

Article

An Improved Robust Thermal Error Prediction Approach for CNC Machine Tools

Honghan Ye¹, Xinyuan Wei^{2,*} , Xindong Zhuang³ and Enming Miao⁴

¹ Department of Statistics, School of Computer, Data & Information Sciences, College of Letters & Science, University of Wisconsin-Madison, Madison, WI 53705, USA; hye42@wisc.edu

² School of Electrical and Information Engineering, Anhui University of Technology, Ma'anshan 230009, China

³ Hangzhou Hikauto Technology Co., Ltd., Hangzhou 310000, China; zhuangxindong@hikauto.com

⁴ School of Mechanical Engineering, Chongqing University of Technology, Chongqing 400054, China; miaom@163.com

* Correspondence: weixy@ahut.edu.cn; Tel.: +86-152-5295-7376

Abstract: Thermal errors significantly affect the accurate performance of computer numerical control (CNC) machine tools. In this paper, an improved robust thermal error prediction approach is proposed for CNC machine tools based on the adaptive Least Absolute Shrinkage and Selection Operator (LASSO) and eXtreme Gradient Boosting (XGBoost) algorithms. Specifically, the adaptive LASSO method enjoys the oracle property of selecting temperature-sensitive variables. After the temperature-sensitive variable selection, the XGBoost algorithm is further adopted to model and predict thermal errors. Since the XGBoost algorithm is decision tree based, it has natural advantages to address the multicollinearity and provide interpretable results. Furthermore, based on the experimental data from the Vcenter-55 type 3-axis vertical machining center, the proposed algorithm is compared with benchmark methods to demonstrate its superior performance on prediction accuracy with 7.05 μm (over 14.5% improvement), robustness with 5.61 μm (over 12.9% improvement), worst-case scenario predictions with 16.49 μm (over 25.0% improvement), and percentage errors with 13.33% (over 10.7% improvement). Finally, the real-world applicability of the proposed model is verified through thermal error compensation experiments.

Keywords: adaptive LASSO; CNC machine tools; thermal errors; robustness; variable selection; XGBoost



Citation: Ye, H.; Wei, X.; Zhuang, X.; Miao, E. An Improved Robust Thermal Error Prediction Approach for CNC Machine Tools. *Machines* **2022**, *10*, 624. <https://doi.org/10.3390/machines10080624>

Academic Editors: Fang Cheng, Qian Wang, Tegoeh Tjahjowidodo and Ziran Chen

Received: 4 July 2022

Accepted: 27 July 2022

Published: 29 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to changes in heat sources internally and externally in a machining process, thermal deformation of machine tools occurs and thus changes the relative position between the tool and workpiece, which is known as thermal errors or thermally induced errors [1,2]. Thermal errors have become one of the most important factors affecting the accuracy of computer numerical control (CNC) machine tools, which account for up to 75% of the overall geometrical errors of machined workpieces [3]. Therefore, it is important and imperative to reduce thermal errors to improve the accuracy of CNC machine tools.

To reduce thermal errors, in general, there are two main research directions. The first direction is the numerical analysis, which establishes the analytical model and then simulates and analyzes thermal error law. For example, Creighton et al. [4] proposed a thermal error compensation model using the finite element analysis in a high-speed micro-milling spindle. Xu et al. [5] established thermal behavior models using the finite element method (FEM) for an air-cooling ball screw system to predict and compensate for thermal errors. Li et al. [6] proposed an explicit analytical thermal error model for compensation considering ambient temperature fluctuations and the model was verified by both FEM and an experiment on the machine tool. Xaver et al. [7] proposed a structural model using FEM for the ball screw axes of the machine such that up to 87% of the maximal thermo-elastic

error is reduced and compensated. Naumann et al. [8] compared different basis functions based on regression analysis for thermal error compensation using FEM simulation data from a machine tool demonstrator. More recently, Naumann and Herzog [9] considered a coupled thermo-elastic FEM of a simplified machine tool for optimal sensor placement. Świć et al. [10] proposed a thermo-mechanical method based on the thermal deformation mechanism to improve the accuracy and stability of long low-rigidity shafts. While the numerical analysis provides promising compensation results, in practice, it is extremely difficult for the numerical method to build an exact structural model and simulate the thermal deformation of machine tools due to complex deformation processes.

Alternatively, the second direction, based on statistical prediction models, has attracted increasing attention to compensate for thermal errors because it is cost-effective and easy to use. To establish a statistical prediction model, sensors are first installed at various locations of CNC machines to measure temperature changes, which are considered input variables. Then, thermal errors become the output variable of the model. By measuring the relationship between thermal errors and temperature measurements through statistical models, thermal errors are predicted in real time based on the observed temperature measurements. Therefore, thermal errors are compensated, and the accuracy of CNC machine tools is significantly improved. Along this line of research, enormous efforts have been made to establish statistical prediction models [11], such as multiple linear regression (MLR), support vector machine (SVM) methods, and neural networks. In addition, it is a common practice to select temperature-sensitive variables to alleviate the strong multicollinearity among input variables [12,13]. For example, Yang et al. [14] proposed to group temperature variables according to the correlation among the input variables to reduce multicollinearity. Further, Yan and Yang [15] proposed to first group temperature variables according to the integrated gray correlation between temperature variables and thermal errors, and then pick one temperature variable from each group that has the maximum integrated gray correlation as the selected temperature-sensitive variables. Abdulshahed et al. [16] proposed a combination of fuzzy c-means clustering and the gray correlation methods to select temperature-sensitive variables with an adaptive neuro-fuzzy inference system to establish a thermal error prediction model. A similar variable selection method has been utilized in recent literature [17,18]. For example, Zhang et al. [19] combined fuzzy clustering with correlation coefficient methods to select temperature-sensitive variables and used the sliced inverse regression method to establish a thermal error model. Miao et al. [20] proposed the principal component regression method to reduce the influence of the variability of temperature-sensitive variables on the model robustness. Further, Liu et al. [21] proposed a thermal error model based on the ridge regression algorithm to alleviate multicollinearity, where they used the correlation-coefficient method to select temperature-sensitive variables. Tan et al. [22] utilized the least absolute shrinkage and selection operator (LASSO) method to select temperature-sensitive variables and established the least-square SVM-based thermal error model. Specifically, the LASSO method is adopted to select temperature-sensitive variables by penalizing all coefficients in the regression model with the same penalty term call L1-norm. While the LASSO method is simple to implement in practice, it does not have the oracle property [23]. As a side note, a variable selection procedure with the oracle property means that the procedure can identify the right subset of true variables. Here, the LASSO procedure does not have the oracle property, which means that the selected variables by the LASSO method are not consistent with the underlying true variables. To address this issue, by adding different weights to different coefficients, the adaptive LASSO method [24] has been theoretically proved to enjoy the oracle property with the consistent variable selection. In addition, this is the first work that the adaptive LASSO method has been adopted in the field of thermal error modeling.

Due to the rapid development of machine learning techniques, more advanced thermal error models have been proposed in the literature [25–28]. For example, Liang et al. [29] proposed a thermal error prediction model for heavy-duty CNC machines with long short-

term memory networks. More recently, Liu et al. [30] utilized the long short-term memory to compensate for thermal errors in a spindle system. Unfortunately, these models require a large amount of data to train for adequate prediction accuracy and more severely function as a black box and fail to provide interpretable results. To address this issue, Zhu et al. [24] proposed a thermal error model based on the random forest (RF) algorithm. Although the RF model requires less training data and provides interpretable results, it has a severe limitation that a small change in the hyperparameter will affect almost all trees in the forest and then affect the performance of prediction accuracy. With pre-trained hyperparameters for the whole forest, the RF approach degrades significantly and becomes less robust when test data have more variations than training data. In summary, the existing literature still lacks a more robust thermal error prediction model that has the oracle property of temperature-sensitive variable selection and provides interpretable results. To fill this research gap, in this paper, an improved robust thermal error prediction model is proposed based on the adaptive LASSO and eXtreme Gradient Boosting (XGBoost) algorithms, and the proposed Adaptive LASSO Integrated with Xgboost algorithm is abbreviated as the ALIX method. In particular, the adaptive LASSO method, which enjoys the oracle property by penalizing coefficients of variables with different weights, is first used to select the temperature-sensitive variables, and then the XGBoost algorithm is built based on decision trees is adopted to model thermal errors for CNC machines. As a result, the proposed ALIX method for thermal error modeling and prediction has several unique advantages:

- (1) The adaptive LASSO method enjoys the oracle property for selecting temperature-sensitive variables consistently; namely, it performs as well as if the true underlying model is given in advance. In addition, the adaptive LASSO method can be solved efficiently and achieves superior performance on variable selection in various applications.
- (2) The XGBoost algorithm is robust to multicollinearity in nature while the multicollinearity is commonly seen in thermal error modeling of CNC machines, and the embedded regularization in the XGBoost algorithm helps avoid over-fitting. In addition, different from existing neural network models which function as a black box and fail to interpret, the XGBoost method can provide desirable interpretable results and identify which variables have the most effects on thermal errors.
- (3) Both the adaptive LASSO and XGBoost algorithms are first-ever adopted in the literature to predict thermal errors for CNC machines. Our proposed method contributes to the practice of precision engineering by illustrating how practitioners can utilize the proposed method for accurate and robust thermal error predictions. Based on our experimental data from the Vcenter-55 type 3-axis vertical machining center, compared with several benchmark methods, the proposed ALIX algorithm demonstrates its superior performance in prediction accuracy, robustness, and worst-case scenario prediction.

