ICIC Express Letters
Part B: Applications
Volume **15**, Number **2**, February **2024**

ICIC International ©2024 ISSN 2185-2766
pp. **109–116**

DISCOVERING THE BEST INTERVAL TRAINING SET FOR RAINFALL PREDICTION USING BAYESIAN OPTIMIZATION AND ENSEMBLE MACHINE LEARNING

Prabowo Wahyu Sudarno, Ahmad Ashari* and Mardhani Riasetiawan

Department of Computer Science and Electronics
Universitas Gadjah Mada
Sekip Utara PO BOX BLS 21, Yogyakarta 55281, Indonesia
prabowowahyu45@mail.ugm.ac.id; mardhani@ugm.ac.id
*Corresponding author: ashari@ugm.ac.id

Received April 2023; accepted July 2023

Abstract. *Heavy rains in Indonesia occur yearly, and one of the impacts is flood disasters. Flooding occurs frequently and unpredictably. Modelling such important occurrences can help to identify vulnerable locations and reduce the effects. Recently, researcher applied machine learning to analyzing data and its correlations in order to predict how the climate will perform. However, most machine learning algorithms cannot automatically detect the dataset's quality; for example, how long the time interval for the dataset to make good forecasting predictions is. Using ensemble machine learning and Bayesian optimization, we explored for the best interval and model to predict rainfall. The ensemble machine learning algorithm achieved the best result, showing the superiority of ensemble machine learning over single machine learning in discovering the best interval training set for rainfall prediction. The best interval to predict rainfall is 61-hour, with mean squared error score of 12.97 and mean absolute error score of 2.24.*
Keywords: Ensemble machine learning, Stacking, Extreme gradient boosting, Multilayer perceptron, Bayesian optimization, Floods

1. **Introduction.** Floods caused by various natural factors, occur frequently without warning and pose challenges for disaster managers and environmental scientists [1]. Simulating these events is complex and time-consuming. Accurate flood modeling relies on reliable rainfall data and aids in identifying vulnerable areas for effective mitigation measures [2]. However, weather forecasting is challenging due to its continuous, data-intensive, complex, dynamic, and chaotic nature [3]. In recent years, researchers have shifted from using prediction models based on random numbers to leveraging machine learning for rainfall prediction. Machine learning techniques employ various statistical approaches and learning processes to extract valuable insights from data [4]. For instance, analyzing historical rainfall data enables the estimation of upcoming rainfall [5]. However, assessing dataset quality, including factors like the appropriate time interval for accurate predictions, remains a challenge for most machine learning algorithms.

Additionally, machine learning models may not necessarily be applied directly in all cases. Selecting the right prediction, regression, or classification model can be a complex task that requires a deep understanding of the dataset. In regression problems, users often face difficulties in selecting the most suitable model for their dataset and ensuring effective data cleaning for optimal machine learning outcomes [1]. In the context of weather forecasting, we need to know how long is the best interval to train the data [2]. The longer the interval range is not necessarily good results; even if the interval is long, the training process will take longer because it will require more training data. On the other hand, a shorter interval may be easier to process, but may yield suboptimal result. From the

DOI: 10.24507/icicelb.15.02.109

previously described problems, this becomes the first objective of this research, and we try to find the best range interval and its tuning to achieve maximum results in predicting rainfall to prevent flooding. Our second objective is to produce a robust machine learning rainfall predictor.

Bayesian Optimization (BO) is a robust and adaptable method for finding the best optimization hyper-parameter tuning. This method is mainly used when tuning a machine learning model's parameter with complex optimization problems. At each stage of the search, BO evolves by choosing an area to sample and analyze. To maximize the outcome, BO makes use of a replacement model for the function. By employing what is known as an acquisition function, this surrogate model is utilized to choose the next point to be assessed. In deciding which point to move on to, this acquisition function will weigh between exploration against exploitation [3]. In this research, we implemented BO to automatically find the optimal interval, eliminating the need to select interval ranges manually.

Another common issue in machine learning computation is the dataset's quality. The poor quality of the dataset could have a significant negative result while building a machine learning model. One method to solve dataset's quality problems is the application of ensemble machine learning. This method provides superior prediction outcomes compared to a single algorithm [4]. Because of the limitations of a single classifier approach, models may be built incorrectly, which will lower the quality of the conclusion [5]. Meta-algorithms are used in ensemble approaches to transform weak base estimators into stronger classifiers. The heterogeneous ensemble strategy was used by the majority of researchers since it produces better outcomes [6].

