Transformer Oil Temperature Forecasting Based on Multi-model by Stacking Ensemble Learning

Jianpeng Hu, Lichen Zhang

School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China

1284219022@qq.com

Keywords: Oil temperature forecasting, Power transformer, Stacking, LightGBM, CatBoost

Abstract: The power transformer is a key piece of equipment in power plants and substations. However, abnormal oil temperature in power transformers accelerates insulation aging, shortening their lifespan and leading to accidents. Therefore, predicting and monitoring oil temperature is crucial. To overcome the reliance on experience and rules in traditional prediction methods, an oil temperature forecasting method based on a multi-model combination under the Stacking framework was proposed. Taking into account the differences in data observation and training principles among various algorithms, fully leveraging the strengths of each model, we construct a Stacked ensemble learning oil temperature prediction model embedded with multiple machine learning algorithms. The base learners of the model include Light Gradient Boosting Machine (LightGBM) and Category Boosting (CatBoost).The experimental results indicate that the oil temperature prediction method based on the Stacking ensemble learning approach with multiple model fusion achieves a high level of prediction accuracy.

1. Introduction

Accurate assessment of power transformer load capacity and insulation life is critical for safety and depends on accurate prediction of abnormal oil temperatures.[1]

Currently, oil temperature prediction relies on formula derivation, but these formulas often have strict conditions and overlook the impact of environmental changes, requiring improved accuracy[2-3]. Jia X et al. employed numerical methods to construct a theoretical thermal model for transformers based on formula derivation[4]. However, this model involves numerous condition parameters, often requiring collaboration from multiple parties for practical engineering applications.

In recent years, applying machine learning methods for predicting transformer oil temperature has become common. Alerskans E et al. employed neural networks for top-layer oil temperature prediction[5], but the non-linear fit between inputs and outputs lacks interpretability and physical significance, leading to potential overfitting. Li P et al. utilized a Support Vector Machine (SVM) model to construct an oil temperature prediction model and achieved favourable outcomes[6]. The cited literature singularly employed a predictive model for oil temperature, however, considering the extensive hypothesis space in oil temperature prediction, multiple hypotheses could yield comparable performance on the training set.

While depending on a single model may jeopardize generalization performance due to its inherent stochastic nature, the Stacking prediction approach emphasizes the diversity in data observations among predictive algorithms. This allows for the training of a superior composite model that harnesses different strengths and addresses the limitations of a single model through complementary learning.

2. Related Techniques

2.1. GBDT

The Gradient Boosting Decision Tree (GBDT) is an outstanding machine learning technique renowned for its powerful predictive performance [7]. The essence of this algorithm lies in the progressive construction of a series of decision trees, where each tree aims to correct the prediction errors of its predecessor. This iterative process, utilizing a gradient descent optimization approach, significantly enhances the overall performance of the model.

Given a dataset D={(x_i,y_i):i=1,2,...,n, $x_i \in \boldsymbol{R}$,$y_i \in \boldsymbol{R}$}, where n is the number of samples, each sample has p features. Given the loss function $L(y, f(x))$, the algorithmic steps of GBDT are as follows:

(1) Initialize. $f_0(x) = 0$.

(2) Compute the residuals. $r_m = y_i - f_{m-1}(x_i)$, i=1,2...,n.

(3) Fit a regression tree T($\boldsymbol{x}$, $\boldsymbol{\Theta}_m$) to learn the residuals.

(4) Update. $f_m(\boldsymbol{x}) = f_{m-1}(\boldsymbol{x}) + T(\boldsymbol{x}, \boldsymbol{\Theta}_m)$.

(5) Repeat the iterative steps (2) to (4), minimizing the prediction error to obtain a GBDT model.

2.2. LightGBM

Light Gradient Boosting Machine (LightGBM), a variant of GBDT developed by Microsoft[8], is designed to provide efficient and scalable solutions for large-scale machine learning tasks. Introduced to overcome challenges associated with processing massive datasets, LightGBM employs a distributed computing approach and histogram-based learning, enabling faster and more effective model training. Its innovative features contribute to improved performance in various applications, making it a prominent tool in the realm of machine learning research and applications.

2.2.1. Histogram-based Decision Tree Algorithm

LightGBM innovatively addresses the computational inefficiencies of traditional GBDT algorithms like Extreme Gradient Boosting (XGBoost)[9], which rely on numerical pre-sorting for node splitting. Instead, LightGBM employs a histogram-based approach, discretizing continuous data into bins during preprocessing, reducing time complexity and memory usage. This strategic use of histograms enhances the efficiency and scalability of LightGBM, making it a powerful choice for large-scale machine learning tasks.