The remainder of this paper is organized as follows. In Section 2, the thermal error experiment on the Vcenter-55 type 3-axis vertical machining center is introduced. Section 3 provides the technical details of the ALIX algorithm. In Section 4, the ALIX algorithm is compared with benchmark methods based on the experimental data. Section 5 conducts the experimental verification to demonstrate the real-world applicability of the proposed ALIX algorithm on thermal error compensation. Finally, Section 6 draws conclusions and discusses future research.

2. Thermal Error Experiment

A total of 23 batches of thermal error measurement experiments with different ambient temperatures and spindle speeds were conducted. Experimental data with a wide range of ambient temperatures were used to investigate the prediction accuracy and robustness of thermal error models. In the following, the experimental object is introduced in Section 2.1 and the exploratory data analysis of the experimental data is provided in Section 2.2.

2.1. Experiment Object

The experimental object was a Vcenter-55 type 3-axis vertical machining center, as shown in Figure 1. The five-point measurement method was used to measure thermal errors with reference to ISO 230-3: 2020 [31]. Five displacement sensors were installed on the experimental object with two sensors in each of the X and Y directions and one sensor in the Z direction. The displacement sensor is a MicroSense 5810 (MicroSense, LLC, Lowell, MA, USA) type capacitive sensor with a measurement accuracy of 1 μm after calibration. In addition, twenty temperature sensors, referred to as T1–T20, were installed to measure temperature changes at specific locations of the experimental object. The temperature sensors are PT100 platinum resistors with an accuracy of 0.1 degrees Celsius after calibration.

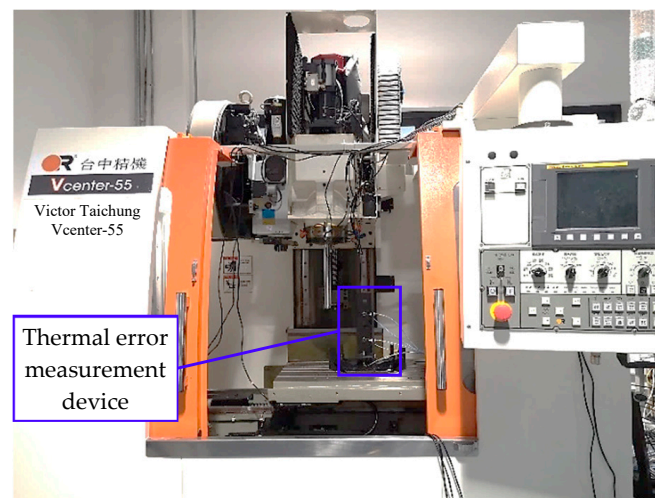


Figure 1. Experimental object.

The layout of the displacement and temperature sensors in the experimental object is shown in Figure 2, and the detailed locations of T1–T20 are listed in Table 1. It is noteworthy that T10 and T20 are not displayed in Figure 2 as T10 is placed on the machine housing to measure the ambient temperature and the installment location of T20 is blocked in Figure 2.

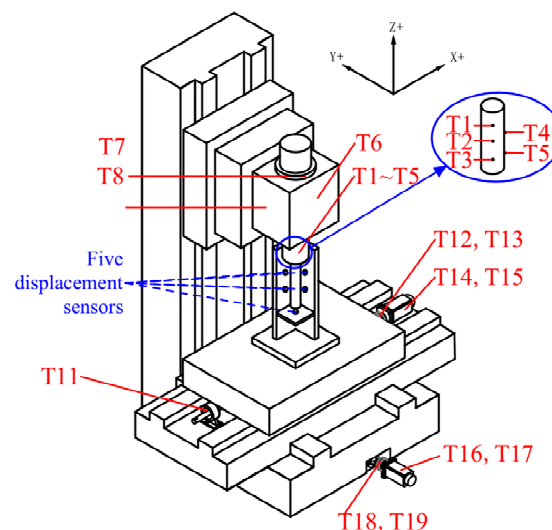


Figure 2. Sensor layout in the experimental object.

Table 1. Detailed installment location of sensors.

Temperature Sensors	Installment Location
T1–T5	Front bearing of the spindle
T6, T9	Spindle box
T7, T8	Spindle motor
T10	Machine housing
T11	Support base on X direction
T12, T13	Screw nut on X direction
T14, T15	Motor on X direction
T16, T17	Motor on Y direction
T18, T19	Screw nut on Y direction
T20	Support base on Y direction

2.2. Exploratory Data Analysis

A total of 23 batches of experiments were conducted in total, and the experimental batches are arranged in ascending order of the initial environment temperature and recorded as K1–K23. Note that K1–K23 is the notation for the 23 batches. Each batch of experiments was conducted under two varying parameters: ambient temperatures and spindle speeds. Furthermore, the data for each batch include both temperature signals and the measured thermal errors. The experimental conditions, including both spindle speed and initial ambient temperature, for each batch of experiments are tabulated in Table 2. From Table 2, two spindle speeds are considered, and initial ambient temperature, which is defined as the ambient temperature naturally measured by T10 at the beginning of each experiment, varies significantly as experiments were performed at different seasons over a year. In each batch of the experiments, the machine spindle was idling, and the spindle speed was constant. The worktable was run back and forth along the X and Y axes at a constant feed rate of $1500 \text{ mm} \cdot \text{min}^{-1}$ with reference to ISO 230-3: 2020 [26]. The temperature and thermal errors were collected every 5 min, and each batch of the experiment lasted more than 6 h.

Table 2. Experimental conditions for each batch of experiments.

Batch	Spindle Speed (rpm)	Initial Ambient Temperature (Degree Celsius)	Batch	Spindle Speed (rpm)	Initial Ambient Temperature (Degree Celsius)
K1	4000	4.38	K13	6000	10.88
K2	4000	4.50	K14	4000	12.94
K3	4000	5.31	K15	4000	14.44
K4	6000	5.75	K16	6000	14.63
K5	6000	6.19	K17	6000	21.69
K6	4000	6.69	K18	6000	24.50
K7	6000	7.06	K19	4000	25.06
K8	4000	9.19	K20	6000	25.63
K9	4000	9.25	K21	6000	25.69
K10	4000	9.63	K22	6000	27.75
K11	6000	9.81	K23	6000	33.13
K12	6000	10.50			

Based on the experimental data, the following exploratory data analysis is conducted. First, the initial environment temperature range is 4.38–33.13 degrees Celsius according to T10 of K1–K23. Figure 3 further shows the temperature curves of T1–T20 for the K1 (lowest initial environment temperature) and K23 (highest initial environment temperature). Based on Figure 3, for both K1 and K23, some sensors have linear relationships over time whereas other sensors have different slopes at different measurement times. Comparing the curves between K1 and K23, the curves in K1 are much steeper than those in K23 when the machines start, which is expected due to the warm-up with the low environmental temperature. It is noteworthy that since the ambient temperature is measured by T10, the

plot of T10 in Figure 3 shows the change of ambient temperature over time for both K1 and K23. In addition, the initial ambient temperature is revealed by the starting point of the T10 line when the measurement time is 0 in Figure 3.

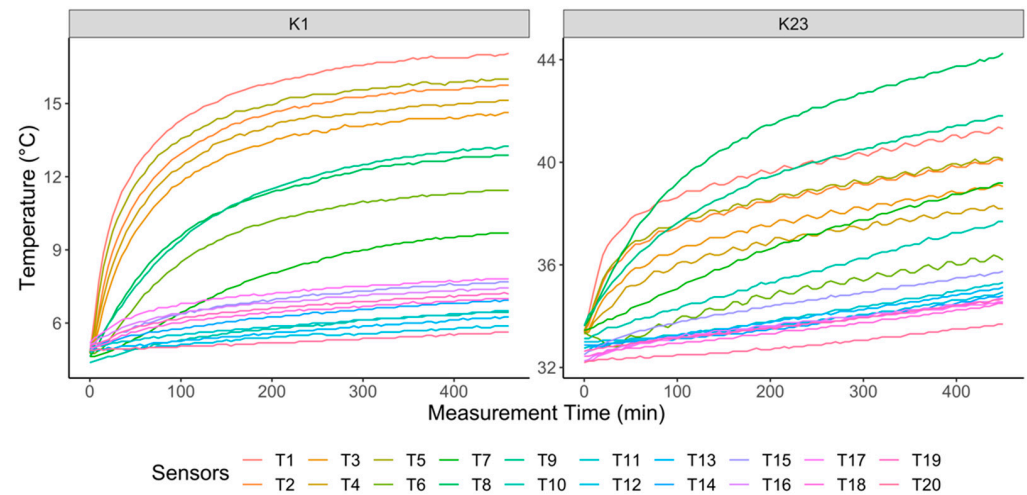


Figure 3. Temperature curves over measurement time on K1 and K23.