In this research, we utilize Random Forest (RF) and Extreme Gradient Boosting (XGB) algorithms as base regressors, known for their effectiveness in solving tabular data and achieving good performance [7]. XGB is particularly robust for regression tasks like rain forecasting, sequentially building trees to minimize errors [8]. RF reduces overfitting and yields improved results. By combining BO with ensemble machine learning, we aim to obtain better interval values and robust machine learning models for rainfall prediction. Our research introduces two key novelties. First, automatic identification of the optimal rainfall interval, eliminating the need for manual selection. Second, the development of a robust ensemble machine learning model for accurate rainfall prediction, applicable in flood prevention, tourism, transportation, and more. This study is organized into sections covering the dataset (Section 2), methods (Section 3), results and discussion (Section 4), and conclusion (Section 5).

2. **Data Processing.** This section provides an overview of the dataset used and the data preprocessing steps. The dataset used in this research is sourced from Visual Crossing Weather Data and consists of 8,784 instances with 5 feature attributes, including temperature, humidity, precipitation, wind speed, and sea level pressure. To address missing and null data, necessary steps were taken to fill in the missing values and correct outliers. This ensures the data is appropriately processed. The imputation method was used to generate additional training data tuples, enhancing the classification performance of the model [9]. The K-Nearest Neighbor (KNN) method was employed to fill in null data and remove outliers, as visualized. To ensure proper evaluation, the training and testing data were split in an 80% : 20% ratio using the train-validation split strategy, enabling the model to learn from the training data throughout the training process.

3. **Methods.** This section explains about method utilized in this study, explains each stage of the process, describes the strengths and weaknesses of the method, and explains how measurements and computations were produced.

3.1. **Stacking method.** Stacking is an ensemble method that utilizes the results of several classification algorithms to provide a more precise forecast [10]. In order to get the final prediction, the procedure begins with acquiring the results predicted by a series of diverse base models, followed by optimally merging the outputs from the base models using a meta-learner [11]. The stacking approach also performs well compared to a single model without ensemble technique, as multiple models can work together to reduce the risk of overfitting and improving the performance [12].

3.2. **Base regressor and meta regressor.** Stacking uses numerous base regressors and a single meta regressor in the learning process. The base regressor is trained on the same dataset and then produces distinct and unique prediction results. Afterwards, the meta regressor predicts the final result by taking in the base regressors' result as its features. We use RF and XGB as the base regressors.

1) RF is one of the methods for regression, classification, and other tasks that use an ensemble learning approach. The algorithm works by assembling decision trees during training and providing a class representing the classification or regression of the different trees [12]. This method randomly selects m features from a total of n data, where k number of distinct decision trees are trained on distinct random data [13]. By doing this, RF gives excellent results while maintaining its simplicity and versatility [14]. RF's unique qualities improve prediction stability and accuracy while avoiding correlation across several regression trees.
2) XGB is an improved distributed gradient boosting method that is more effective, adaptable, and portable. XGB regressor boosts trees in parallel to answer a large number of regression problems properly and quickly by improving the prediction results of weak models using a structural loss function [15]. In addition, pre-sorting, weighted quantile, and identifying sparse matrices in XGB enhances the algorithm's performance [16]. The objective function is to find the nearest function $\hat{f}_x$ to constructor functions f_x by reducing the loss function value $L(y, f(x))$ in the following equation [17]:

$$\hat{f} = \arg\min \sum L(y, f(x)) \tag{1}$$

Each repetition of the training procedure reduces the loss function value with the starting function $f_0(x)$, where Equation (1) can be extended even more as follows:

$$y_m h_m = \arg\min \sum_{m=1}^{M} L_m, \quad L_m(y, f(x)) = L\left(y_i, f^{m-1}(x_i) + y_m h_m(x_i)\right) \tag{2}$$

where y, M, f, and h_m represent the real value known from the training dataset, the boosting steps number, the imperfect model, and the estimator, respectively.

