POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Machine learning models for predicting volumetric errors based on scale and master balls artefact probing data
Auteurs: Authors:	Min Zeng, Miao Feng, J. R. René Mayer, Elie Bitar-Nehme, & Xuan Truong Duong
Date:	2025
Type:	Article de revue / Article
Référence: Citation:	Zeng, M., Feng, M., Mayer, J. R. R., Bitar-Nehme, E., & Duong, X. T. (2025). Machine learning models for predicting volumetric errors based on scale and master balls artefact probing data. CIRP Journal of Manufacturing Science and Technology, 59, 135-157. https://doi.org/10.1016/j.cirpj.2025.03.003

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/64401/
Version:	Version officielle de l'éditeur / Published version Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	Creative Commons Attribution-Utilisation non commerciale-Pas d'oeuvre dérivée 4.0 International / Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND)

Document publié chez l'éditeur officiel
Document issued by the official publisher

Titre de la revue: Journal Title:	CIRP Journal of Manufacturing Science and Technology (vol. 59)
Maison d'édition: Publisher:	Elsevier
URL officiel: Official URL:	https://doi.org/10.1016/j.cirpj.2025.03.003
Mention légale: Legal notice:	© 2025 The Author(s). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

Contents lists available at ScienceDirect

CIRP Journal of Manufacturing Science and Technology

journal homepage: www.elsevier.com/locate/cirpj

Machine learning models for predicting volumetric errors based on scale and master balls artefact probing data

Min Zeng [a,*], Miao Feng [b], J.R.R. Mayer [a], Elie Bitar-Nehme [a], Xuan Truong Duong [c]

[a] Department of Mechanical Engineering, Polytechnique Montréal, 2500, chemin de Polytechnique, Montréal, Québec H3T 1J4, Canada
[b] Department of Computer Science and Operations Research, Université de Montréal, 2900, boul. Edouard-Montpetit, Montréal, Québec, H3T 1J4, Canada
[c] Department of Mechanical Engineering, Dawson College, 3040, rue Sherbrooke, Montréal, Québec H3Z 1A4, Canada

ARTICLE INFO

Keywords:
Volumetric errors
Five-axis machine tool
Machine learning
SAMBA

ABSTRACT

The volumetric accuracy of machine tools is important to both machine tool manufacturers and users. Predicting volumetric errors (VEs) is a pre-requisite for their compensation yielding increased dimensional quality of machined parts. However, predicting VEs in five-axis machine tools is challenging due to the complexity of error sources and their associated physics-based model. Machine learning (ML) is used to predict VEs under no load and stable thermal conditions. Data is acquired using a scale and master ball artefact (SAMBA) and on-machine touch probing. A general process for determining the minimum number of balls required to generate data to satisfactorily train an ML model is proposed. The VEs prediction is verified using synthetic data for inter-axis and some intra-axis geometric errors, and then validated using only experimental data. Different datasets based on decreasing number of balls are tested to train either a Neural Networks (NN) or an eXtreme Gradient Boosting (XGBoost) algorithm to compare their performances. The results show that, both NN and XGBoost are effective to predict VEs of a five-axis machine tool with wCBXfZY(S)t topology regardless of the geometric error parameter values. By using only experimental data of twenty balls to train the models, XGBoost outperforms NN in all four error metrics and processing time. A time efficient scheme was tested whereby only two master balls plus one scale bar dataset and an additional master ball (when only the spindle rotates) were used for training NN.

1. Introduction

Ensuring the machining accuracy of multi-axis, especially five-axis, machine tools is necessary for machine part quality and is the subject of much research [1,2]. There are many different factors affecting the machining accuracy, which can be classified into two categories: the inaccuracy of the machine tool itself and the error sources present during the machining process. In general, machine accuracy is defined in terms of its volumetric errors (VEs) [3]. VEs compensation is an attractive way of improving machining accuracy and has been widely studied. Before correctly compensating for the VEs, it must be accurately predicted. There are two main methods for predicting VEs: one is to use physics-based models and the other is to use data driven, which includes, amongst others, lookup tables and machine learning (ML) models.

Researchers used homogeneous transformation matrices (HTMs), either to describe the relationships between the machine tool structural elements [4] or to represent the geometric error of translation and rotation of each axis, and squareness errors between two linear axes [5]. The result is a generic kinematic model of VEs for five-axis machine tools [6]. Other researchers firstly used an error budget to define thermal errors [7] or kinematic errors [8]. Then they applied homogenous matrix method to propagate assembly errors in multitasking machines [7], or applied homogenous matrix method, FEM analysis along with experiment to evaluate errors in a CNC turning center [8]. Iñigo et al. [9] presented a methodology for simulating and analyzing machine tool error mapping process. Xiang and Altintas [10] proposed a systematic method to model, measure and compensate VEs due to both position dependent (intra-axis) and position independent (inter-axis) geometric errors of five-axis CNC machine tools. They used the screw theory to model the forward and inverse kinematics of the machine tool, and used error motion (intra-axis errors) twists to represent 41 errors (covering both intra-axis and inter-axis errors) of a five-axis machine. Rahman and Mayer [11] predicted VEs of five-axis machine tool through an indirect estimation of intra- and inter-axis error parameters by probing facets on a scale enriched uncalibrated indigenous artefact and considered the

[*] Corresponding author.
E-mail address: min.zeng@polymtl.ca (M. Zeng).

https://doi.org/10.1016/j.cirpj.2025.03.003
Received 19 December 2024; Received in revised form 13 March 2025; Accepted 15 March 2025
Available online 29 March 2025
1755-5817/© 2025 The Author(s). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

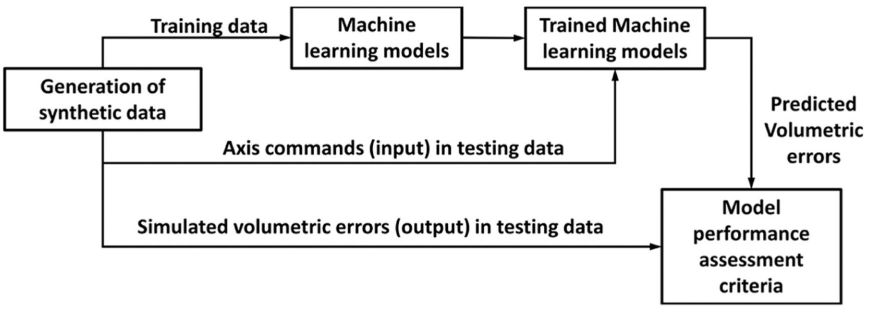
Training data
Machine learning models
Trained Machine learning models
Generation of synthetic data
Predicted Volumetric errors
Axis commands (input) in testing data
Simulated volumetric errors (output) in testing data
Model performance assessment criteria

Fig. 1. Process of study the possibility of ML models for VEs prediction.