Second, as thermal errors caused by the axial thermal expansion of the machine tool spindle account for the main part of the total thermal errors [32], thermal errors in the Z direction are the focus of this study. Figure 4 shows the thermal error curves in the Z direction of all 23 batches. It can be observed that the thermal errors rise quickly from the beginning to around the 100th minute and then the curves become flat after 100 min. Next, the proposed methodology is presented to first select temperature-sensitive variables and then model the relationship between thermal errors and temperature sensors.

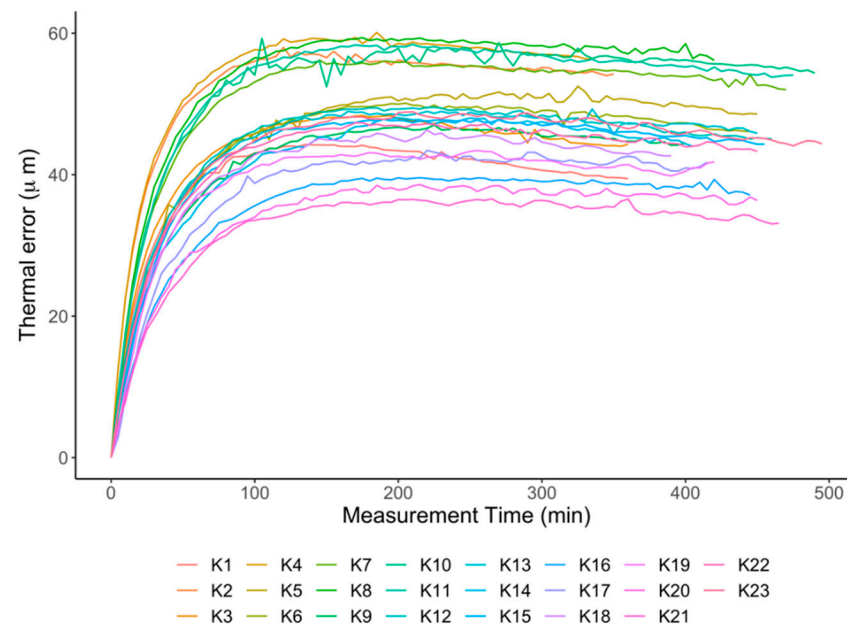


Figure 4. Thermal error curves in the Z direction of all 23 batches.

3. Methodology

As mentioned before, the existing literature lacks a robust thermal error model that enjoys the oracle property of the variable selection and provides desirable interpretable results. To fill this research gap, the ALIX algorithm is the first-ever proposed in the field of

thermal error modeling and prediction by combining the adaptive LASSO method with the XGBoost algorithm. A flowchart of the proposed methodology is presented in Figure 5 to systematically illustrate our ALIX algorithm. Specifically, during the modeling stage, the adaptive LASSO method [28] is first used, which will be presented in Section 3.1, to select temperature-sensitive variables among all installed sensors. Then, the XGBoost algorithm [29] using the selected temperature-sensitive variables is adopted for thermal error modeling, which will be provided in Section 3.2. During the real-time compensation stage, the established ALIX method is deployed to predict and compensate for the thermal error based on the selected temperature signals.

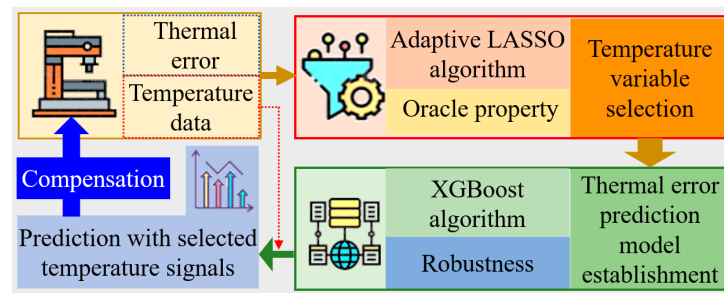


Figure 5. Flowchart of the proposed ALIX methodology.

3.1. Temperature-Sensitive Variable Selection

Suppose that $\mathbf{y} = (y_1, \dots, y_n)^T$ is the response vector of thermal errors, and $\mathbf{x}_j = (x_{1,j}, \dots, x_{n,j})^T$ are temperature measurements from sensor j , $j = 1, \dots, p$, after deducting from the initial environment temperature. Note that n is the number of observations and p is the number of temperature sensors in total. Without loss of generality, the data are assumed to be centered, so the intercept is not included in the model. Assume that $\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\varepsilon}$, where $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_p]$ is the predictor matrix, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^T$ is the true parameter of the model, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^T$ are independent identically distributed random variables with mean 0 and variance σ^2 . Further, denote that $\hat{\boldsymbol{\theta}}(\delta) = (\hat{\theta}_1, \dots, \hat{\theta}_p)^T$ are the coefficient estimators produced by a fitting method δ . For example, when δ is ordinary least square (OLS), the objective function is to minimize the least squared error, namely $\arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2$, and the unbiased estimators $\hat{\boldsymbol{\theta}}(OLS)$ can be easily obtained in (1):

$$\hat{\boldsymbol{\theta}}(OLS) = \mathbf{X}^T \mathbf{X}^{-1} \mathbf{X}^T \mathbf{y} \quad (1)$$

While the OLS is easy to implement, it gives non-zero values to all coefficients. However, in practice, not all variables are significant, especially when there is strong collinearity among temperature variables in the thermal error modeling of CNC machines. Such strong collinearity would severely deteriorate the model prediction accuracy and robustness [16]. Thus, it is commonly used to select temperature-sensitive variables before establishing a thermal error model.

As reviewed in Section 1, many existing methods of selecting temperature-sensitive variables are heuristic and they cannot guarantee consistent selection. To address this issue, the adaptive LASSO method has been proposed by Zou [33] that enjoys the oracle property and guarantees the optimal variable selection. More importantly, the adaptive LASSO method has shown superior performance on variable selection in various applications. Therefore, the adaptive LASSO method is adopted here to select temperature-sensitive variables for our thermal error modeling and predictions. The coefficient estimators of the adaptive LASSO method, denoted as $\hat{\boldsymbol{\theta}}(aLASSO)$, can be obtained by optimizing the following objective function:

$$\hat{\boldsymbol{\theta}}(aLASSO) = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 + \lambda \sum_{j=1}^p \hat{w}_j |\theta_j| \quad (2)$$

As shown in (2), unlike the LASSO method that selects variables by penalizing all coefficients with the same penalty term [22], the adaptive LASSO method performs different penalizations for each coefficient through \hat{w}_j . As a result, if a variable is important, \hat{w}_j is small and the variable remains in the model. In contrast, if a variable is not important, \hat{w}_j is large and the variable is more likely to be eliminated. In theory, there are multiple ways to construct \hat{w}_j [33]. Here, we choose the following commonly used method to obtain \hat{w}_j . First, the traditional LASSO model is built by solving the following equation:

$$\hat{\theta}(\text{LASSO}) = \underset{\theta}{\operatorname{argmin}} \|y - X\theta\|^2 + \lambda \sum_{j=1}^p |\theta_j| \quad (3)$$

To determine the value of λ in (3), 10-fold cross-validation is used and the loss function is mean squared error (MSE), which is defined as $(y - X\hat{\theta})^2$. To solve (3), the *glmnet* function can be used in the R language [34], which adopts the cyclical coordinate descent algorithm [35]. Specifically, the cyclical coordinate descent algorithm successively optimizes the objective function over each parameter with other parameters fixed, and cycles repeatedly until the algorithm converges. Since the coordinate descent method is widely used and embedded in the *glmnet* function, we do not present details here. Then, \hat{w}_j can be further obtained by the following equation:

$$\hat{w}_j = \frac{1}{|\hat{\theta}_j(\text{LASSO})|} \quad (4)$$

Plugging (4) into (2), the adaptive LASSO model can be written as follows:

$$\hat{\theta}(\text{aLASSO}) = \underset{\theta}{\operatorname{argmin}} \|y - X\theta\|^2 + \lambda \sum_{j=1}^p \frac{|\theta_j|}{|\hat{\theta}_j(\text{LASSO})|} \quad (5)$$

Similarly, λ in (5) can be determined using 10-fold cross-validation and the loss function is also set as MSE. After obtaining $\hat{\theta}(\text{aLASSO})$ using the same *glmnet* function with different parameters, the set of temperature-sensitive variables $\mathcal{O}(TS)$ can be selected as follows:

$$\mathcal{O}(TS) = \{j : \hat{\theta}_j(\text{aLASSO}) \neq 0 \text{ where } j = 1, \dots, p\} \quad (6)$$

It is noteworthy that in this article, we only use one batch of data as training data to fit the model, and then use the trained model to predict the remaining batches of data. Therefore, for each batch as training data, different $\mathcal{O}(TS)$ are generated via the adaptive LASSO method.