3.3. **Bayesian optimization.** This method uses the Bayesian approach, which uses the chance of failure as a metric of the degree of uncertainty about potential failures and results [18]. We utilized BO to tune the ensemble models' hyperparameters during the training process, as it outperformed several popular techniques like the grid-search, manual search, and random search [19]. The objective function of BO is as follows:

$$X = \arg\max f(x), \quad x \in X \tag{3}$$

where f is defined as a non-closed form black-box function and $f : X \to R$ is a function that is defined under the subset of $X \subseteq R_d$. BO aims to find the highest value of the black-box function $f(x)$ with a probabilistic model for $f(x)$. Afterwards, the model is employed to generate results by assessing the function under uncertain conditions. These results can identify the least complex non-convex functions with minimal evaluations, although this requires additional computational resources. The BO algorithm is outlined as follows [20].

Input:	init data D_0
Process:	For $t = 1$ to T do:
	• Fit a Gaussian Process from D_t and obtain $x_t = \arg\max a(x)$
	• Evaluate $y_t = f(x_t)$ and $D_t = D_{t-1} \cup (x_t, y_t)$
	End
Output:	$x_{\max}$, $y_{\max}$

To fit the Gaussian Process from D_t and obtain $x_t = \arg\max a(x)$, we utilize the Gaussian Process to model an unknown target function. The acquisition function $a(x)$ is used to select the optimal point for evaluation based on the trade-off between exploitation (choosing points with high values) and exploration (choosing points with high uncertainty). x_t represents the selected point for the next evaluation. Next, we evaluate $y_t = f(x_t)$, where f is the target function we aim to optimize. Then, we update $D_t = D_{t-1} \cup (x_t, y_t)$, which combines the previous data D_{t-1} with the latest evaluation result (x_t, y_t). This update aims to refresh the data used to train the Gaussian Process in the subsequent iteration. Repeat steps 1 to 3 until reaching the final iteration T. The output $x_{\max}$, $y_{\max}$ represents the point with the highest value discovered during the iterations. In BO algorithm, these steps are used to search for the maximum value of a target function f through adaptive iterations and intelligent selection of evaluation points based on Gaussian Process and the acquisition function.

3.4. **Performance validation.** In order to determine performance model in a regression task, the Mean Squared Error (MSE) and Mean Absolute Error (MAE) are used as evaluation measures. Equation (4) corresponds to calculating MSE, which quantifies the average squared difference between predicted and actual values. Equation (5) represents the MAE, which measures the average absolute difference between predicted and actual values. y is the expected label, $\hat{y}$ is the predicted label, and n is the number of data in the dataset.

$$MSE_{(y,\hat{y})} = \frac{1}{n}\sum_{i=0}^{n-1}(y - \hat{y})^2 \tag{4}$$

$$MAE_{(y,\hat{y})} = \frac{1}{n}\sum_{i=0}^{n-1}|y - \hat{y}|^2 \tag{5}$$

4. **Results and Discussion.** First, we fine-tune the hyperparameters for both the RF and XGB. This ensures that we use the best RF and XGB models for the ensemble stacking later. The hyperparameters and their values are tabulated in Table 1 for RF and Table 2 for XGB.

To create an ensemble stacking model, we combined the previously developed RF and XGB models. Subsequently, we trained the ensemble model using different interval values,

Table 1. RF parameter tuning

Parameter	Value	Description
Max depth	170	The longest path between the leaf node and the root node.
Max features	0.5174	The number of features for the best split.
Min samples leaf	0.1586	The number samples in the leaf node after a node has been split.
Min sample split	0.3111	The minimum number of splitting an internal node.
N estimators	451	The number of trees before taking the maximum voting.

Table 2. XGB parameter tuning

Parameter	Value	Description
Eta	0.0067	The value of step size prevents overfitting.
Col sample by tree	0.7086	The subsampling ratio column once when constructing every tree.
Max depth	13	The maximum depth of tree.
Min child weight	5.3843	Minimum total of weights in a child
Subsample	0.6427	The training instance subsampling ratio.

Table 3. Best interval result

Interval (hour)	MSE
10	13.97
41	13.89
33	13.88
78	13.73
10	13.97
86	13.55
29	13.46
17	13.40
26	13.37
61	**12.97**

utilizing BO to determine the optimal intervals. This approach was crucial as manually selecting specific interval spans would result in longer training times.