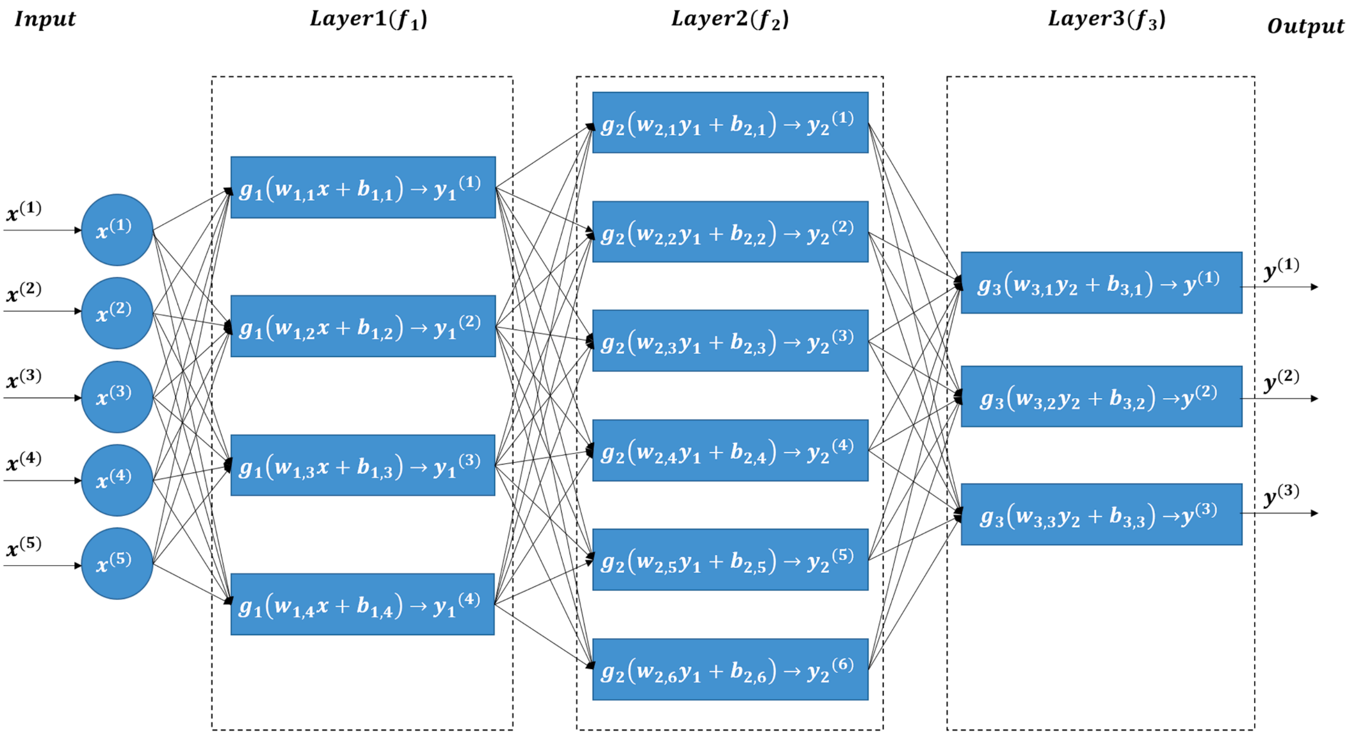
$Input$
$Layer1(f_1)$
$Layer2(f_2)$
$Layer3(f_3)$
$Output$
$x^{(1)}$
$x^{(2)}$
$x^{(3)}$
$x^{(4)}$
$x^{(5)}$
$g_1(w_{1,1}x + b_{1,1}) \to y_1{}^{(1)}$
$g_1(w_{1,2}x + b_{1,2}) \to y_1{}^{(2)}$
$g_1(w_{1,3}x + b_{1,3}) \to y_1{}^{(3)}$
$g_1(w_{1,4}x + b_{1,4}) \to y_1{}^{(4)}$
$g_2(w_{2,1}y_1 + b_{2,1}) \to y_2{}^{(1)}$
$g_2(w_{2,2}y_1 + b_{2,2}) \to y_2{}^{(2)}$
$g_2(w_{2,3}y_1 + b_{2,3}) \to y_2{}^{(3)}$
$g_2(w_{2,4}y_1 + b_{2,4}) \to y_2{}^{(4)}$
$g_2(w_{2,5}y_1 + b_{2,5}) \to y_2{}^{(5)}$
$g_2(w_{2,6}y_1 + b_{2,6}) \to y_2{}^{(6)}$
$g_3(w_{3,1}y_2 + b_{3,1}) \to y^{(1)}$
$g_3(w_{3,2}y_2 + b_{3,2}) \to y^{(2)}$
$g_3(w_{3,3}y_2 + b_{3,3}) \to y^{(3)}$
$y^{(1)}$
$y^{(2)}$
$y^{(3)}$

Fig. 2. Structure of the multilayer perceptron neural network. (adapted from [19]).

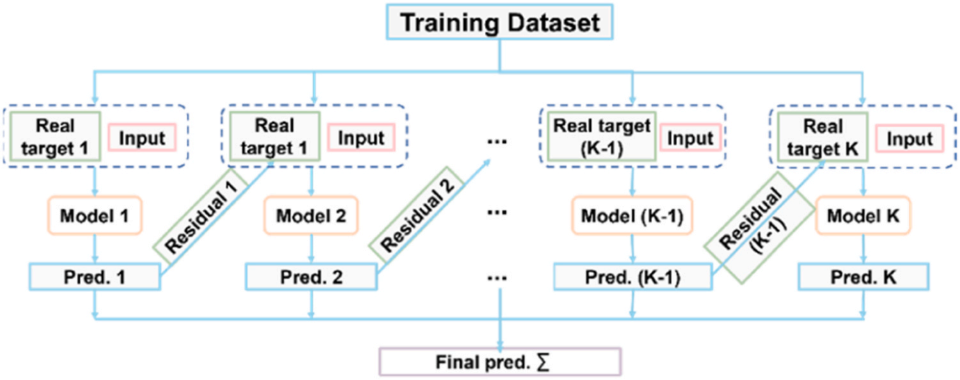

Fig. 3. Work process of XGBoost.

relationship between prediction capability and the complexity of the model. They found that by gradually increasing the modelled errors from only inter-axis errors with ten parameters to also intra-axis for a total of 86 parameters, the prediction capability increases from a mean volumetric residual vectors' norm of 2.87 μm down to 1.38 μm. However, the model complexity and the risk of parameter coupling increases and the model needs to be specifically developed for each new machine topology. Also, the ability to represent the errors is limited by the model

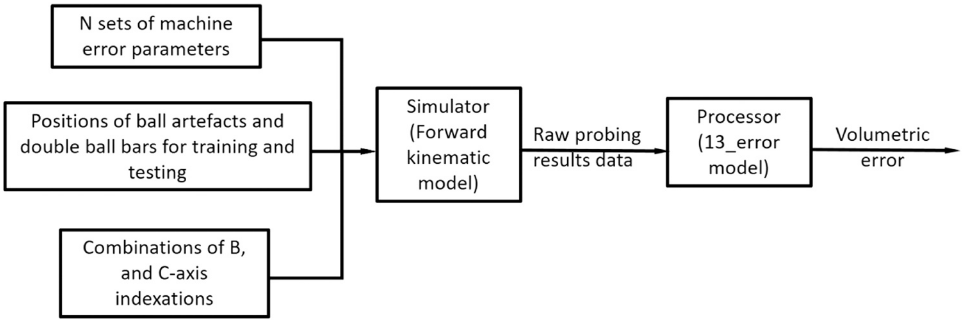
N sets of machine error parameters
Positions of ball artefacts and double ball bars for training and testing
Combinations of B, and C-axis indexations
Simulator (Forward kinematic model)
Raw probing results data
Processor (13_error model)
Volumetric error

Fig. 4. General process of generating synthetic data.

Table 1
Machine error parameters.

Symbol	Description	Range
Workpiece branch location errors		
EA(0Z)B	Out-of-squareness angle of the B-axis relative to the Z-axis	−0.025 µrad to + 0.025 µrad
EC(0X)B	Out-of-squareness angle of the B-axis relative to the X-axis	−0.025 µrad to + 0.025 µrad
EX(0B)C	Distance between the B and C axes.	−10 µm to + 10 µm
EA(0B)C	Out-of-squareness of the C-axis relative to the B-axis	−0.025 µrad to + 0.025 µrad
EB(0X)C	Out-of-squareness of the C-axis relative to the X-axis	−0.025 µrad to + 0.025 µrad
Tool branch location errors		
EB(0X)Z	Out-of-squareness of the Z-axis relative to the X-axis	−0.025 µrad to + 0.025 µrad
EA(0Z)Y	Out-of-squareness of the Y-axis relative to the Z-axis	−0.025 µrad to + 0.025 µrad
EC(0X)Y	Out-of-squareness of the Y-axis relative to the X-axis	−0.025 µrad to + 0.025 µrad
Spindle location errors		
EX(0B)S	X offset of the spindle relative to the B-axis	−10 µm to + 10 µm
EY(0 C)S	Y offset of the spindle relative to the C-axis	−10 µm to + 10 µm
Linear axes component errors		
EXX	Positioning linear error term of the X-axis	−0.025 µm/m to + 0.025 µm/m
EYY	Positioning linear error term of the Y-axis	−0.025 µm/m to + 0.025 µm/m
EZZ	Positioning linear error term of the Z-axis	−0.025 µm/m to + 0.025 µm/m

itself. For example, Slamani and Mayer [12] found that polynomial models of degree three to four are sufficient for most intra-axis errors, but in some cases a quartic polynomial was needed. Li et al. [1] used the 13-line method and the static R-test to obtain the geometric error of the linear and rotary axes, respectively, and constructed the VEs prediction model according to the kinematic model and the identified error. The applied VEs compensation method based on the sag error compensation and table multiplication functions in Siemens 840D numerical control system reduced the largest geometric error of an NAS (National Aeronautical standard) test piece by 63.5 %. The mentioned physics-based models used to predict VEs are usually topology dependent and can only take into account errors that are considered in the model. Modelling those requires expert knowledge.