3.2. Thermal Error Modeling

With the temperature-sensitive variables selected by the adaptive LASSO method, the XGBoost algorithm is further used to fit the data. The XGBoost algorithm is a scalable tree boosting system [36]. Different from the existing algorithms which assume the linear relationship between thermal errors and temperature from sensors, the decision-tree-based XGBoost algorithm is nonparametric for regression and is robust to multicollinearity by nature. Furthermore, compared with the RF method that builds decision trees independently and heavily depends on hyperparameters to optimize the model, the XGBoost algorithm originates from the gradient boosting model, which combines many weak learners into a stronger learner in an iterative fashion [37]. In this way, the XGBoost method only applies hyperparameters to one tree at the beginning. Mathematically, the XGBoost algorithm predicts thermal errors, i.e., \hat{y}_i , using the input variables $x_{i,j}$, where $i = 1, \dots, n$, $j \in \mathcal{O}(TS)$ and \mathcal{L} additive functions:

$$\hat{y}_i = \sum_{\ell=1}^{\mathcal{L}} f_{\ell}(x_i), f_{\ell} \in \mathcal{F}, \quad (7)$$

where $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,|\mathcal{O}(TS)|})$ and $|\mathcal{O}(TS)|$ is the cardinality of the $\mathcal{O}(TS)$. Here, f_ℓ represents an independent decision tree structure v with leaf scores ω , and \mathcal{F} is the space of trees. In practice, it is impossible to enumerate all the possible tree structures. Instead, a greedy algorithm is used that starts from a single leaf and iteratively adds branches to the tree. The objective function J at t time iteration, as shown in (8), consists of both the training loss function and the regularization term:

$$J(t) = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{\ell=1}^t \Omega(f_\ell) \quad (8)$$

where L is a differentiable convex loss function, such as square loss, which measures how well the model fits on the training data. In addition, Ω is the regularization term, such as L1 norm or L2 norm, which penalizes the complexity of the model. As a side note, the time iteration t represents the t -th iteration during the training process, rather than the measurement time during the data collection. Then, the predicted thermal errors at time iteration t are obtained based on (9):

$$\hat{y}_i(t) = \sum_{\ell=1}^t f_\ell(\mathbf{x}_i) = \hat{y}_i(t-1) + f_t(\mathbf{x}_i). \quad (9)$$

Furthermore, as defined in [36], Ω for a decision tree f_ℓ can be calculated by the following equation:

$$\Omega(f_\ell) = \gamma T + \frac{1}{2} \lambda \sum_{q=1}^T \omega_q^2 \quad (10)$$

where γ is the complexity of each leaf, T is the number of leaves in the decision tree, λ is a parameter to scale the penalty, and ω_q is the score of the q -th leaf. Then, the second-order Taylor expansion, instead of the first-order in the gradient boosting decision tree, is used to approximate the training loss function in the XGBoost algorithm. In this paper, we adopt the MSE as the loss function, and the objective function can be finalized after removing the constant as below:

$$J(t) \approx \sum_{i=1}^n \left[g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t) \quad (11)$$

where g_i and h_i are the first and second derivatives of the MSE loss function, respectively. Since each data sample \mathbf{x}_i corresponds to only one leaf node, the loss function can be expressed by the sum of loss values of each leaf node. Then, plugging (10) into (11), the objective function can be rewritten as follows:

$$\begin{aligned} J(t) &\approx \sum_{i=1}^n \left[g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{q=1}^T \omega_q^2 \\ &= \sum_{q=1}^T \left[\left(\sum_{i \in I_q} g_i \right) \omega_q + \frac{1}{2} \left(\sum_{i \in I_q} h_i + \lambda \right) \omega_q^2 \right] + \gamma T \end{aligned} \quad (12)$$

where $I_q = \{i | v(\mathbf{x}_i^k) = q\}$ represents all the data samples at leaf q . Therefore, the optimization of the objective function in (12) can be transformed into a problem of finding the minimum of a quadratic function. After a certain node split in the decision tree, the model performance is evaluated based on the objective function. If the model performance is improved, then the split is adopted, otherwise, the split will be stopped. More importantly, since the regularization term $\Omega(f_\ell)$ is added into the objective function, the XGBoost model is able to alleviate the problem of overfitting. Further, as a decision-tree-based algorithm, the XGBoost algorithm is robust to multicollinearity. To implement the XGBoost algorithm, the 'xgbTree' method is adopted in the 'caret' package for the R language [38].

4. Performance Evaluation

In this section, the performance of the proposed ALIX algorithm is evaluated based on the experimental data introduced in Section 2. As mentioned before, each batch of data is used as training data for the temperature-sensitive variable selection and XGBoost thermal error model training. Then, the trained model is evaluated to predict based on the remaining batches of data that are not involved in the model training. In the following, the temperature-sensitive variable selection results by the adaptive LASSO method are presented in Section 4.1, the hyperparameter setting and interpretable results involved in the XGBoost algorithm are detailed in Section 4.2, and the performance comparison of the ALIX method with existing algorithms is provided in Section 4.3.

4.1. Temperature-Sensitive Variable Selection

In the adaptive LASSO method, the parameter λ in (3) and (5) is tuned based on the 10-fold cross-validation. Specifically, a sequence of different values of λ is first generated, and for each value of λ , the 10-fold cross-validation is applied and the MSE can be obtained accordingly. Then, the optimal value of λ is selected with the minimum MSE. For example, the parameter of λ in (5) is tuned based on data from K2. Figure 6 plots the MSE for each value of λ . The optimal value of λ can be found as 0.17567, where the $\log(\lambda)$ is -1.739 , with the minimum MSE of 0.544. Then, based on the optimal value of λ , the coefficients in (5) can be estimated with $\hat{\theta}_1(aLASSO) = 4.72$, $\hat{\theta}_{11}(aLASSO) = -13.16$ and the remaining coefficients are zero. Thus, sensors T1 and T11 are selected as temperature-sensitive variables with training data K2.

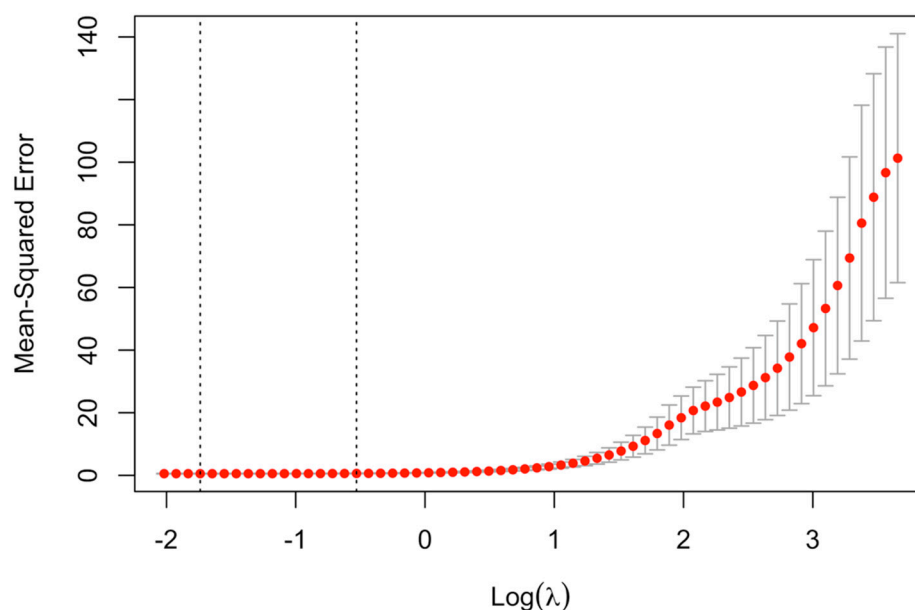


Figure 6. Mean squared errors over parameter λ in (5) with training data K2.