By employing BO in conjunction with RF and XGB stacking algorithms, we generated the top ten intervals, as outlined in Table 3. Notably, our study achieved the best predictive model by utilizing 61-hour intervals, resulting in an impressive MSE score of 12.97.

We conducted ablation analysis on the 61-hour intervals to confirm the superiority of our ensemble model over standalone RF and XGB models. Without ensemble stacking, RF method yielded MSE and MAE scores of 25.72 and 3.59, respectively. Without ensemble stacking, XGB method achieved MSE and MAE scores of 17.14 and 2.61. However, by employing stacking ensemble machine learning with RF and XGB algorithms on the 61-hour intervals, we achieved improved performance with MSE and MAE scores of 12.97 and 2.24, respectively. Please refer to Figure 1 for visualizations of these results.

Our findings show that the ensemble stacking method produces better MSE and MAE performance when compared to the method without stacking. This result indicates that the stacking method can optimize the performance of a regression model (which in this study: finding the best interval to predict rain precipitation). The comparison between the models is tabulated in Table 4.

5. **Conclusions.** Different climate datasets and geographic locations can lead to varying interval results for rainfall prediction. However, determining optimal intervals using range-based values can be suboptimal and time-consuming, requiring iteration over various intervals and diverse rainfall data. In this research, we successfully combine a robust ensemble model with BO to automatically search for the best rainfall interval, resulting in improved prediction performance. Our study identified the optimal 61-hour interval, producing an MSE score of 12.97 with ensemble stacking of RF and XGB models. This outperforms the individual RF and XGB models, which yield MSE scores of 25.72 and