2.2.2. The Leaf-wise Leaf Growth Strategy with Depth Constraints

LightGBM adopts a leaf-wise leaf growth strategy with depth constraints, which is a distinctive feature in its algorithmic design. Unlike traditional level-wise growth, the leaf-wise strategy expands the tree by growing the leaves with the highest information gain, allowing for a more efficient and adaptive approach. The depth limitation ensures controlled tree complexity, striking a balance between model accuracy and computational efficiency. This innovative combination contributes to LightGBM's ability to handle large-scale datasets while maintaining high predictive performance.

2.3. CatBoost

Category Boosting (CatBoost), an algorithm developed by the Russian search giant Yandex in April 2017, represents an innovative approach to gradient boosting[10]. As a gradient boosting algorithm, CatBoost demonstrates unique advantages in handling classification and regression problems, particularly excelling in the efficient treatment of categorical features without the need for extensive preprocessing.

2.3.1. Ordered Target Encoding

In CatBoost, the utilization of ordered target encoding represents a refined approach to encoding categorical features, akin to mean encoding but with a distinctive focus on reducing overfitting. CatBoost transforms all categorical feature values into numerical results using the following formula:

$$\text{CatBoost Encoding} = \frac{\text{OptionCount+prior}}{\text{TotalCount}+1} \tag{1}$$

CatBoost traverses each sample in sequence from the beginning to the end. In the formula, "OptionCount" represents the count of samples with a target value of 1 for the category to which the current sample belongs. The variable "prior" denotes the initial value of the numerator, determined based on initial parameters. "TotalCount" represents the count of samples, including the current sample, that share the same categorical feature value across all samples.

2.3.2. Symmetric Decision Tree

Unlike traditional decision trees, which tend to grow asymmetrically, the Symmetric Decision Tree algorithm maintains a balanced structure during its growth process. Each node in the tree attempts to split the data in a manner that maintains symmetry among its child nodes. This symmetry is achieved by considering both left and right child nodes equally, resulting in a tree structure that is more uniform and less prone to overfitting.

3. Oil Temperature Forecasting Based on Multi-model by Stacking Ensemble Learning

Stacking ensemble learning is a powerful paradigm in machine learning that involves combining the predictions of multiple models to achieve better overall performance than any individual model[11]. In the Stacking ensemble learning model, evaluating individual base learners and collectively assessing the combined impact of diverse base learners are essential steps to optimize predictive performance.

In terms of the predictive capabilities of base learners, this paper, in the first layer of the Stacking model, employs LightGBM and CatBoost algorithms as base learners. This contributes to an overall improvement in the predictive performance of the model. To prevent overfitting, this study employs the Random Forest algorithm as the meta-learner in the second layer to rectify biases.

In consideration of the foregoing, the model processing flowchart for oil temperature forecasting model is depicted in Figure 1. Feature Engineering refers to the process in machine learning and data analysis where operations such as creation, transformation, selection, and extraction are applied to raw data features to enhance model performance and data representation. Therefore, after analysing the dataset, statistical features are derived from the numerical columns, serving as a part of the input features for the model.

Obviously, the training set for the meta-learner is generated from the outputs of the base learners. However, directly using the training set of the base learners to retrain the meta-learner can lead to severe overfitting when there is a significant mismatch between the distributions of the training set and test set. To prevent the repetition of learning in a two-layered model, this paper employs K-Fold cross-validation during the training of the base learners. In each iteration, the data is divided into K subsets based on the temporal dimension, with one subset used as the validation set and the remaining subsets used as the training set to train the base learners. This iterative process, performed K times, results in outputs with stronger generalization.

LightGBM and CatBoost, as base learners, can each produce a prediction result for their respective test datasets. These results can be consolidated into a new dataset, which is of the same size as the original dataset, thus achieving feature transformation from input features to output features for all data. Finally, the combined data is input into the meta-learner for training, resulting in the prediction of transformer oil temperature. The selection of a Random Forest model as the meta-learner stems from the imperative to enhance the generalization capabilities of the model.