The temperature-sensitive variable selection results based on each batch are summarized in Table 3. Note that Table 3 presents the indices of selected temperature sensors, and the real location of these sensors can be correspondingly found in Figure 2. From Table 3, the selection of temperature-sensitive variables varies based on different batches of training data. For example, for K2, K7 and K8, only two temperature-sensitive variables are selected for XGBoost modeling and predictions. In contrast, for K1 and K19, more than 10 temperature-sensitive variables are selected. In addition, since the adaptive LASSO method is data-driven, the selection of temperature-sensitive variables highly depends on the training data. The large variability of the temperature-sensitive variables for different batches results from the fact that each batch of data was collected under various experimental conditions. It is noteworthy that Table 3 only shows the selection results when each batch is

considered as training data separately. The reason why one single batch of data is used for training is to mimic more challenging situations when only a limited amount of training data are available in real-world applications. If the training data contain multiple batches, then multiple batches of data will be utilized together to select the temperature-sensitive variables for prediction. Next, these selected temperature-sensitive variables are used for thermal error modeling and predictions in the XGBoost algorithm.

Table 3. Selected temperature-sensitive variables for each batch by the adaptive LASSO method.

Batch	Selected Temperature Sensors	Batch	Selected Temperature Sensors
K1	1, 3, 7, 11, 12, 13, 14, 16, 18, 19, 20	K13	1, 2, 3, 7, 13, 14, 20
K2	1, 11	K14	1, 11, 20
K3	1, 11, 14	K15	2, 5, 7, 11, 20
K4	1, 10, 12	K16	2, 3, 5, 7, 11, 16, 20
K5	1, 7, 10, 11, 17, 20	K17	2, 3, 7, 11, 12, 13, 20
K6	1, 10, 20	K18	1, 3, 11, 19
K7	1, 11	K19	2, 6, 7, 8, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20
K8	1, 11	K20	1, 8, 11
K9	1, 3, 7, 10, 11, 17	K21	1, 6, 7, 8, 10, 11, 16, 18, 20
K10	1, 7, 11, 13	K22	1, 3, 11, 12, 20
K11	1, 10, 12, 13, 20	K23	1, 8, 11
K12	5, 12, 13, 14, 20		

4.2. Hyperparameter Setting and Interpretable Results

In the XGBoost algorithm, several hyperparameters need to be tuned to maximize the model performance. The *number of iterations* is the number of trees that are fitted into the model. The *maximum depth* is the maximum number of splits. The *eta* (also known as learning rate) is used to shrink the weights of each step to make the model more robust. The *minimum child weight* defines the minimum sum of weights of the smallest leaf nodes to reduce overfitting. The *subsample* defines the sampling rate of all training samples. The *colsample bytree* is the sampling rate for input variables when constructing each tree. A node is split only when the resultant split improves the loss function. The *gamma* specifies the minimum loss function reduction required to make a split. Since the amount of data in each batch is not large, the *subsample* and *colsample bytree* are set as 1. For tuning the remaining hyperparameters, the details are summarized in Table 4.

Table 4. Detailed hyperparameters configuration.

Hyperparameters	Search Space
<i>number of iterations</i>	{500, 1000}
<i>maximum depth</i>	{4, 6}
<i>eta</i>	{0.01, 0.05}
<i>minimum child weight</i>	{0, 20}
<i>gamma</i>	{0, 50}

The grid search technique, which is one of the most used methods for hyperparameter optimization, is used to find the optimal parameters based on Table 4. Specifically, we search through the manually defined subset, which is Table 4 here, of hyperparameters for the XGBoost model. As a result, the search space is defined, and the goal is to optimize the MSE of the XGBoost method given this search space. During the training process, 10-fold cross-validation is adopted. Taking the batch K23 as training data, for example, the XGBoost model is established based on $\mathcal{O}(TS) = \{1, 8, 11\}$ from Table 3. Figure 7 shows the RMSE of the XGBoost method over iterations with two different values of eta when other hyperparameters are fixed. As expected, with a larger learning rate, the RMSE converges faster. After the grid searching, the optimal parameters are obtained with the minimum MSE: the number of iterations is 500, maximum depth is 4, eta is 0.05, gamma is 0, and

minimum child weight is 0. As expected, the thermal error model based on each batch may differ from each other in terms of the optimal parameters. In other words, when working conditions change or training data change, these selected parameters need to be re-tuned.

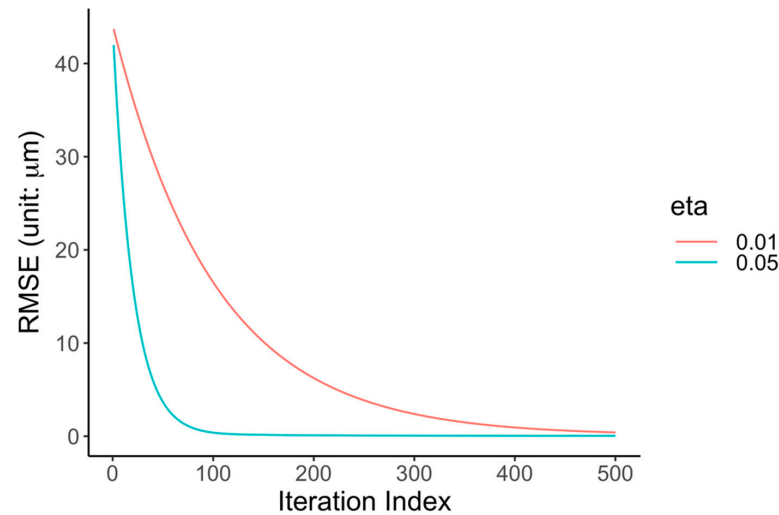


Figure 7. RMSE over iterations with two different values of eta for training data K23.

In addition, different from existing machine learning algorithms that function as a black box, the XGBoost algorithm has the advantage to provide desirable interpretable results by ranking the importance of input variables. Take the batch K5 as training data for example. After selecting temperature-sensitive variables by the adaptive LASSO and training the XGBoost model, the importance of each temperature-sensitive variable can be ranked as follows: $T1 > T7 > T11 > T20 > T10 > T17$. Such interpretable ranking information can help practitioners gain more insights into understanding the relationship between the temperature-sensitive variables and thermal errors, and potentially guide the design of sensor installment in their future experiments.

4.3. Performance Comparison

To evaluate the ALIX method and compare performance with different algorithms, four evaluation metrics are used including prediction accuracy, prediction robustness, worst-case scenario prediction, and percentage errors. The prediction accuracy for a model that is trained based on data from the batch k ($k = 1, \dots, 23$) is denoted as S_k , which is calculated as follows:

$$S_k = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}} \quad (13)$$

where N is the number of all experimental observations except the training data from the batch k , y_i is the i th actual thermal error measurement, and \hat{y}_i is the i th predicted thermal error. Second, the prediction robustness for a model that is trained based on data from the batch k is denoted as R_k , which is calculated as follows:

$$R_k = \sqrt{\frac{\sum_{i=1}^N (r_i - \bar{r}_i)^2}{N - 1}} \quad (14)$$

where $r_i = y_i - \hat{y}_i$ and $\bar{r}_i = \frac{1}{N} \sum_{i=1}^N r_i$. Based on (14), the prediction robustness is the standard deviation of the residuals between the actual and predicted thermal errors [39]. Third, the worst-case scenario prediction for a model that is trained based on data from the batch k is denoted as W_k , which is calculated as follows:

$$W_k = \max_{i=1, \dots, N} |y_i - \hat{y}_i| \quad (15)$$

Based on (15), the worst-case scenario prediction is the maximum absolute deviation between the actual and predicted thermal errors. Last, the percentage error for a model that is trained based on data from the batch k is denoted as P_k , which is calculated as follows:

$$P_k = \frac{\sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i}}{N} \times 100\%. \quad (16)$$

From (16), the percentage error for a model is the average percentage of the predicted thermal errors deviated from the actual thermal errors. Here, we abuse the notation N by excluding cases where $y_i = 0$. Please note that all four evaluation metrics are calculated based on the unseen data, which are not involved in the model training. Thus, the evaluation results are credible in practice.

For the performance comparison, three existing methods are selected including the OLS, the SVM combined with the LASSO (LASSO-SVM), and the RF algorithms. Specifically, for the SVM method, the radial basis function is used as the kernel function and the parameters are tuned based on 10-fold cross-validation. Furthermore, the RF algorithm, which is a decision-tree-based ensemble learning method, is considered to ensure a fair comparison in terms of computational complexity. In the RF algorithm, the number of variables randomly sampled as candidates at each split (ranges from 1 to 20 with the step size of 1), the minimal size of terminal nodes (ranges from 3 to 9 with the step size of 2), and the number of trees (selected from 100, 500, 1000) are tuned based on 10-fold cross-validation.

The comparison results are summarized as follows. First, the results of S_k for each model that is trained on batch k are plotted in Figure 8. From Figure 8, it can be seen that the ALIX method performs better than the other three methods with the highest prediction accuracy (i.e., smallest S_k) in most cases. In particular, the OLS method fluctuates significantly ranging from less than 5 μm to over 20 μm . The LASSO-SVM method also has large variations over different batches and has the highest S_k for batches 19–21, compared with the other three methods. On the contrary, both RF and ALIX methods, which are decision-tree-based methods, are more stable as they are more likely to avoid overfitting. More importantly, the ALIX method consistently performs better than the RF method, although S_k of the ALIX method is slightly higher than that of the RF method in batches 1 and 3.