Araie et al. [13] designed a method and apparatus for correcting positioning errors on a machine tool. In this patent, the neural networks inferred the thermal deformation of the machine components based on measured temperatures. A correction signal is then obtained using the inferred thermal deformation values. The axis correction signals, in combination with drive signals, are used to move the tool and/or workpiece to correct any positioning error. Fines and Agah [14] tested several artificial neural network architectures to compensate machine tool positioning errors. They selected and implemented the artificial neural network optimized with Levenberg-Marquardt algorithm on a single spindle two-axis CNC lathe resulting in a reduction of the maximum average error from 142 µm down to 24 µm. He et al. [15] collected data using the step diagonal test introduced in [16], and compared the motion errors predicted by Least Square Support Vector Machine (LS-SVM) and by ANN. LS-SVM outperforms ANN, with a maximum average relative prediction error of 4.67 % and 28.42 % for LS-SVM and ANN, respectively. Wan et al. [17] proposed a Gaussian process regression based (GPR-based) method to predict and compensate the VEs. The mean value of the VEs and its prediction error were 0.16 mm and 0.02 mm, respectively. Nguyen et al. [18] trained linear, Support Vector Machine (SVM), eXtreme Gradient Boosting (XGBoost), Artificial Neural Networks (ANNs) with Single, MultiOutput and

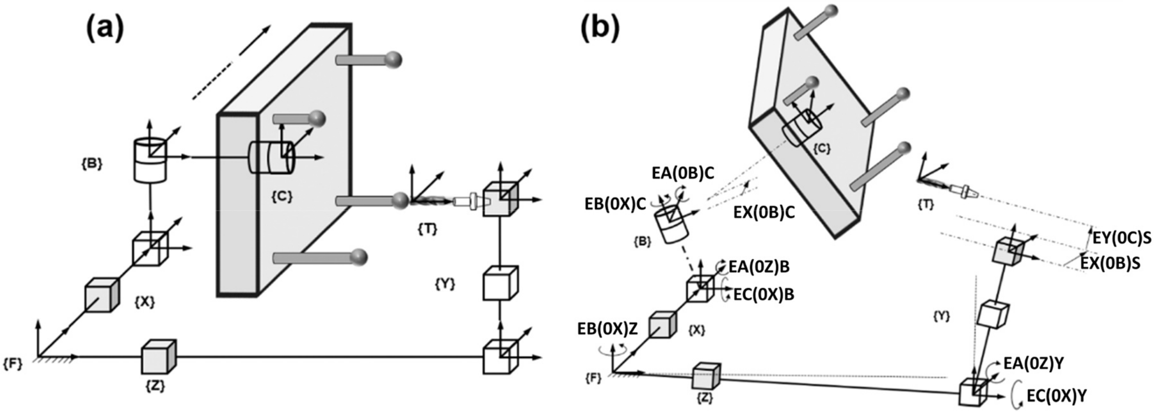

Fig. 5. Kinematic model (a) without errors; (b) with errors.
(adapted from [26]).

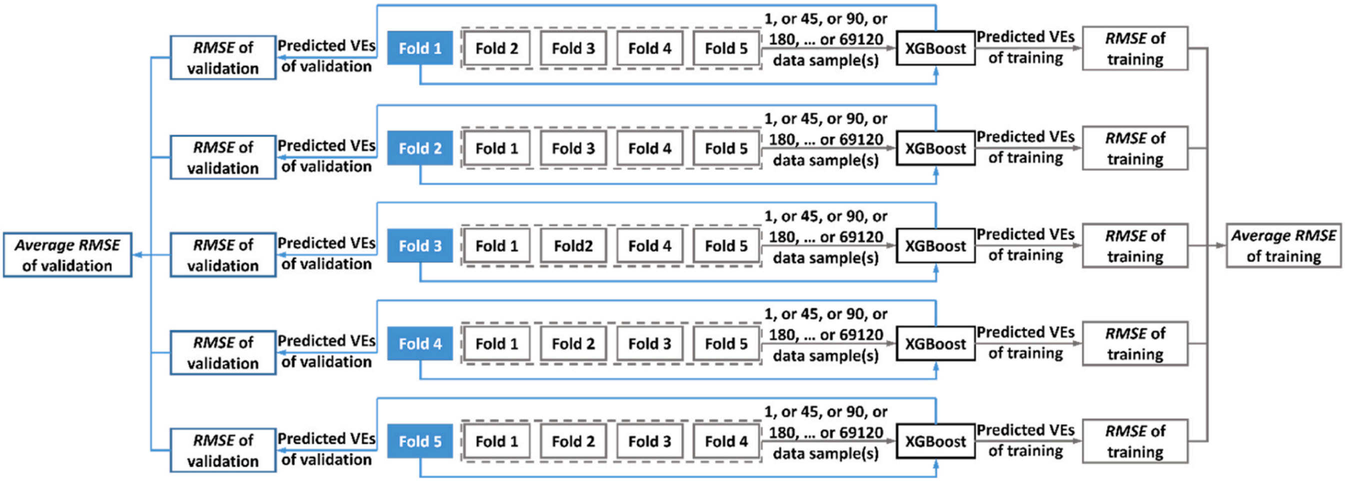
Average RMSE of validation
RMSE of validation
Predicted VEs of validation
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
1, or 45, or 90, or 180, … or 69120 data sample(s)
XGBoost
Predicted VEs of training
RMSE of training
Average RMSE of training

Fig. 6. Process of generating learning curve of an XGBoost regression model based on synthetic data.

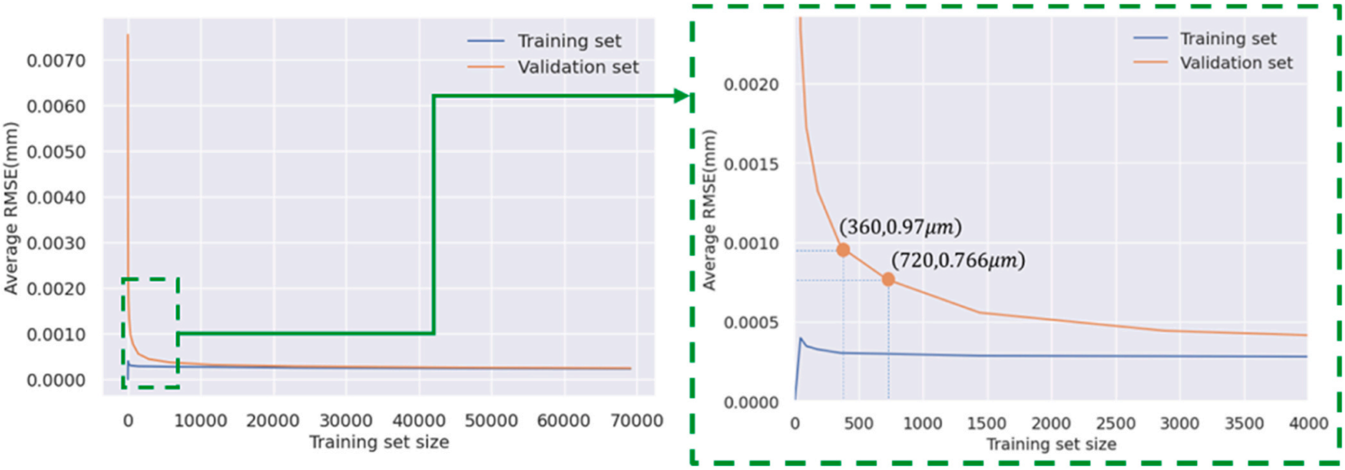

Fig. 7. Learning curve of an XGBoost regression model (average *RMSE* for training set and validation set using different training size).

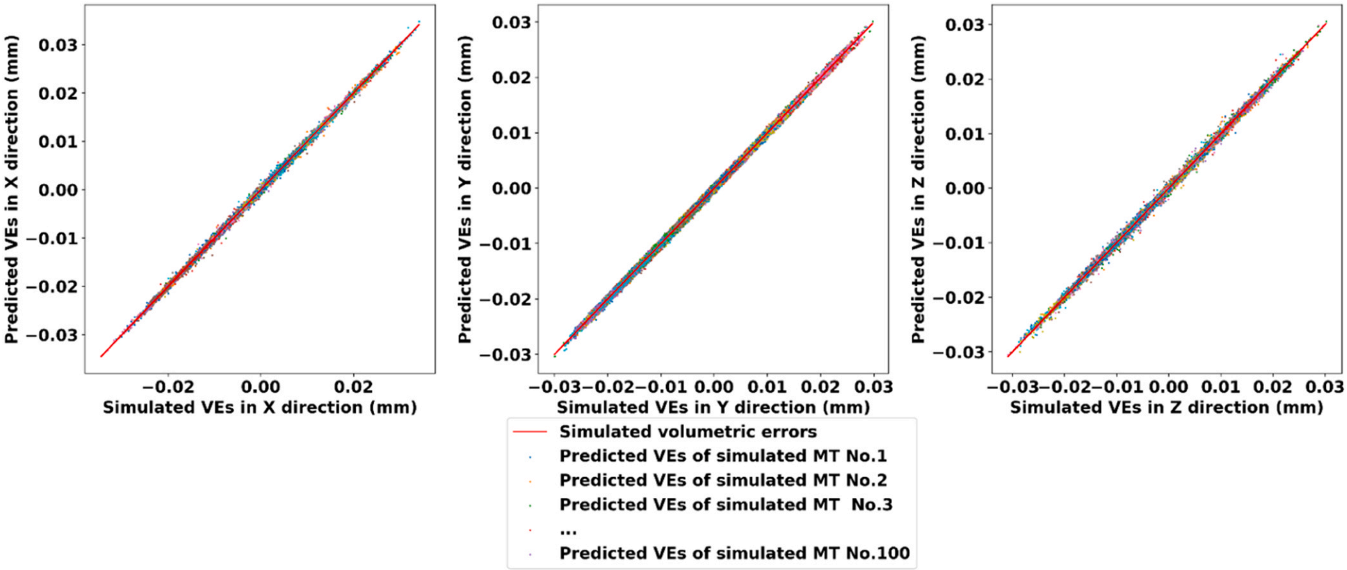

Fig. 8. Total of 19200 (192 *100) predicted VEs of one hundred simulated machine tool instances by NN trained with synthetic data that uses a randomly generated measurement strategy and 100 instances of machine tool's inter- and intra-axis geometric errors.

