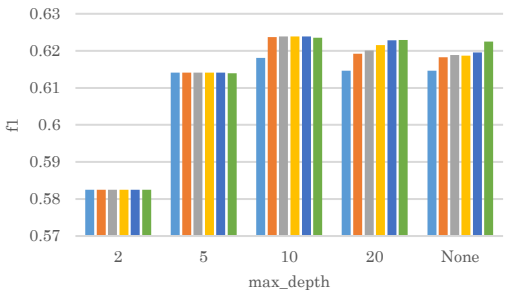
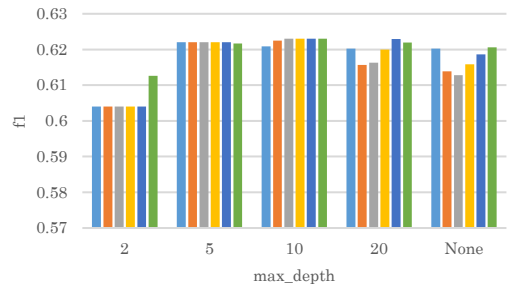
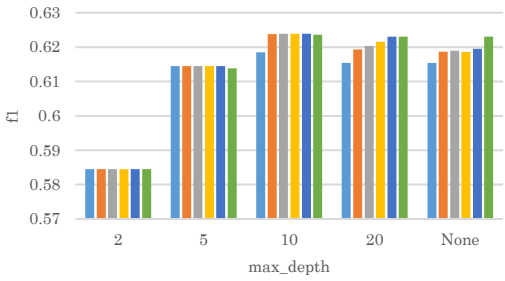
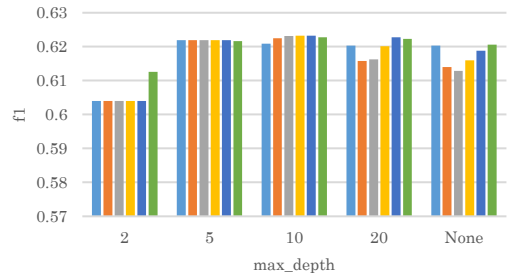
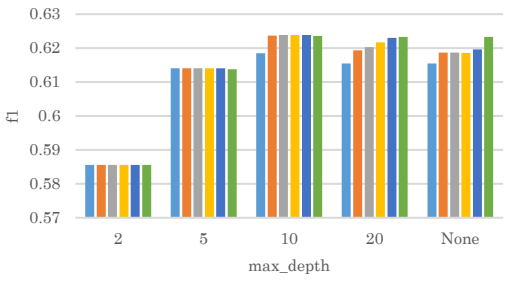
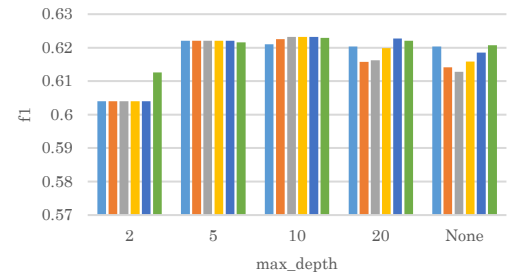
Table 3 shows the best and the worst results of alarm prediction. The best F-measure in this experiment is recorded when *n_estimators* is 550, *max_depth* is 10, *max_leaf_nodes* is 1000, and *max_features* is *"sqrt"*. Conversely, the worst F-measure is recorded when *n_estimators* is 500, *max_depth* is 2, *max_leaf_nodes* is *None*, and *max_features* is *"sqrt"*. In comparison, accuracy and precision are almost equal, but recall is improved by about 7.3%, and F-measure is improved by about 4.2%.

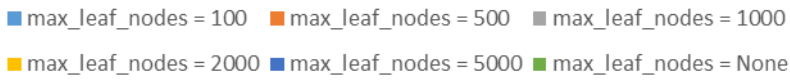
Table 4 shows the all results of alarm prediction on each grid search point. In order to compare the prediction accuracy, the vertical axis of the graph uses the F-measure obtained by OOB verification.