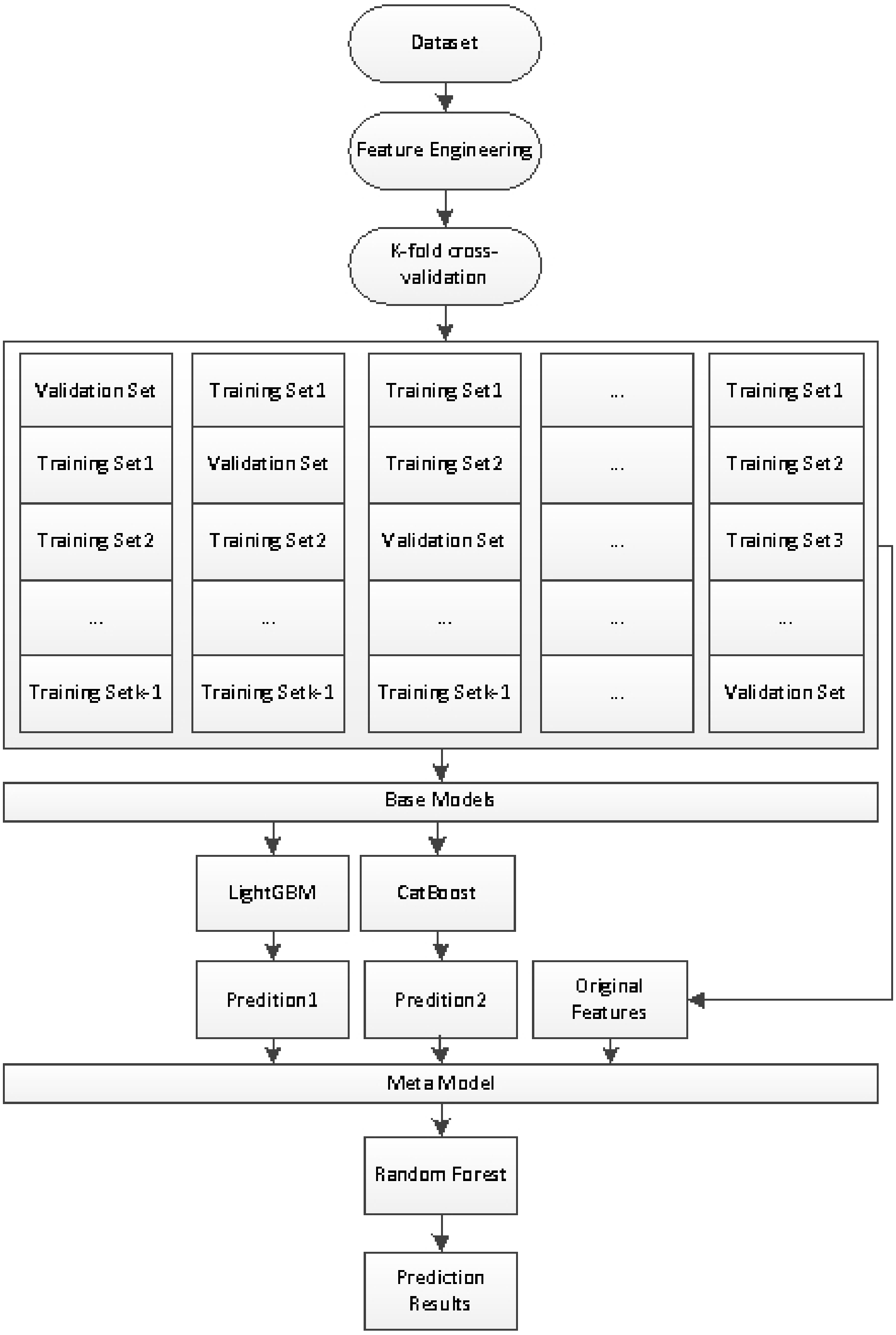

Figure 1 Flowchart for oil temperature forecasting

4. Experiments and Result Analysis

4.1. Experimental Environment

The experimental environment was deployed on the CentOS7 operating system, utilizing an Intel E5-2686 v4 CPU with 8GB of RAM for the training process. The model code was implemented in Python 3.8.

4.2. Experimental Dataset

The experimental dataset comprises 1400 sets of power transformer data from a specific region in China. Each dataset represents the records of a single transformer over the course of one day, with data recorded at half-hour intervals, resulting in a total of 48 data points per set. The specific field explanations can be found in the field descriptions provided in Table 1.