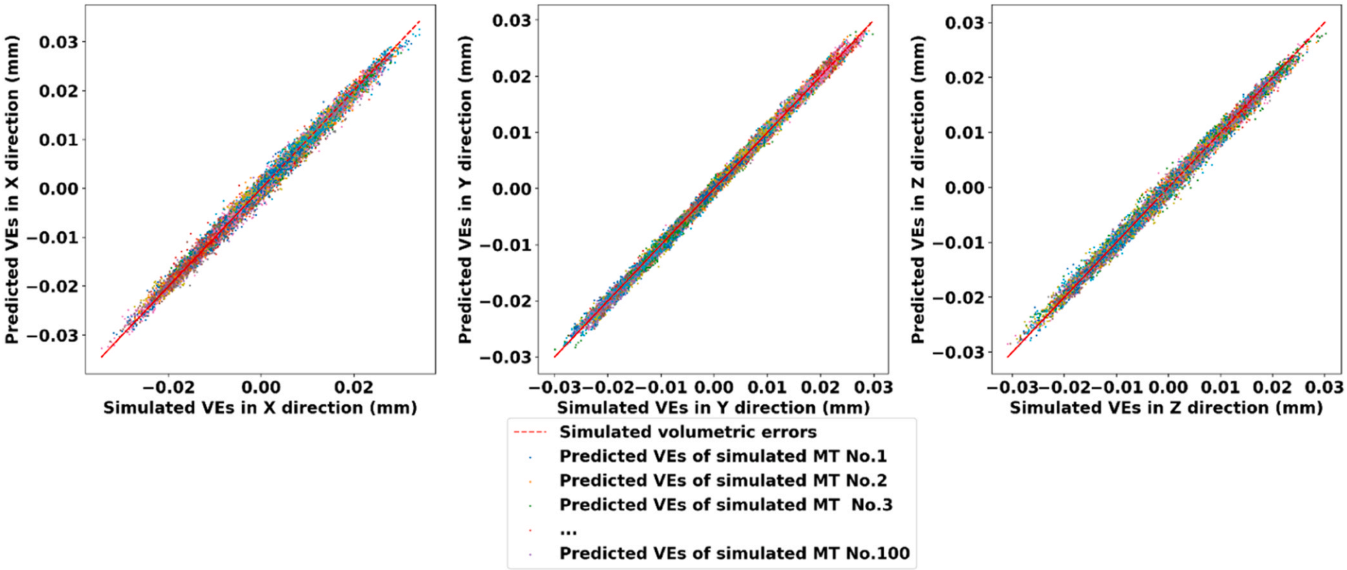

Fig. 9. Total of 19200 (192 *100) predicted VEs of one hundred simulated machine tool instances by XGBoost trained with synthetic data that uses a randomly generated measurement strategy and 100 instances of machine tool's inter- and intra-axis geometric errors.

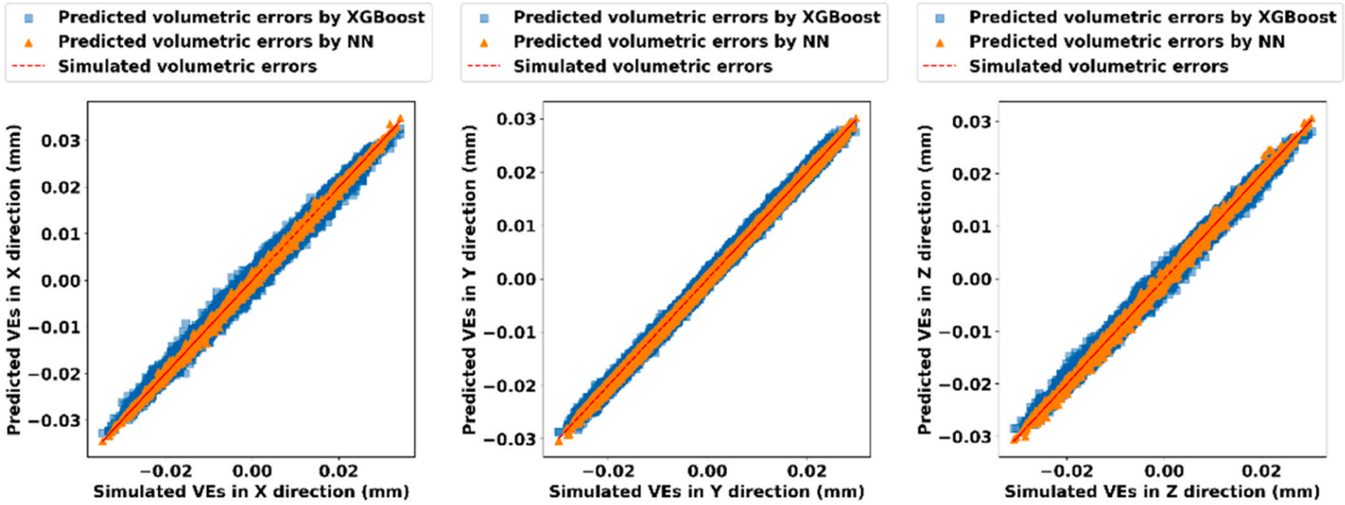

Fig. 10. Comparison of VEs predicted using NN and XGBoost based on synthetic data that uses a randomly generated measurement strategy and 100 instances of machine tool's inter- and intra-axis geometric errors.

Chained MultiOutput regression using VEs of a five-axis large gantry tilting head machine tool, collected by a laser tracker, then, tested and compared their performances based on three statistical metrics: Root Mean Square Error *(RMSE*), Mean Absolute Error (*MAE*) and Coefficient of Determination (R^2). The SVR with Single regression model had the best VEs predictive performance with the *RMSE* of 0.032 mm, a 27 % improvement compared to the polynomial technique.

In the abovementioned ML-based VEs prediction methods, each article only studies VEs of one specific machine tool error instance, i.e. under a specific case of error condition. The three main contributions of this paper are: 1) to study the robustness of selected ML models to predict VEs of machines with the same topology but different machine error instance; 2) to investigate complementing the limited number of experimental data with semi-synthetic data from a calibrated kinematic model; and 3) to propose a general process for determining the minimum number of measurement data required to train a well-performing ML model. The paper is organized as follows: the possibility of ML models for VEs prediction is studied by simulations in Section 2. Section 3 presents the experimental validation, followed by a discussion on selecting the suitable data in Section 4 and finally Section 5 presents conclusions.

2. Performance study of ML models for VEs prediction using synthetic data

Volumetric error (VE) is the positioning error Cartesian vector of the tool tip with respect to its desired position in the workpiece frame projected in the machine coordinate system. A general process (Fig. 1) is proposed to discuss the use of ML models for VEs prediction using synthetic data. Generated synthetic data is split into training and testing sets. Two ML models are trained using the training set, and evaluated using the testing set. The performance of the two trained ML models is studied and compared.