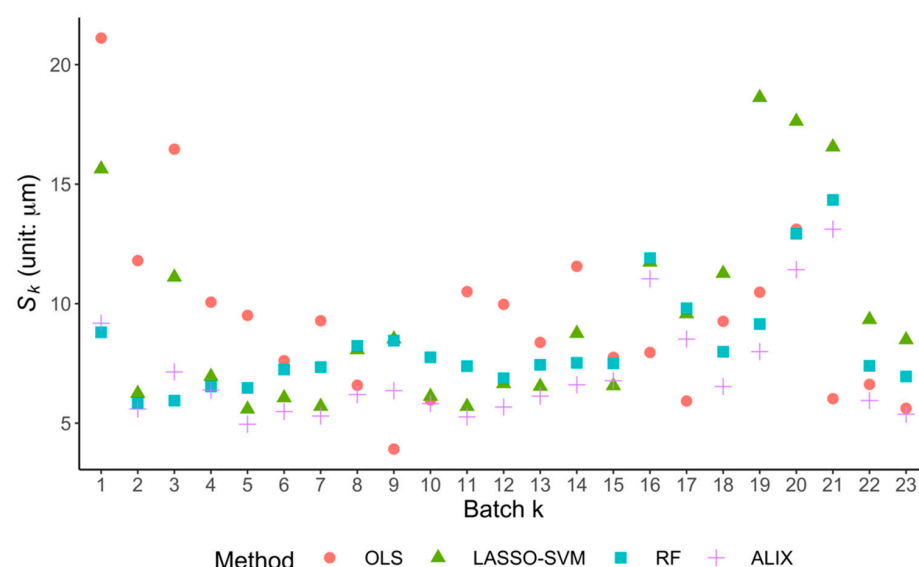


Figure 8. Results of S_k for each model that is trained on batch k .

Next, the results of robustness R_k for each model that is trained on batch k are plotted in Figure 9. From Figure 9, similar observations can be drawn. Both OLS and LASSO-SVM methods have large variations whereas the RF and ALIX methods are more stable. Overall, the ALIX method is more robust than the other three compared methods with smaller R_k .

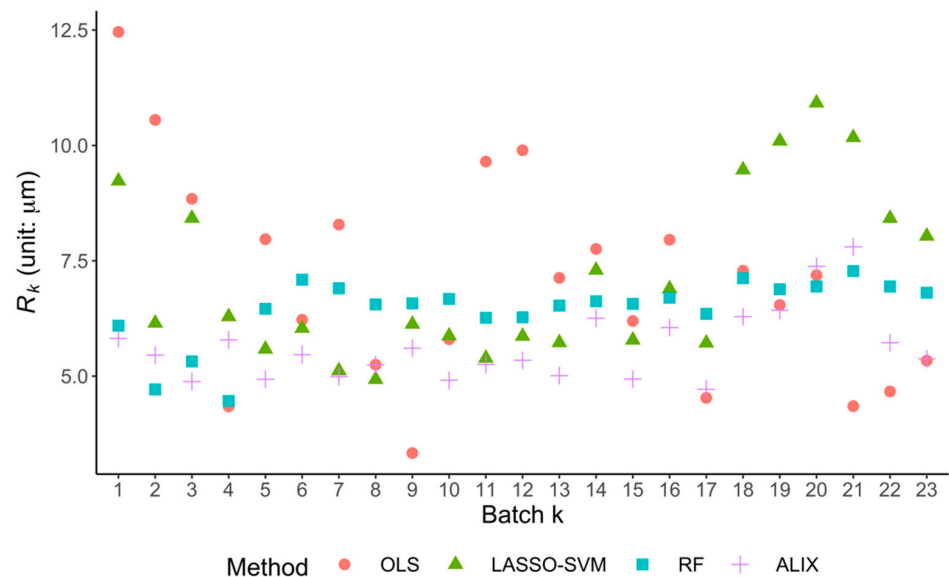


Figure 9. Results of R_k for each model that is trained on batch k .

Further, the results of worst-case scenario prediction W_k for each model that is trained on batch k are plotted in Figure 10, from which we can see that the proposed ALIX method has the best performance with the smallest W_k .

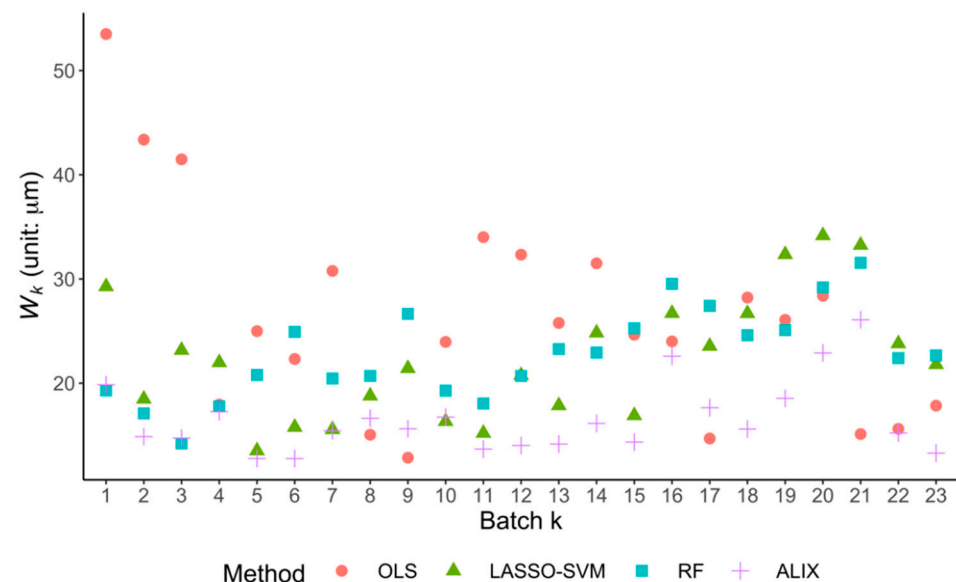


Figure 10. Results of W_k for each model that is trained on batch k .

Moreover, the results of the percentage error P_k for each model that is trained on batch k are plotted in Figure 11, from which we can see similar trends that the proposed ALIX method has better performance than other methods in general.

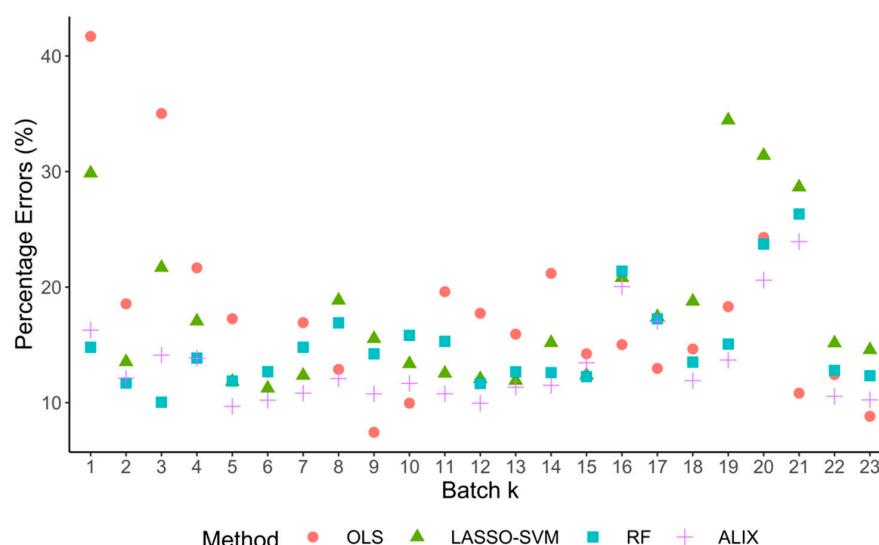


Figure 11. Results of P_k for each model that is trained on batch k .

As a side note, the performance of the proposed method is the best compared with benchmark methods in most cases in Figures 8–11. However, there are still a few cases where the performance of the proposed method is not the best. Since our proposed method is data-driven, the performance of the ALIX method heavily depends on the training data. The 23 batches of experimental data in this study were carried out under different experimental conditions (ambient temperature and spindle speed), so the thermal errors in each batch of data have different laws. This may result in a certain algorithm having good performance in a certain batch of data. However, in practical applications, we need to comprehensively consider different situations. That is why we consider the overall performance over all 23 batches and leverage the statistical analysis to test that our proposed ALIX is statistically significantly better than other methods, which will be described in the following.