Table 1 Field descriptions

Field Name	Type	Description
Transformers	String	Transformer ID
Time	String	Timestamp, formatted as: Hour: Minute.
L1	Float	The external load value of the transformer 1.
L2	Float	The external load value of the transformer 2.
L3	Float	The external load value of the transformer 3.
L4	Float	The external load value of the transformer 4.
L5	Float	The external load value of the transformer 5.
L6	Float	The external load value of the transformer 6.
oil_temperature	Float	Oil temperature

4.3. Evaluation Metrics

Expanding on the initial results, the evaluation metrics encompassed Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2). MAE quantifies the average absolute differences between predicted and actual values, providing a measure of the model's overall accuracy. MSE extends this evaluation by considering the squared differences, emphasizing the impact of larger errors. R-squared (R^2) complements these metrics, offering an indication of the proportion of variance in the dependent variable explained by the model. Together, these metrics provide a comprehensive assessment of the model's predictive performance, addressing accuracy, precision, and explanatory power. Their calculation formulas are expressed as follows, n is the number of data points, y_i represents the actual values, $\hat{y}_i$ represents the predicted values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{2}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2} \tag{4}$$

4.4. Experimental Results

To validate the effectiveness of the proposed transformer oil temperature prediction model in this paper, the following control experiments were conducted. All model algorithms were based on the physical environment described in Section 4.1 and the dataset outlined in Section 4.2. The experimental results are presented in Table 2.

Table 2 The experimental results

Model	MAE	MSE	R^2
Random Forest	1.97	7.82	0.95
LightGBM	2.04	7.86	0.95
CatBoost	2.08	8.08	0.95
Multilayer Perceptron(MLP)	3.23	18.05	0.88
Stacking Model	1.04	2.66	0.98

Based on the experimental results presented above, it is evident that the Stacking Model achieved the minimum MAE and MSE among all compared models. Furthermore, the corresponding R^2 reached the maximum value, indicating the effectiveness of the proposed model in addressing the transformer oil temperature prediction task.

5. Conclusion

This paper draws inspiration from cutting-edge algorithmic techniques in the field of artificial intelligence and machine learning. In the Stacking ensemble algorithm model, a comprehensive utilization of LightGBM, CatBoost, and Random Forest algorithms is employed to observe the data space and structure from different perspectives. This enables each algorithm to complement the strengths and weaknesses of others. Through empirical tests, it is demonstrated that the proposed Stacking model achieves optimal predictive results for transformer oil temperature prediction.

In future work, a more in-depth exploration will be conducted to address the following issues. During the training of the Stacking model, the extended computational time of base models poses a challenge in integrating a larger number of models. Therefore, it is imperative for future research to deploy distributed computing environments, effectively reducing algorithmic time complexity. This will involve designing larger-scale Stacking models for improved transformer oil temperature prediction.

References

[1] Zhong, J.-L., & Huang, J.-J. (2019). Review of Measurement and Calculation Methods for Hot Spot Temperature of Power Transformers. Electrical Switchgear, 3.

[2] IEEE. (2012). IEEE Guide for Loading Mineral-Oil-Immersed Transformers and Step-Voltage Regulators. IEEE Std C57.91-2011 (Revision of IEEE Std C57.91-1995), 1-123. DOI: 10.1109/IEEESTD.2012.6166928.

[3] Jing, X., Shuxin, L., Zhengwen, L. I., Chao, W., & Junci, T. (2022, March). Modeling and Simulation Analysis of Transformer Hot Spot Temperature Based on Multi Physical Field Coupling Calculation and Temperature Rise Characteristics. In 2022 4th Asia Energy and Electrical Engineering Symposium (AEEES) (pp. 465-471). IEEE.

[4] Jia, X., Lin, M., Su, S., Wang, Q., & Yang, J. (2022). Numerical study on temperature rise and mechanical properties of winding in oil-immersed transformer. Energy, 239, 121788.

[5] Alerskans, E., Nyborg, J., Birk, M., & Kaas, E. (2022). A transformer neural network for predicting near-surface temperature. Meteorological Applications, 29(5), e2098.

[6] Li, P., Zhang, S., Zhu, M., & Li, Y. (2022, August). Hot spot prediction based on SVM and multi-physical field coupling. In 4th International Conference on Information Science, Electrical, and Automation Engineering (ISEAE 2022) (Vol. 12257, pp. 290-302). SPIE.

[7] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[8] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[9] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.

[10] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems, 31.

[11] Ribeiro, M. H. D. M., & dos Santos Coelho, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. Applied soft computing, 86, 105837.