BEAM ADJUSTMENT BASED ON THE GRADIENT BOOSTING DECISION TREE ANALYSIS IN THE KEK ELECTRON/POSITRON INJECTOR LINAC

T. Sakai*[1], H. Ego[1], S. Shitara[1], S. Ushimoto[2], T. Abe[1], T. Higo[1], Y.Higashi[1]
[1]High Energy Accelerator Research Organization (KEK), Tsukuba, Japan
[2]Mitsubishi Electric System & Service Co., Ltd, Tokyo, Japan

Abstract

KEK-LINAC is an electron/positron linear accelerator used as the injector for the synchrotron radiation facilities (PF ring and PF-AR) and SuperKEKB. The stable operation of experiments at these facilities requires reliable beam supply from the LINAC. We have newly introduced an analytical method based on gradient boosting decision tree (GBDT) to further enhance our beam adjustment capability. GBDT is one of the machine learning methods and has been used as an exceptionally effective model for tabular data. The GBDT analysis handling hundreds of LINAC operating parameters accurately predicted beam-charge and position in the LINAC. Furthermore, by performing SHAP analysis, we have identified key parameters for the beam adjustment and correlations between the parameters. This provided a better understanding of the linac parameters that had previously been difficult to interpret.

INTRODUCTION

KEK-LINAC is a 700-meter S-band (2856 MHz) linear accelerator that generates multi-quality electron and positron beams. This accelerator consists of various components, and the beam conditions can be adjusted by changing the parameters of these components. However, it is difficult to tune all of the hundreds of parameters, and furthermore, it is unclear which parameters are effective for improving the beam quality; as a result, beam tuning relies on the intuition and experience of skilled operators. In this study, we trained a gradient-boosted decision tree (GBDT) [1, 2] using linac parameter data and performed a SHAP analysis to quantitatively evaluate which parameters contribute to the model prediction and how they affect the predicted beam behavior.

ANALYSIS WITH GBDT

Assuming that beam charge and beam size are determined by factors such as the klystron phase, pulse magnet current, ambient temperature, etc., and that time-dependent variations are negligible, the linac operating parameter can be treated as tabular data. In tabular data analysis, gradient-boosted decision trees(GBDT) [1, 2] is one of the most suitable methods and has been in use for a long time [3]. Furthermore, GBDT is a computationally efficient method. In this study, we employed XGBoost, one of the GBDT models [2]. XGBoost is particularly suitable for SHAP-based [4] interpretation, because TreeSHAP [5] provides an exact and efficient algorithm for computing SHAP values for tree ensemble models.

The input data for this study comprised phase and power readings at the klystron output, the current in the pulsed magnets, and the device and ambient temperatures. The input vector comprised approximately 200 parameters. The dataset contains about 100,000 records. The target variable for the machine learning model was the beam charge or the beam size downstream of the linac, or an objective function derived from these quantities.

In this study, the adjustable parameters were actively manipulated to investigate their effects on beam size and charge. For each parameter, a range was defined within the limits of previous operating data, and values were randomly sampled from a uniform distribution within that range.

Figures 1 and 2 compare the true and predicted values of the objective function derived from the beam charge and beam size downstream of the linac. These figures were generated using test data that was not used for training. The objective function J is defined as follows, with the first and second terms representing the beam size and charge terms, respectively:

$$J = \frac{\text{beamsize}}{3} + \left(\frac{1.95}{\text{charge} + \epsilon}\right)^3.$$

Since beam tuning aims to increase the charge and reduce the beam size, a smaller value of this function is considered preferable. The value 1.95 in the second term of J represents the target charge. This term is cubed in order to impose a strong penalty on charge loss. J is preliminary and may be refined or replaced by a more appropriate objective function in future work. As shown in Fig. 2, several data points exhibit inaccurate predictions. This discrepancy is partly attributable to the limited accuracy of the machine learning model, it is also likely affected by the uncertainty in the beam size data itself. The beam size is obtained by fitting the screen image; however, because a Gaussian function is used to fit beam profiles that are not necessarily Gaussian, the fitted values may deviate from the actual beam size.

SHAP ANALYSIS

SHAP analysis was performed to interpret the trained machine learning model and to identify the parameters that strongly contributed to the prediction. SHAP, which is based on Shapley values from game theory, quantifies the contribution of each feature to the model prediction and provides a measure of the feature importance. Here, the term “feature” has the same meaning as “parameter”. Since the term

* sakai520@post.kek.jp

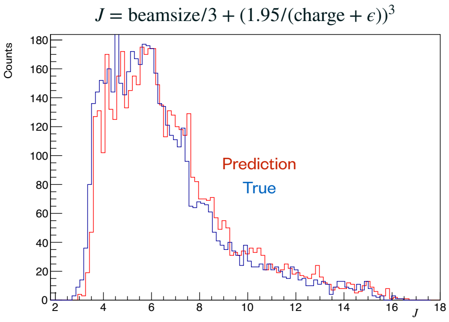

Figure 1: Histogram of prediction and true data.

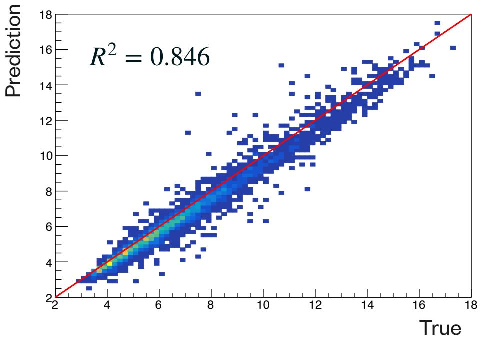

Figure 2: Comparison plot of prediction and true data.