Focusing on *n_estimators*, the highest F-measure is recorded when 550 is set, and the F-measure decreases when *n_estimators* is increased or decreased. In addition, for certain *n_estimators*, the maximum value of the F-measure is higher when *max_features* is *"sqrt"* than when *None*. However, it can be seen that the effect on the F-measure is relatively smaller than other hyperparameters.

Focusing on *max_depth*, the F-measure is often the maximum when it is set to 10, and decreases both when it is made larger and smaller. Among the parameters verified in this experiment, the effect on the F-measure is

Table 4. The F-measure on each grid search point.

n_estimators	max_features	
	"sqrt"	None
100		
500		
550		
600		



the largest, indicating that the difference between the maximum and minimum values is about 4%.

Focusing on *max_leaf_nodes*, when *max_features* is *"sqrt"*, F-measure tends to improve when there are less restrictions such as large numbers or *None*. On the other hand, when *max_features* is *None*, the F-measure did not improve even if the restriction was relaxed.

Focusing on *max_features*, in the case of *"sqrt"*, the F-measure changes greatly depending on *max_depth*, and the F-measure improves as *max_leaf_nodes* increases. On the other hand, when *max_features* is *None*, the change in F-measure due to *max_depth* is slightly small, and a large value of *max_leaf_nodes* does not necessarily improve the F-measure. When comparing with the same *n_estimators*, the maximum value of the F-measure is higher for *"sqrt"* than for *None*.

5 Conclusion

We made a prediction model of the leakage current alarm for the next day using a random forest and optimized the hyperparameters by grid search. The highest score was recorded when *n_estimators*,

max_depth, *max_leaf_nodes,* and *max_features* were 550, 10, 1000, and "sqrt". The recall was improved to 0.61462 and F-measure recorded 0.62407. According to Eqn. (3), recall means that how well the prediction matches when the alarm actually issue. Recall's improve will reduce missing the alarm.

In this study, we predicted leakage current alarms in order to improve the efficiency of the security work for the increasing number of private electrical facilities. In addition, it can be expected that predicting the severity of alarms, that is, how serious failure has occurred, will lead to further improvement in the efficiency of electrical safety operations. In order to predict the severity of alarms, we will conduct verification such as creating a random forest with multiple labels for the objective variable.

References

1. D. Liu, K. Sun, *Random forest solar power forecast based on classifion optimization*, Energy, **187**, 15 (2019)
2. K. Wang, C. Xu, Y. Zhang, S. Guo, A. Y. Zomaya, *Robust Big Data Analytics for Electricity Price Forecasting in the Smart Grid*, IEEE Trans. Big Data, **5**, 1, 34-45 (2019)
3. Z. Wang, Y. Wang, R. Zeng, R. S. Srinivasan, S. Ahrentzen, *Random Forest based hourly building energy prediction*, Energy and Buildings, **171**, 15, 11-25 (2018)
4. M. W. Ahmad, M. Mourshed, Y. Rezgui, *Trees vs Neurons : Comparison between random forest and ANN for high-resolution prediction of building energy consumption*, Energy and Buildings, **147**, 15, 77-89 (2017)
5. Z. Chen, F. Han, L. Wu, J. Yu, S. Cheng, P. Liu, H. Chen, *Random forest based intelligent fault diagnosis for PV arrays using array voltage and string currents*, Energy Conversion and Management, **178**, 15, 250-264 (2018)
6. A. Yokote, N. Yamaguchi, K. Kato, M. Suzuki, *Prediction of Alarm of Insulation Monitoring System on Customer Facility using Random Forest*, IEEJ Transactions on Electronics, Information and Systems, **140**, 2 (2020) (Accepted, in Japanese)
7. A. Yokote, N. Yamaguchi, K. Kato, M. Suzuki, *Prediction method of leakage current at alarm report of customer facilityusing random forest*, The 2019 Annual Meeting of IEEJ, 4-235 (2019) (in Japanese)
8. scikit-learn: machine learning in Python (https://scikit-learn.org/stable/)
9. A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, (O'REILLY, 2017)