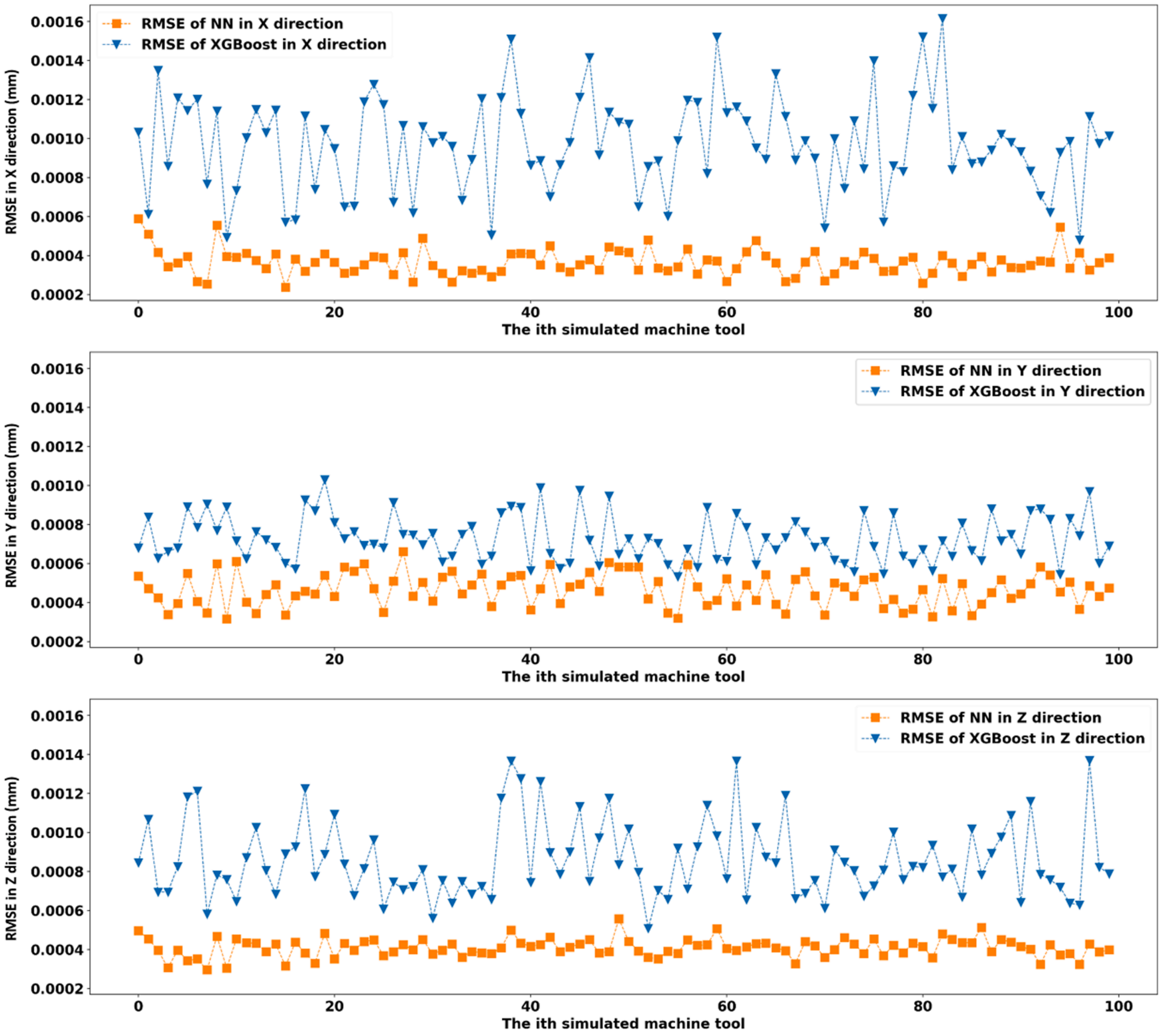

Fig. 11. *RMSE* of NN and XGBoost trained with synthetic data that uses a randomly generated measurement strategy and 100 instances of machine tool's inter- and intra-axis geometric errors.

2.1. The machine learning models considered

In order to verify the possibility of using ML models to predict VEs, two popular ML models are selected for study and performance comparison: Neural network and eXtreme Gradient Boosting. Those two ML models could both deal with tabular data and could capture the nonlinear relationship between input and output.

2.1.1. Neural network [19]

A neural network (NN) is a mathematical function that can be simply expressed as Eq. (1). f_{NN}, as a nested function has different forms depending on the number of NN layers. Let us take a three-layer NN that returns a scalar as an example (see Eq. (2)),

$$\boldsymbol{y} = \boldsymbol{f_{NN}}(\boldsymbol{x}) \tag{1}$$

$$\boldsymbol{y} = \boldsymbol{f_{NN}}(\boldsymbol{x}) = \boldsymbol{f_3}(\boldsymbol{f_2}(\boldsymbol{f_1}(\boldsymbol{x}))) \tag{2}$$

In Eq.(2), f_3 is a scalar function for the regression task, f_1 and f_2 are vector functions in the following form (see Eq. (3)):

$$f_l(z) \overset{\text{def}}{=} g_l(W_l z + b_l) \tag{3}$$

where l is the layer index, ranging from one to any number of layers; g_l is an activation function; the matrix W_l and the vector b_l are parameters of each layer learned by optimizing a specific cost function using the gradient decent.

The multilayer perceptron (MLP), or vanilla NN, architecture of NN is considered. An MLP with five-dimensional input, two layers consisting of four and six neural units and one output layer with three units is depicted in Fig. 2. Firstly, all the original inputs in blue circle are sent directly to each neural unit in the first layer. Then all the inputs sent to each neural unit are combined together to form an input vector. After that, each neural unit applies a linear transformation on each input vector. Then each neural unit applies an activation function on the result of the linear transformation to obtain the output value. The output of the first layer becomes the input of the second layer, and the same process is repeated. Finally, the output of layer 2 is sent to the neural unit of layer 3, after the same operation, the final output value is calculated.

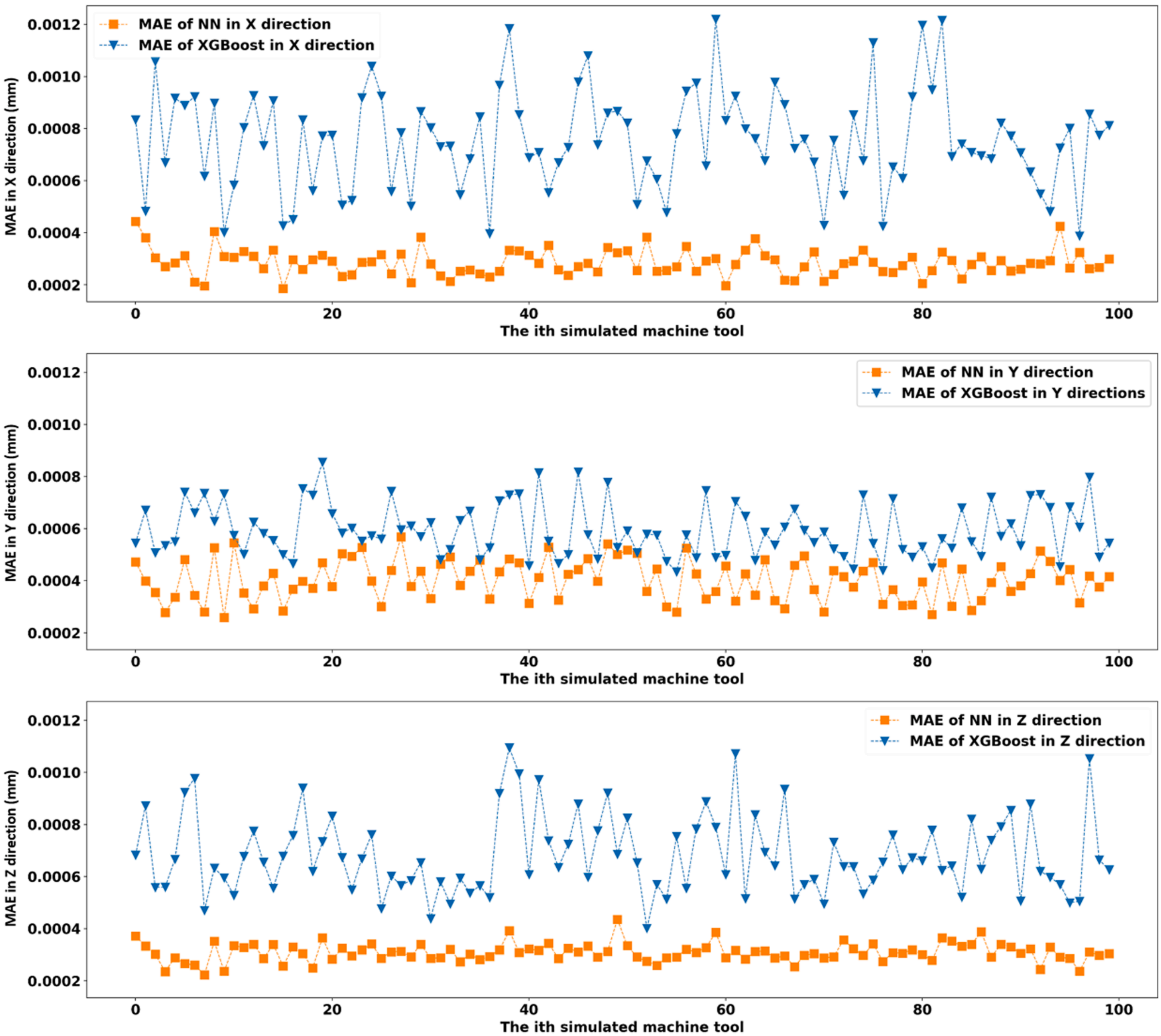

Fig. 12. *MAE* of NN and XGBoost trained with synthetic data that uses a randomly generated measurement strategy and 100 instances of machine tool's inter- and intra-axis geometric errors.

Depending on the activation function of the neural unit in the last layer, the NN can solve either regression problem using linear activation function or to solve a binary classification problem by applying a logistic function.