"feature" is commonly used in machine learning, it is used throughout this section to refer to the input parameters of the model. The relationship between the SHAP values and the model prediction is expressed as follows

$$f(\mathbf{x}) = \phi_0 + \sum_{i=1}^{N_{\text{features}}} \phi_i,$$

where $f(\mathbf{x})$ is the predicted value, ϕ_0 is the base value, and ϕ_i is the SHAP value of the i-th feature. The base value represents the expected model output, which corresponds to the average prediction over the reference dataset.

Figure 3 shows the results of the SHAP analysis as a beeswarm plot. In the plot, each point represents a single data sample, and the horizontal axis indicates the SHAP value. A positive SHAP value increases J, whereas a negative SHAP value decreases it. The features are ordered according to their importance, with the most influential features shown at the top. The color of each point represents the feature value, where red and blue indicate high and low values, respectively.

Figure 4 shows a detailed SHAP dependence plot for pxa1m (one of the pulse magnets), which was ranked as the most important feature in Fig. 3. The plot indicates that higher current values of pxa1m are associated with lower SHAP values. Thus pxa1m contributes to reducing the predicted value of J within the trained model. It should be noted that this result does not necessarily imply that setting the pxa1m current to approximately −0.25 is universally optimal. This behavior may depend on the correlations and interactions between pxa1m and other features within the trained model.

SHAP values are evaluated in the context of the other input features. Figure 5 is an example illustrating the relationship between two features of pxa1m and another pulse magnet, pxc74 and their SHAP values. The horizontal axis represents pxa1m, the vertical axis represents pxc74 and the color scale indicates to the sum of their SHAP values. This visualization helps identify regions in the two-feature space that are expected to contribute to a reduction in J.

It should be noted that the SHAP values indicate feature contributions within the trained model and do not necessarily imply causal relationships. In this study, the analysis focused on parameters that were actively manipulated. However, care must be taken when interpreting parameters that vary due to multiple factors, such as environmental changes or feedback control, because their SHAP values may reflect correlations rather than direct causal effects.

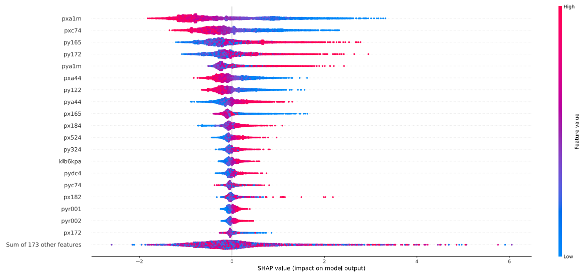

Figure 3: SHAP beeswarm plot.

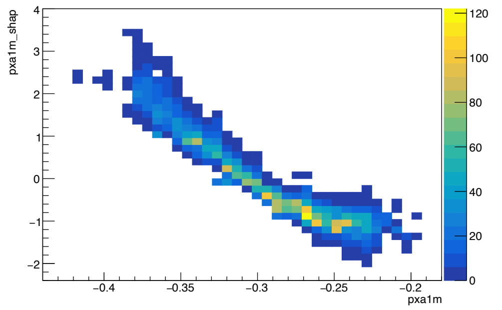

Figure 4: Distribution of SHAP values of pxa1m with respect to pxa1m values.

CONCLUSIONS

In this study, we applied an analysis using GBDT and SHAP to KEK-LINAC data. This method enables the identification of key parameters and promising parameter regions that are expected to improve the objective function. This provided a better understanding of the linac parameters that had previously been difficult to interpret. This facilitates more efficient optimization using Bayesian optimization and supports daily beam tuning by operators.

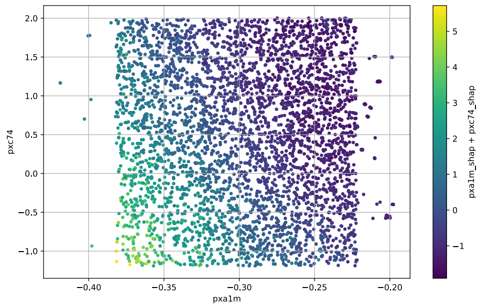

Figure 5: pxa1m vs pxc74 SHAP distribution.

REFERENCES

[1] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Ann. Stat., vol. 29, no. 5, pp. 1189–1232, 2001. doi:10.1214/aos/1013203451

[2] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD’16), San Francisco, CA, USA, Aug. 2016, pp. 785–794. doi:10.1145/2939672.2939785

[3] L. Grinsztajn, E. Oyallon, and G. Varoquaux, “Why do tree-based models still outperform deep learning on typical tabular data?” in Adv. Neural Inf. Process. Syst., vol. 35, New Orleans, LA, USA, Nov.–Dec. 2022, pp. 507–520. doi:10.52202/068431-0037

[4] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Adv. Neural Inf. Process. Syst., vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 4765–4774.

[5] S. M. Lundberg, G. G. Erion, and S.-I. Lee, “Consistent individualized feature attribution for tree ensembles,” arXiv:1802.03888, 2018. doi:10.48550/arXiv.1802.03888