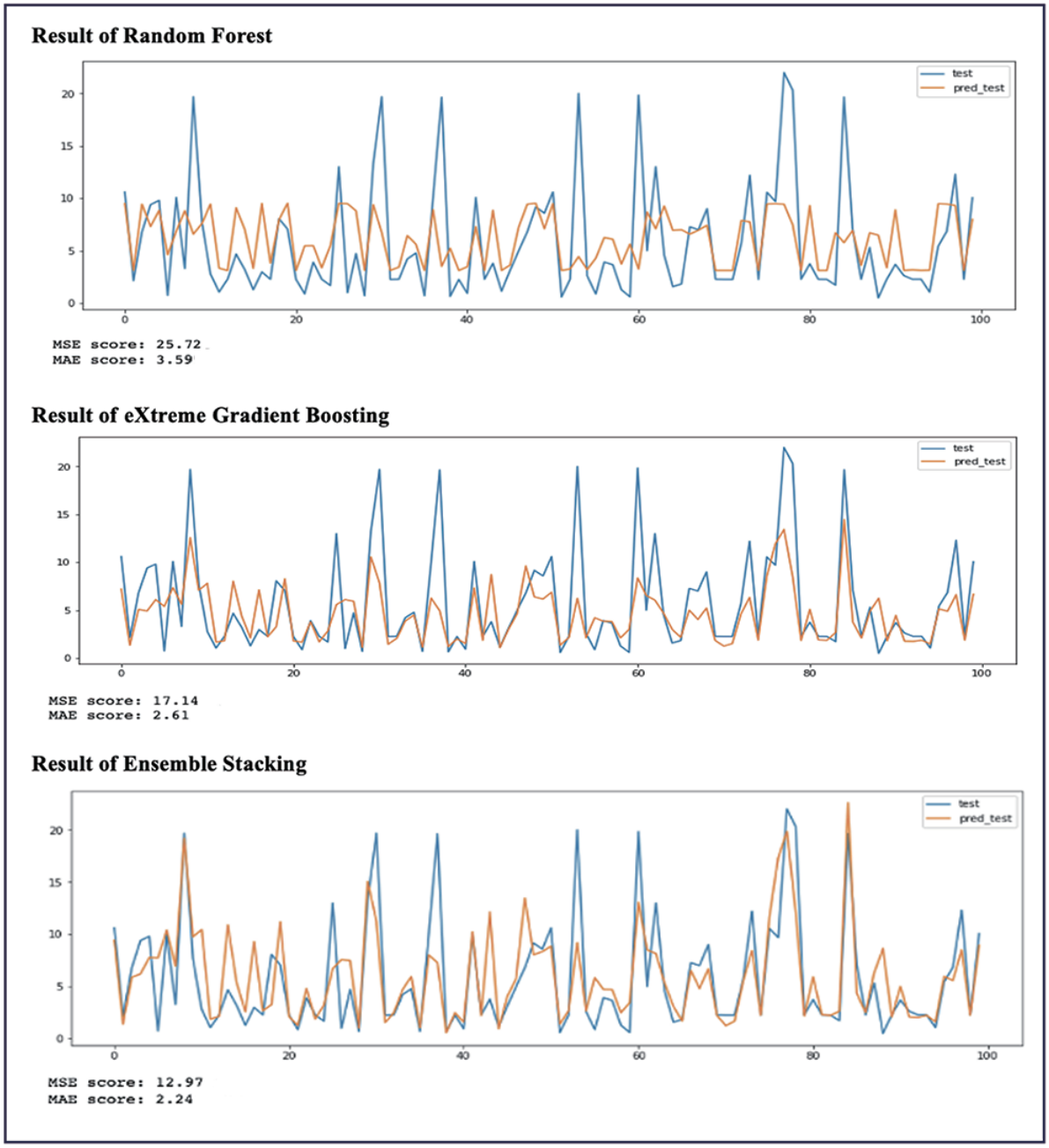

FIGURE 1. Best interval result

TABLE 4. Results comparison throughout algorithms

Algorithm	MSE	MAE
RF	25.72	3.59
XGB	17.14	2.61
Stacking	**12.97**	**2.24**

17.14, respectively, highlighting the superiority of ensemble stacking in rainfall prediction. We recommend exploring broader methods for ensemble stacking and extending the application of this approach to different domains.

Acknowledgment. Researchers recognize the National Research and Innovation Agency of the Republic of Indonesia, the Directorate of Research and Community Service, and the Deputy for Strengthening Research and Development in the PMDSU program.

REFERENCES

[1] H. Riza, E. W. Santoso, I. G. Tejakusuma and F. Prawiradisastra, Advancing flood disaster mitigation in Indonesia using machine learning methods, *Proc. of the 7th International Conference on ICT for Smart Society (ICISS 2020)*, Bandung, Indonesia, DOI: 10.1109/ICISS50791.2020.9307561, 2020.

[2] V. A. Rangari, K. V. Gopi, U. V. Nanduri and R. Bodile, ANN based scaling of rainfall data for urban flood simulations, *Proc. of the 1st IEEE Bangalore Humanitarian Technology Conference (B-HTC 2020)*, DOI: 10.1109/B-HTC50970.2020.9297866, 2020.

[3] R. Vijayan, V. Mareeswari, P. Mohankumar and G. G. K. Srikar, Estimating rainfall prediction using machine learning techniques on a dataset, *International Journal of Scientific & Technology Research*, vol.9, no.6, pp.440-445, 2020.

[4] A. Faruk, E. S. Cahyono, N. Eliyati and I. Arifieni, Prediction and classification of low birth weight data using machine learning techniques, *Indonesian Journal of Science and Technology*, vol.3, no.1, DOI: 10.17509/ijost.v3i1.10799, 2018.

[5] M. Mohammed, R. Kolapalli, N. Golla and S. S. Maturi, Prediction of rainfall using machine learning techniques, *International Journal of Scientific & Technology Research*, vol.9, no.1, 2020.

[6] S. Patankar, H. Prajapati, J. Shah and A. Upadhyay, AutoML – Learning, understanding and applying machine learning to datasets, *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS 2023)*, pp.919-922, DOI: 10.1109/ICACCS57279.2023.10113008, 2023.

[7] M. M. Jazzar, Flood congestion simulation and prediction using IOT wireless networks on dynamic streets routes, *J. Theor. Appl. Inf. Technol.*, vol.99, no.10, 2021.

[8] I. Roman, J. Ceberio, A. Mendiburu and J. A. Lozano, Bayesian optimization for parameter tuning in evolutionary algorithms, *2016 IEEE Congress on Evolutionary Computation (CEC 2016)*, DOI: 10.1109/CEC.2016.7744410, 2016.

[9] T. Zhang and J. Li, Credit risk control algorithm based on stacking ensemble learning, *Proc. of 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA 2021)*, DOI: 10.1109/ICPECA51329.2021.9362514, 2021.