In addition, an efficient algorithm called backpropagation is used to update the parameter values of the NN. During gradient descent, the parameters of the NN are updated in each iteration of training, in proportion to the partial derivatives of the cost function with respect to the current parameters. The NN used in our study employs backpropagation.

2.2. eXtreme gradient boosting

With the advantages of short processing time, high predictive accuracy and scalability, eXtreme Gradient Boost (XGBoost) has been widely used to solve regression problems in many fields, such as medical, banking and mechanical machining [18]. Considering both the accuracy, the ability of handling non-linear relationships, its robustness against overfitting and processing time of the model, XGBoost is chosen as one of the models to test the possibility of predicting VEs.

XGBoost is a gradient tree boosting algorithm [20] introduced by Tianqi Chen in 2015, which follows gradient boosting [21]. Instead of obtaining all the trees simultaneously, the algorithm iteratively fits the model based on the residuals from the previous training process and the newly added trees (Fig. 3) [22]. To prevent the model from overfitting, its objective function (Eq.(4)) introduces a regularization term to balance the loss and complexity of the model [20].

$$\mathrm{L} = l(\widehat{Y}, Y) + \sum_{k=1}^{K} \Omega(F_{Tk}) \tag{4}$$

where Ω(.) is the regularization term to measure the complexity of the model to penalize the complicated models. For more details about tree-based algorithms, please refer to the related text book [23].

2.3. Model performance assessment criteria

Three metrics are calculated to analyze the results in each direction: root mean square error (*RMSE*) [24], mean absolute error (*MAE*) [24],

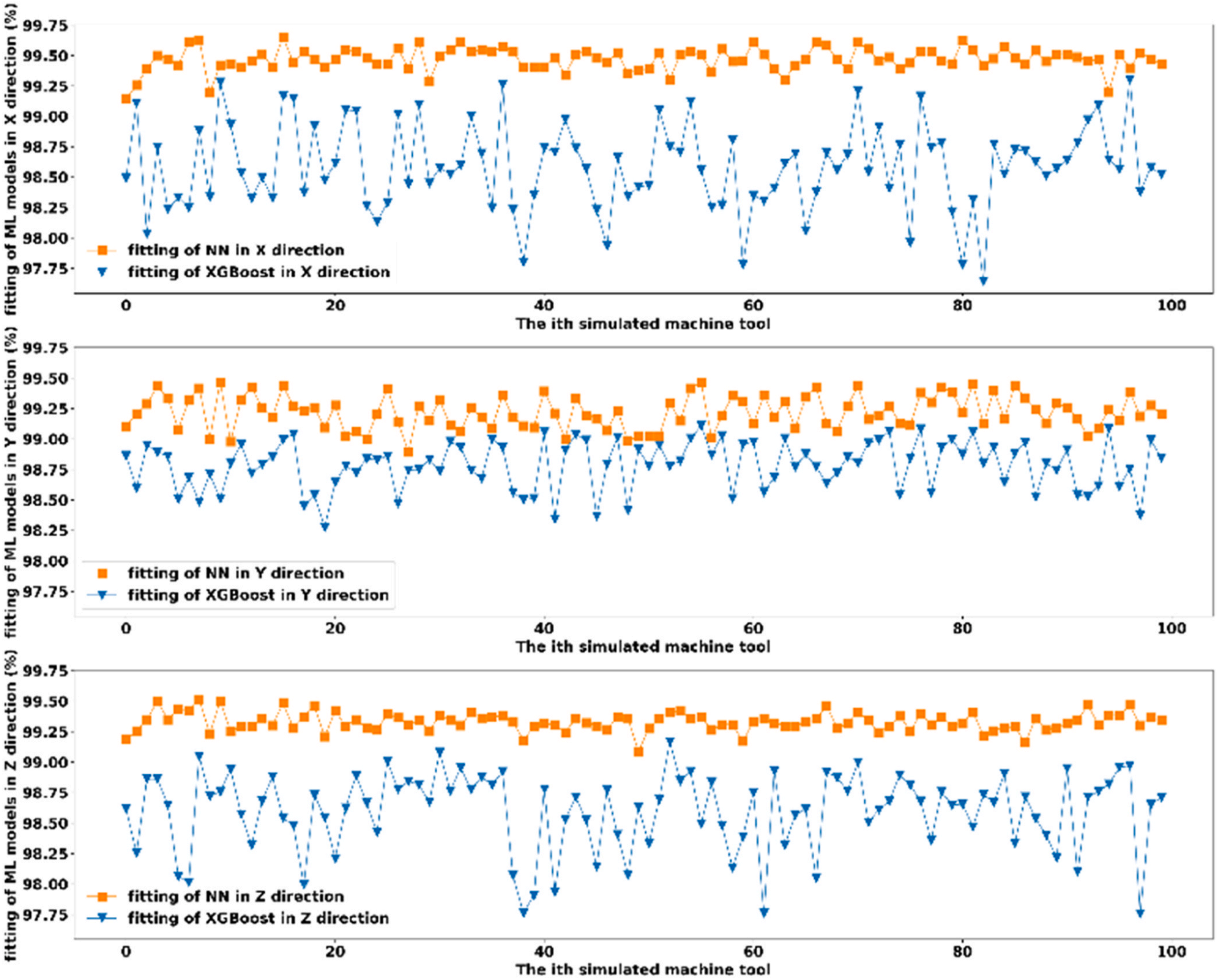

Fig. 13. *fitting* of NN and XGBoost trained with synthetic data that uses a randomly generated measurement strategy and 100 instances of machine tool's inter- and intra-axis geometric errors.

and *fitting* [25] using Eqs. (5)–(7), respectively, where n is the number of observations, $\hat{y}$ is the predicted value of the VEs component, y is the true target value, and $y_{\max}$ and $y_{\min}$ are the maximum and minimum values of the true target values, respectively.

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2}{\boldsymbol{n}}} \tag{5}$$

$$\boldsymbol{MAE} = \frac{\sum_{i=1}^{n} |\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i|}{\boldsymbol{n}} \tag{6}$$

$$\boldsymbol{fitting}(\%) = (\mathbf{1} - \frac{\boldsymbol{RMSE}}{\boldsymbol{y}_{max} - \boldsymbol{y}_{min}}) \times \mathbf{100} \tag{7}$$

Moreover, to have an overview of the performance of different models, prediction error norm ratio with respect to the norm of the VE ($PENR_{VE}$) (Eq.(8)) and average $PENR_{VE}$ ($\overline{PENR_{VE}}$) of each model are calculated.

$$\boldsymbol{PENR}_{\boldsymbol{VE}} = \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{VE}\|} \tag{8}$$

where $\|e\|$ and $\|VE\|$ are the norm of prediction error and norm of VE, respectively.

2.4. Generation of synthetic data

In Section 2, the two ML models are trained using synthetic data based on two measuring strategies: 1) a large randomly generated measuring strategy; and 2) the same measurement strategy as the one to be used for the experimental validation, which takes into account realistic situations, such as collisions and accessibility of the measured calibration balls. The general process of generating synthetic data is illustrated in Fig. 4. N = 100 sets of machine error parameters are generated, representing N different instances of machine tool's inter- and intra-axis geometric errors with the same topology wCBXfZY(S)t. As shown in Table 1, each set consists of thirteen machine error parameters, including axis location errors such as out-of-squareness between the Y and X axes, spindle location errors and the linear axes positioning linear errors. The randomly generated measurement strategy consists of randomly generated nominal positions of two double ball bars and balls of different stem lengths, among which, some are used for training purpose, and others for testing. Two sets of rotation axis position pairs, denoted as (b_i, c_i), where b_i and c_i are the command angular positions of the B- and C-axis, respectively, are created, one set of 180 (b_i, c_i) pairs

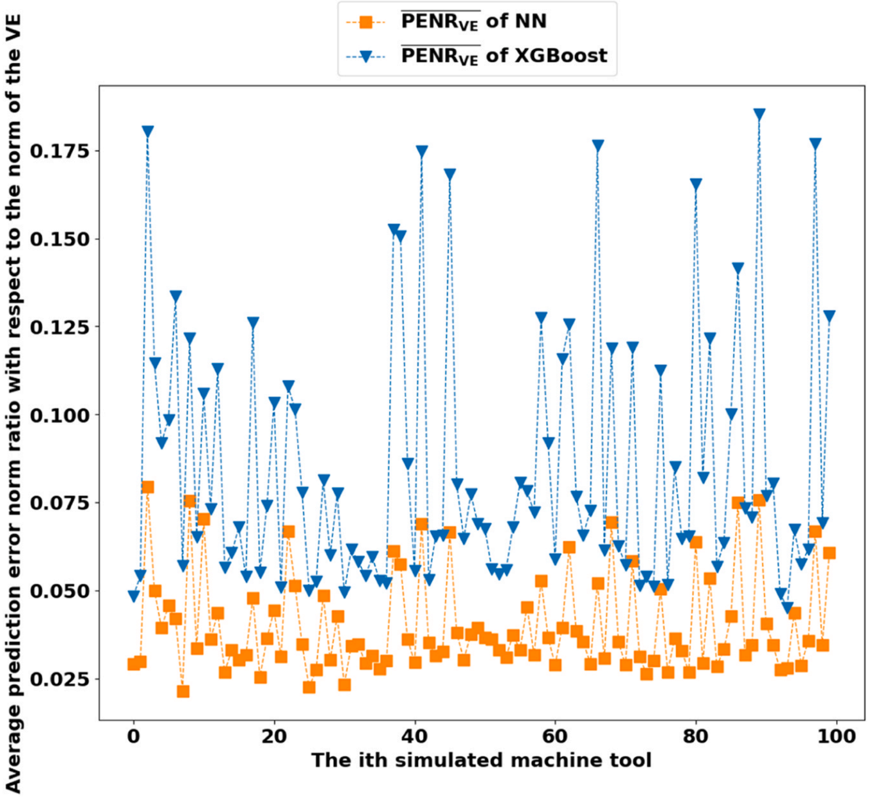

Fig. 14. Average prediction error norm ratio with respect to the norm of the VE using NN and XGBoost trained with synthetic data that uses a randomly generated measurement strategy and 100 instances of machine tool's inter- and intra-axis geometric error.