Finally, the average values of S_k , R_k , W_k , and P_k over all 23 models (denoted as \bar{S} , \bar{R} , \bar{W} , and \bar{P} , respectively) that are trained on each batch are summarized in Table 5. From Table 5, the ALIX performs better than the OLS, LASSO-SVM, and RF methods in terms of all four metrics: prediction accuracy (S_k), prediction robustness (R_k), worst-case scenario prediction (W_k), and percentage error (P_k). Specifically, compared with the RF method, the proposed ALIX method improves the prediction accuracy by 14.5%, the prediction robustness by 12.9%, the worst-case scenario prediction by 27.6%, and the percentage error by 10.7%. Please note that to establish a model for each batch of data, there are 240 combinations (20 by 4 by 3) of hyperparameters to tune for the RF algorithm whereas there are only 32 combinations for our ALIX method. This shows that the ALIX method can achieve better results while keeping the computational cost of tuning hyperparameters low. In addition, as mentioned before, the nonparametric Mann–Whitney U tests are conducted to test whether the ALIX method is statistically significantly better than benchmark methods. For each evaluation metric, the Mann–Whitney U test returns a p -value that is less than 0.05, indicating that our proposed ALIX method is statistically better than benchmark methods at the significance level of 0.05.

Table 5. Average values of S_k , R_k , W_k , and P_k over all 23 models for each method.

Method	OLS	LASSO-SVM	RF	ALIX
\bar{S} (unit: μm)	9.37	9.42	8.25	7.05
\bar{R} (unit: μm)	7.02	7.07	6.44	5.61
\bar{W} (unit: μm)	26.29	22.00	22.78	16.49
\bar{P} (unit: %)	17.39	17.86	14.94	13.33

5. Experimental Verification

In this section, the experimental verification was conducted to demonstrate the real-world applicability of the proposed ALIX method on thermal error compensation of the CNC system of a machine. The machine tool coordinate origin offset function was adopted as the principle for thermal error compensation [40]. Specifically, during thermal error compensation, the temperature measurements were obtained from temperature sensors (T1–T20), and the thermal error was predicted in real time using the proposed ALIX model. Then, the origin of the workpiece coordinate system was modified based on the predicted thermal error. As a result, the thermal error was offset by corresponding shifts in the origin of the coordinate system. Accordingly, the thermal error compensation was achieved in real time. The compensation function was achieved by a thermal error compensator, which can communicate with the CNC system of the machine tool in real time. The coordinate origin offset function of machine tool was realized by programming the internal programmable logic controller of the machine tool.

To verify the practicability of the proposed ALIX method, the ALIX method was first established based on the experimental data K5 (note that K5 was chosen here only for the purpose of demonstration). Then, three other batches of experimental data with different spindle speeds were selected, denoted as E1–E3. The spindle speeds were controlled at 4000 rpm, 6000 rpm, and 6000 rpm, respectively, and the initial ambient temperatures of E1–E3 were 10.5, 9.8, and 33.1 degrees Celsius, respectively. The worktable feed rate was set at $1500 \text{ mm} \cdot \text{min}^{-1}$. For each batch of the experiment, the thermal error compensation function was set in the sequence “off-on-off-on”, and each state (on or off) lasted for 1 h. The measurement results of thermal errors in the Z direction are plotted in Figure 12, which shows that the thermal errors predicted by the ALIX model were effectively compensated under different spindle speeds and initial ambient temperatures. Specifically, when the compensation function was turned on, thermal errors of the machine tool were well controlled within $10 \text{ }\mu\text{m}$. On the contrary, when the compensation function was turned off, thermal errors were significantly increased. Thus, the real-time applicability of the proposed ALIX algorithm was verified.

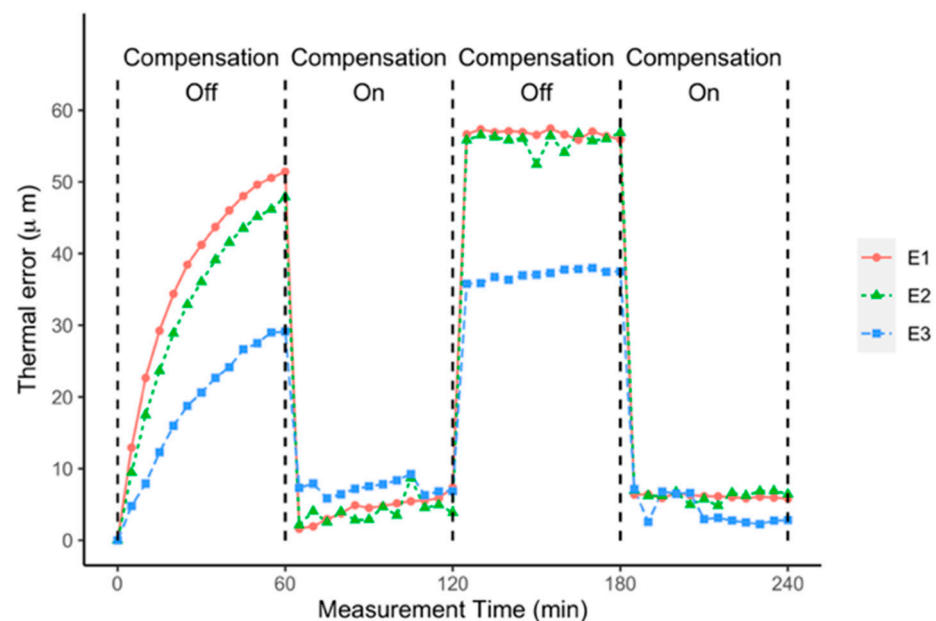


Figure 12. Thermal error measurements in the Z direction when the compensation function is turned off and on.

6. Conclusions

In this paper, an improved robust thermal error prediction model, namely the ALIX method, is proposed based on the adaptive LASSO and XGBoost algorithms. Specifically, the adaptive LASSO method is adopted to select the temperature-sensitive variables, which enjoys the oracle property of variable selection. Further, the XGBoost algorithm is used to provide interpretable results and predict thermal errors based on the selected temperature-sensitive variables. Since the XGBoost algorithm is built based on decision trees, it is immune to multicollinearity and robust to outliers. Based on our experimental data, the ALIX method performs statistically significantly better than the other three existing benchmark methods, in terms of the prediction accuracy with 7.05 μm (over 14.5% improvement than benchmark methods), robustness with 5.61 μm (over 12.9% improvement), worst-case scenario predictions with 16.49 μm (over 25.0% improvement), and percentage error with 13.33% (over 10.7% improvement). The experiment verification indicates that the proposed method can be effectively implemented for the practical thermal error compensation.

Several important directions need future research. First, the exhausted grid search is used for parameter tuning in this paper. However, it is worth studying to introduce more advanced parameter optimization methods, such as particle swarm optimization and genetic algorithm. More importantly, transfer-learning techniques can be incorporated into the current framework to obtain more accurate predictions when only a few measured data points are available under the new working condition. In addition, the spatial correlation on the placement of temperature sensors is not considered in the thermal error modeling. It would be interesting to investigate how the spatial correlation can be leveraged to improve thermal error predictions in the future. Last but not least, the proposed method was only verified on a machine tool in the idle state, and the prediction performance of the proposed method in a real cutting scenario needs future verification.

Author Contributions: Conceptualization, H.Y. and X.W.; Methodology, H.Y., X.Z. and E.M.; Validation, X.W. and X.Z.; Investigation, H.Y. and X.W.; Resources, X.Z. and E.M.; Writing—original draft preparation, H.Y., X.W. and E.M.; Writing—review and editing, H.Y. and X.Z.; Visualization, X.W., X.Z. and E.M.; Funding Acquisition, X.W.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Anhui Provincial Key Research and Development Project of China (grant number 2022f04020005).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zimmermann, N.; Lang, S.; Blaser, P.; Mayr, J. Adaptive Input Selection for Thermal Error Compensation Models. *CIRP Ann.* **2020**, *69*. [\[CrossRef\]](#)
2. Chiu, Y.C.; Wang, P.H.; Hu, Y.C. The Thermal Error Estimation of the Machine Tool Spindle Based on Machine Learning. *Machines* **2021**, *9*, 184. [\[CrossRef\]](#)
3. Mayr, J.; Jedrzejewski, J.; Uhlmann, E.; Donmez, M.A.; Knapp, W.; Härtig, F.; Wendt, K. Thermal Issues in Machine Tools. *CIRP Ann.* **2012**, *61*, 771–791. [\[CrossRef\]](#)
4. Creighton, E.; Honegger, A.; Tulsian, A.; Mukhopadhyay, D. Analysis of Thermal Errors in a High-Speed Micro-Milling Spindle. *Int. J. Mach. Tools Manuf.* **2010**, *50*, 386–393. [\[CrossRef\]](#)
5. Xu, Z.Z.; Liu, X.J.; Kim, H.K.; Shin, J.H.; Lyu, S.K. Thermal Error Forecast and Performance Evaluation for an Air-Cooling Ball Screw System. *Int. J. Mach. Tools Manuf.* **2011**, *51*, 605–611. [\[CrossRef\]](#)
6. Li, F.; Li, T.; Jiang, Y.; Wang, H.; Ehmann, K.F. Explicit Error Modeling of Dynamic Thermal Errors of Heavy Machine Tool Frames Caused by Ambient Temperature Fluctuations. *J. Manuf. Process.* **2019**, *48*, 320–338. [\[CrossRef\]](#)
7. Thiem, X.; Kauschinger, B.; Ihlenfeldt, S. Online Correction of Thermal Errors Based on a Structure Model. *Int. J. Mechatron. Manuf. Syst.* **2019**, *12*, 49–62. [\[CrossRef\]](#)