[10] W. D. Ahmad and A. A. Bakar, Ensemble machine learning model for higher learning scholarship award decisions, *International Journal of Advanced Computer Science and Applications*, vol.11, no.5, DOI: 10.14569/IJACSA.2020.0110540, 2020.

[11] N. M. Baba, M. Makhtar, S. A. Fadzli and M. K. Awang, Current issues in ensemble methods and its applications, *Journal of Theoretical and Applied Information Technology*, vol.81, no.2, 2015.

[12] D. Martin and S. S. Chai, A study on performance comparisons between KNN, Random Forest and XGBoost in prediction of landslide susceptibility in Kota Kinabalu, Malaysia, *Proc. of 2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC 2022)*, pp.159-164, DOI: 10.1109/ICSGRC55096.2022.9845146, 2022.

[13] M. T. Anwar, E. Winarno, W. Hadikurniawati and M. Novita, Rainfall prediction using Extreme Gradient Boosting, *Journal of Physics: Conference Series*, DOI: 10.1088/1742-6596/1869/1/012078, 2021.

[14] B. Tarle and M. Akkalakshmi, View of improving classification performance of neuro-fuzzy classifier by imputing missing data, *International Journal of Computing*, vol.18, no.4, pp.495-501, 2019.

[15] M. A. Muslim and Y. Dasril, Company bankruptcy prediction framework based on the most influential features using XGBoost and stacking ensemble learning, *International Journal of Electrical and Computer Engineering*, vol.11, no.6, DOI: 10.11591/ijece.v11i6.pp5549-5557, 2021.

[16] S. Sridhar, S. Mootha and S. Kolagati, A university admission prediction system using stacked ensemble learning, *Proc. of 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA 2020)*, DOI: 10.1109/ACCTHPA49271.2020.9213205, 2020.

[17] R. Al-Hajj, A. Assi and M. M. Fouad, Stacking-based ensemble of support vector regressors for one-day ahead solar irradiance prediction, *The 8th International Conference on Renewable Energy Research and Applications (ICRERA 2019)*, DOI: 10.1109/ICRERA47325.2019.8996629, 2019.

[18] S. Raghavendra and J. S. Kumar, Performance evaluation of random forest with feature selection methods in prediction of diabetes, *International Journal of Electrical and Computer Engineering*, vol.10, no.1, DOI: 10.11591/ijece.v10i1.pp353-359, 2020.

[19] U. Harita, V. U. Kumar, D. Sudarsa, G. R. Krishna, C. Z. Basha and B. S. S. P. Kumar, A fundamental study on suicides and rainfall datasets using basic machine learning algorithms, *Proc. of the 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA 2020)*, DOI: 10.1109/ICECA49313.2020.9297440, 2020.

[20] N. J. Ria, J. F. Ani, M. Islam and A. K. M. Masum, Standardization of rainfall prediction in Bangladesh using machine learning approach, *2021 12th International Conference on Computing*

Communication and Networking Technologies (ICCCNT), Kharagpur, India, DOI: 10.1109/ICCCNT51525.2021.9579472, 2021.

[21] X. Liao, N. Cao, M. Li and X. Kang, Research on short-term load forecasting using XGBoost based on similar days, *Proc. of 2019 International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS 2019)*, DOI: 10.1109/ICITBS.2019.00167, 2019.

[22] G. V. Sajan and P. Kumar, Forecasting and analysis of train delays and impact of weather data using machine learning, *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, DOI: 10.1109/ICCCNT51525.2021.9580176, 2021.

[23] T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, DOI: 10.1145/2939672.2939785, 2016.

[24] A. Guzman-Urbina, A. Aoyama and E. Choi, A polynomial neural network approach for improving risk assessment and industrial safety, *ICIC Express Letters*, vol.12, no.2, pp.97-107, DOI: 10.24507/icicel.12.02.97, 2018.

[25] W. Yotsawat, P. Wattuya and A. Srivihok, Improved credit scoring model using XGBoost with Bayesian hyper-parameter optimization, *International Journal of Electrical and Computer Engineering*, vol.11, no.6, DOI: 10.11591/ijece.v11i6.pp5477-5487, 2021.