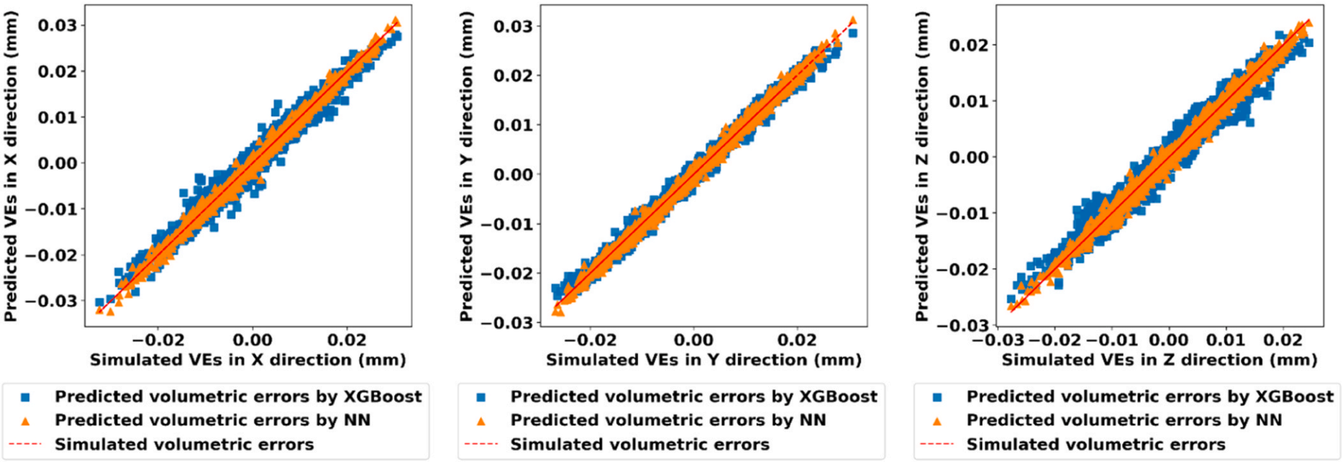

Fig. 15. Comparison of predicted VEs obtained by NN and XGBoost using synthetic data that uses the same measurement strategy as the one to be used for the experimental validation and 100 instances of machine tool's inter- and intra-axis geometric errors.

for training and the other set of 45 (b_i, c_i) pairs for testing. The strategies also include which balls are measured at each (b_i, c_i) rotary axes position pairs. The strategies are identical for all simulated machine tool error instances. Axis commands are computed using the nominal kinematic model (illustrated in Fig. 5(a)). Finally, the raw probing results data are fed into the thirteen_error model (illustrated in Fig. 5(b)) to generate the N sets of translational VEs in the machine tool foundation frame. A translational VE vector is obtained for each ball measurement.

2.5. Learning Curve

To determine the amount of training data required to achieve good prediction performance, the variation of average *RMSE* of the training set and of the validation set of XGBoost trained with different number of training data, also known as the learning curve, is generated. Based on the general process of generating synthetic data (Fig. 4), one simulated machine tool error instance of the thirteen errors (Table 1), nominal position of four master balls and one scale bar ball, (b_i, c_i) pairs are used to generate a strategy. The B- and C-axis are rotated from −90 degrees to + 90 degrees and from −180 degrees to + 180 degrees, respectively. To make the generation of random degrees uniformly distributed, the B-axis and C-axis are designed to have 600 and 300 random degree values in 5-degree increments, respectively, so that there are 21600 (b_i, c_i) pairs for each ball, and in total 86400 (21600 *4) measurements will be used in

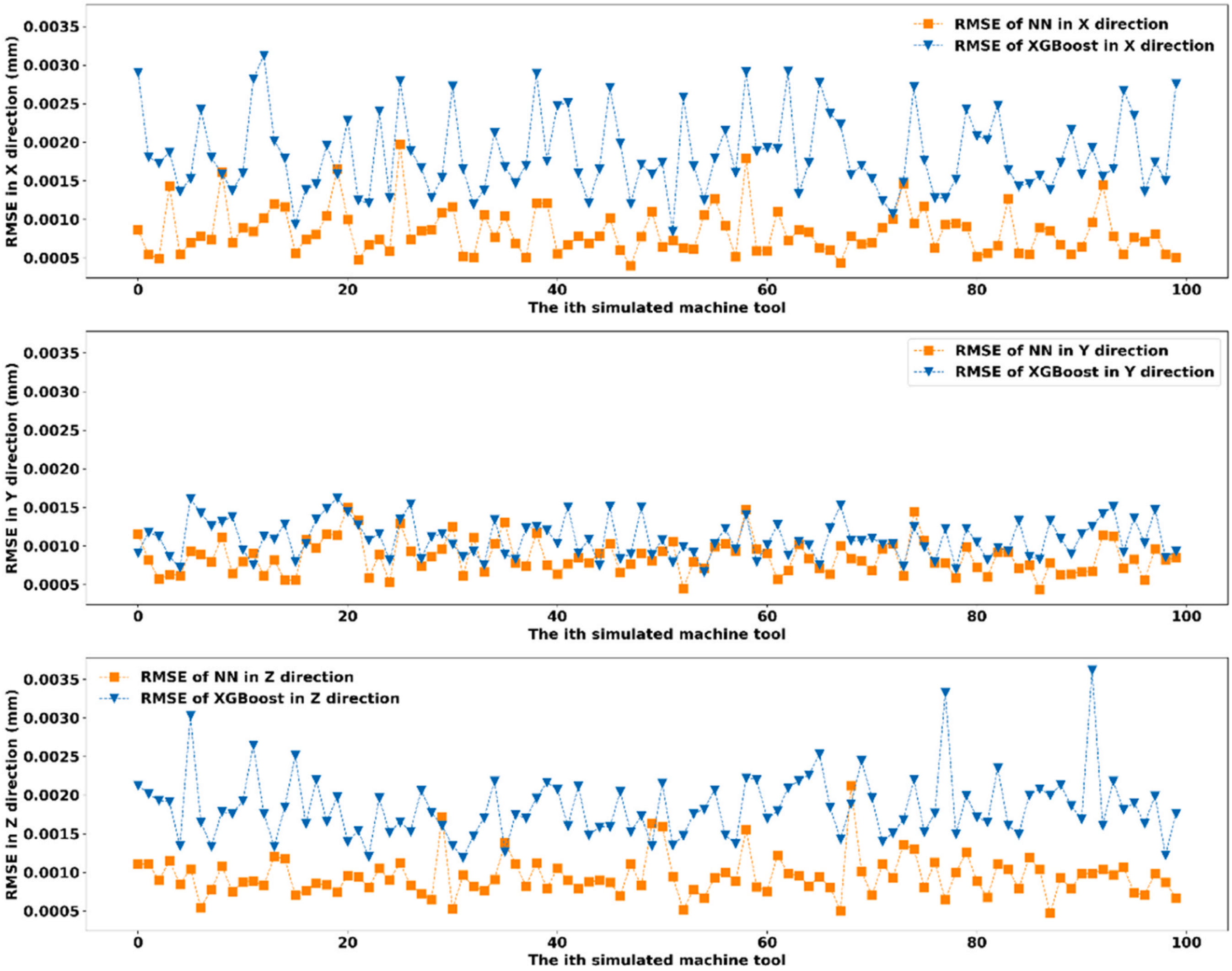

Fig. 16. *RMSE* of NN and XGBoost with synthetic data that uses the same measurement strategy as the one to be used for the experimental validation and 100 instances of machine tool's inter- and intra-axis geometric errors.