8. Naumann, C.; Glänzel, J.; Putz, M. Comparison of Basis Functions for Thermal Error Compensation Based on Regression Analysis—A Simulation Based Case Study. *J. Mach. Eng.* **2020**, *20*, 28–40. [\[CrossRef\]](#)
9. Naumann, A.; Herzog, R. Optimal Sensor Placement for Thermo-Elastic Coupled Machine Models. *PAMM* **2021**, *20*, e202000255. [\[CrossRef\]](#)
10. Šwić, A.; Gola, A.; Sobaszek, Ł.; Šmidová, N. A Thermo-Mechanical Machining Method for Improving the Accuracy and Stability of the Geometric Shape of Long Low-Rigidity Shafts. *J. Intell. Manuf.* **2021**, *32*, 1939–1951. [\[CrossRef\]](#)
11. Ramesh, R.; Mannan, M.A.; Poo, A.N. Error Compensation in Machine Tools—A Review: Part II: Thermal Errors. *Int. J. Mach. Tools Manuf.* **2000**, *40*, 1257–1284. [\[CrossRef\]](#)
12. Liu, H.; Miao, E.; Wang, J.; Zhang, L.; Zhao, S. Temperature-Sensitive Point Selection and Thermal Error Model Adaptive Update Method of CNC Machine Tools. *Machines* **2022**, *10*, 427. [\[CrossRef\]](#)
13. Liu, H.; Miao, E.; Zhang, L.; Tang, D.; Hou, Y. Correlation Stability Problem in Selecting Temperature-Sensitive Points of CNC Machine Tools. *Machines* **2022**, *10*, 132. [\[CrossRef\]](#)
14. Yang, J.G.; Deng, W.G.; Ren, Y.Q.; Li, Y.S.; Dou, X.L. Grouping Optimization Modeling by Selection of Temperature Variables for the Thermal Error Compensation on Machine Tools. *China Mech. Eng.* **2004**, *15*, 478–481.
15. Yan, J.Y.; Yang, J.G. Application of Synthetic Grey Correlation Theory on Thermal Point Optimization for Machine Tool Thermal Error Compensation. *Int. J. Adv. Manuf. Technol.* **2009**, *43*, 1124–1132. [\[CrossRef\]](#)
16. Abdulshahed, A.M.; Longstaff, A.P.; Fletcher, S.; Myers, A. Thermal Error Modelling of Machine Tools Based on ANFIS with Fuzzy C-Means Clustering Using a Thermal Imaging Camera. *Appl. Math. Model.* **2015**, *39*, 1837–1852. [\[CrossRef\]](#)
17. Liu, Y.; Miao, E.; Liu, H.; Feng, D.; Zhang, M.; Li, J. CNC Machine Tool Thermal Error Robust State Space Model Based on Algorithm Fusion. *Int. J. Adv. Manuf. Technol.* **2021**, *116*, 941–958. [\[CrossRef\]](#)
18. Wei, X.; Miao, E.; Wang, W.; Liu, H. Real-Time Thermal Deformation Compensation Method for Active Phased Array Antenna Panels. *Precis. Eng.* **2019**, *60*, 121–129. [\[CrossRef\]](#)
19. Zhang, T.; Ye, W.; Shan, Y. Application of Sliced Inverse Regression with Fuzzy Clustering for Thermal Error Modeling of CNC Machine Tool. *Int. J. Adv. Manuf. Technol.* **2016**, *85*, 2761–2771. [\[CrossRef\]](#)
20. Miao, E.; Liu, Y.; Liu, H.; Gao, Z.; Li, W. Study on the Effects of Changes in Temperature-Sensitive Points on Thermal Error Compensation Model for CNC Machine Tool. *Int. J. Mach. Tools Manuf.* **2015**, *97*, 50–59. [\[CrossRef\]](#)
21. Liu, H.; Miao, E.M.; Wei, X.Y.; Zhuang, X.D. Robust Modeling Method for Thermal Error of CNC Machine Tools Based on Ridge Regression Algorithm. *Int. J. Mach. Tools Manuf.* **2017**, *113*, 35–48. [\[CrossRef\]](#)
22. Tan, F.; Yin, M.; Wang, L.; Yin, G. Spindle Thermal Error Robust Modeling Using LASSO and LS-SVM. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 2861–2874. [\[CrossRef\]](#)
23. Fan, J.; Li, R. Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360. [\[CrossRef\]](#)
24. Zhu, M.; Yang, Y.; Feng, X.; Du, Z.; Yang, J. Robust Modeling Method for Thermal Error of CNC Machine Tools Based on Random Forest Algorithm. *J. Intell. Manuf.* **2022**, 1–14. [\[CrossRef\]](#)
25. Wei, X.; Ye, H.; Miao, E.; Pan, Q. Thermal Error Modeling and Compensation Based on Gaussian Process Regression for CNC Machine Tools. *Precis. Eng.* **2022**, *77*, 65–76. [\[CrossRef\]](#)
26. Gao, X.; Guo, Y.; Hanson, D.A.; Liu, Z.; Wang, M.; Zan, T. Thermal Error Prediction of Ball Screws Based on PSO-LSTM. *Int. J. Adv. Manuf. Technol.* **2021**, *116*, 1721–1735. [\[CrossRef\]](#)
27. Liu, J.; Ma, C.; Gui, H.; Wang, S. Transfer Learning-Based Thermal Error Prediction and Control with Deep Residual LSTM Network. *Knowl.-Based Syst.* **2022**, *237*, 107704. [\[CrossRef\]](#)
28. Li, Z.; Zhu, B.; Dai, Y.; Zhu, W.; Wang, Q.; Wang, B. Research on Thermal Error Modeling of Motorized Spindle Based on Bp Neural Network Optimized by Beetle Antennae Search Algorithm. *Machines* **2021**, *9*, 286. [\[CrossRef\]](#)
29. Liang, Y.C.; Li, W.D.; Lou, P.; Hu, J.M. Thermal Error Prediction for Heavy-Duty CNC Machines Enabled by Long Short-Term Memory Networks and Fog-Cloud Architecture. *J. Manuf. Syst.* **2020**, *62*, 950–963. [\[CrossRef\]](#)
30. Liu, J.; Ma, C.; Gui, H.; Wang, S. Thermally-Induced Error Compensation of Spindle System Based on Long Short Term Memory Neural Networks. *Appl. Soft Comput.* **2021**, *102*, 107094. [\[CrossRef\]](#)
31. ISO Copyright Office ISO 230-3 Test Code for Machine Tools Part 3: Determination of Thermal Effects 2020.
32. Li, Y.; Zhao, W.; Lan, S.; Ni, J.; Wu, W.; Lu, B. A Review on Spindle Thermal Error Compensation in Machine Tools. *Int. J. Mach. Tools Manuf.* **2015**, *95*, 20–38. [\[CrossRef\]](#)
33. Zou, H. The Adaptive Lasso and Its Oracle Properties. *J. Am. Stat. Assoc.* **2006**, *101*, 1418–1429. [\[CrossRef\]](#)
34. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Stat. Softw.* **2010**, *33*, 1–22. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Friedman, J.; Hastie, T.; Höfling, H.; Tibshirani, R. Pathwise Coordinate Optimization. *Ann. Appl. Stat.* **2007**, *1*, 302–332. [\[CrossRef\]](#)
36. Chen, T.; Guestrin, C. Xgboost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
37. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [\[CrossRef\]](#)
38. Kuhn, M. Building Predictive Models in R Using the Caret Package. *J. Stat. Softw.* **2008**, *28*, 1–26. [\[CrossRef\]](#)

-
39. Miao, E.; Gong, Y.; Niu, P.; Ji, C.; Chen, H. Robustness of Thermal Error Compensation Modeling Models of CNC Machine Tools. *Int. J. Adv. Manuf. Technol.* **2013**, *69*, 2593–2603. [[CrossRef](#)]
 40. Wei, X.; Miao, E.; Liu, H.; Liu, S.; Chen, S. Two-Dimensional Thermal Error Compensation Modeling for Worktable of CNC Machine Tools. *Int. J. Adv. Manuf. Technol.* **2019**, *101*, 501–509. [[CrossRef](#)]