this section. Applying the five-fold cross-validation technique [27] (Fig. 6), 86400 samples are randomly divided into five equal size folders of 17280 samples. Five training and validation processes are conducted. For each training and validation process, data from four out of the five folders are used for training, and data from the remaining folder is used for validation. Thus, five sets of predictions are made. The *RMSE* is computed for each dataset and the average *RMSE* considering all datasets. Initially, only one sample is taken from the training set (data in the four folders) to train the model and then compute the average *RMSE* for the validation set and the individual training data. The average *RMSE* of the model trained with only one training sample is 0.003 nm, while the average *RMSE* of the validation samples is about 7.5 μm. The reason for this is that a model trained with only one sample does not have the generalization ability to accurately predict unseen data. Forty-five samples from the training dataset (data in the four folders) are then used to train, each time doubling the training size until 46080, and finally the entire training set of 69,120 instances is used. The variation of average *RMSE* of the training and validation sets with increasing training size is shown in Fig. 7. When the training size is very small, the learning curve of the validation set is at the highest point but decreases with the increase of the sample size of the training set and gets closer to the learning curve of the training set. The average *RMSE* of the validation set decreases with the increasing size of the training set, and the gap between the two is smaller, indicating that the model is a better fit and there is no overfitting or underfitting. The validation average *RMSE* decreases rapidly until the sample size is 360 (average *RMSE* of 0.97 μm), after which the improvement is slower and hardly changes after the sample size reaches 3000. We can conclude that with XGBoost, 3000 training samples are needed to get very good performance. However, using 720 training samples (four balls and approximately 225 (b_i, c_i) pairs; however, not all four balls are measured at each of the 225 (b_i, c_i) pairs), the average VE norm of the validation set is about 12 μm, and the average *RMSE* is only 0.766 μm. Therefore, 720 samples are used as the training size for generating synthetic data.

Although the size of 720 samples is small compared to 3000 samples, it is still difficult to obtain such amount of data in practice. Thus, in Section 3, we use a dataset containing 264 experimental data from [26] to validate and compare the performances of ML models.

2.6. Performance of ML models trained with the large synthetic dataset that uses a randomly generated measurement strategy

Based on the results obtained in Section 2.4, the desired training size is at least 720. Those 720 samples are generated using four balls and 180 (b_i, c_i) pairs. To provide the same type of information as for a SAMBA test, we add another 12 samples, which includes four data samples from the two scale bar balls, four data samples from one of the four balls (when only the spindle is rotated), and four data samples from four balls

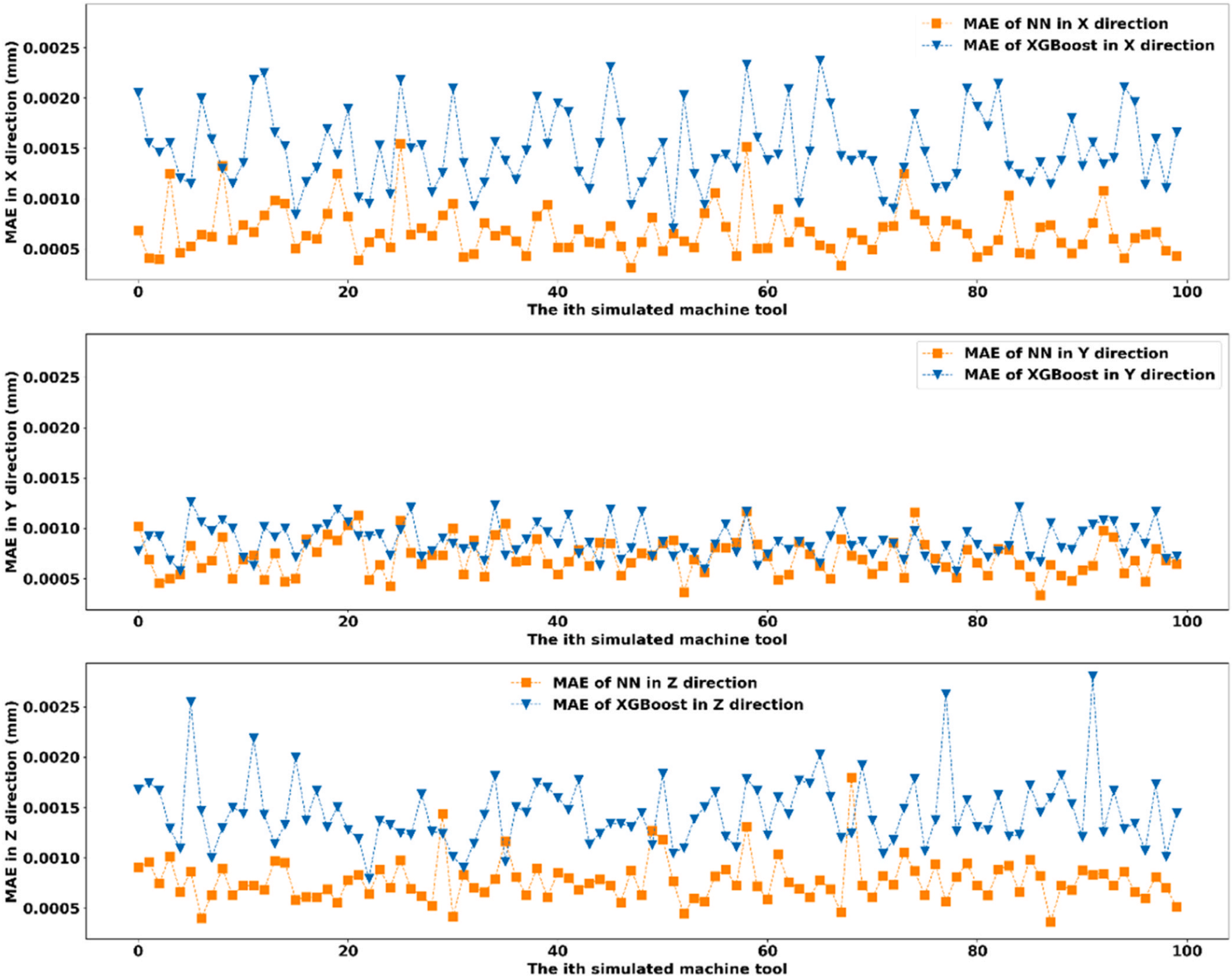

Fig. 17. *MAE* of NN and XGBoost with synthetic data that uses the same measurement strategy as the one to be used for the experimental validation and 100 instances of machine tool's inter- and intra-axis geometric errors.

(when the B-axis and C-axis are at 0°). The training is conducted for one hundred different simulated instances of machine tool errors. For testing, different four balls location and 45 new (b_i, c_i) pairs are used. Both XGBoost and NN models are evaluated.

2.6.1. Performance of NN trained with the large synthetic dataset that uses a randomly generated measurement strategy

Fig. 8 shows the predicted VEs using NN. In Fig. 8, the red line indicates the perfect prediction line, and the dots of different colours indicate the predicted VEs of different simulated machine tool error instances. Most predicted VE components are very close to the simulated (true) VEs, for simulated machine tool error instances. The results show that the same NN structure and hyper-parameters provide similar results regardless of the machine tool's geometric error parameters.

2.6.2. Performance of XGBoost trained with the large synthetic dataset that uses a randomly generated measurement strategy

The predicted VEs by XGBoost are shown in Fig. 9. Similar to the previous NN plot, the red line indicates a perfect prediction, and the different colour dots are the predicted VEs of different simulated machine tool instances. The coloured dots are located near the red line, meaning that the model can closely predict VEs. Results for the x and z components are not as good as for the y component. The results show that, independently of the simulated machine tool error instance, the same structure and hyper-parameter of XGBoost yields similar results.

2.6.3. Comparison results using NN and XGBoost based on the large synthetic data that uses a randomly generated measurement strategy

Translational VEs predicted using NN and XGboost are shown in Fig. 10. The orange and blue dots indicate the VEs predicted by NN and XGBoost, respectively, while the red dotted line represents a perfect prediction. The width of the blue region of the XGBoost predicted VEs is larger than the width of the orange zone of the NN predicted VEs, which indicates predicted VEs by XGBoost are further away from the perfect prediction than the predicted VEs by NN. Results show that NN performs better than XGBoost in predicting the VEs when training the ML model using synthetic data for the simulation conditions used.

In all three directions, the *RMSE* (Fig. 11) and *MAE* (Fig. 12) of NN are smaller than those of XGBoost, and *fitting* (Fig. 13) of NN is larger than that of XGBoost, also with less fluctuation. The results show that NN has a more robust performance in predicting VEs based on synthetic data that uses a randomly generated measurement strategy.

Fig. 14 shows the average prediction error norm ratio with respect to the norm of the VE (Eq. (8)) of each simulated machine tool using NN and XGBoost. NN performs better than XGBoost with the worst-case prediction error norm ratio of 0.08 and 0.19, respectively.

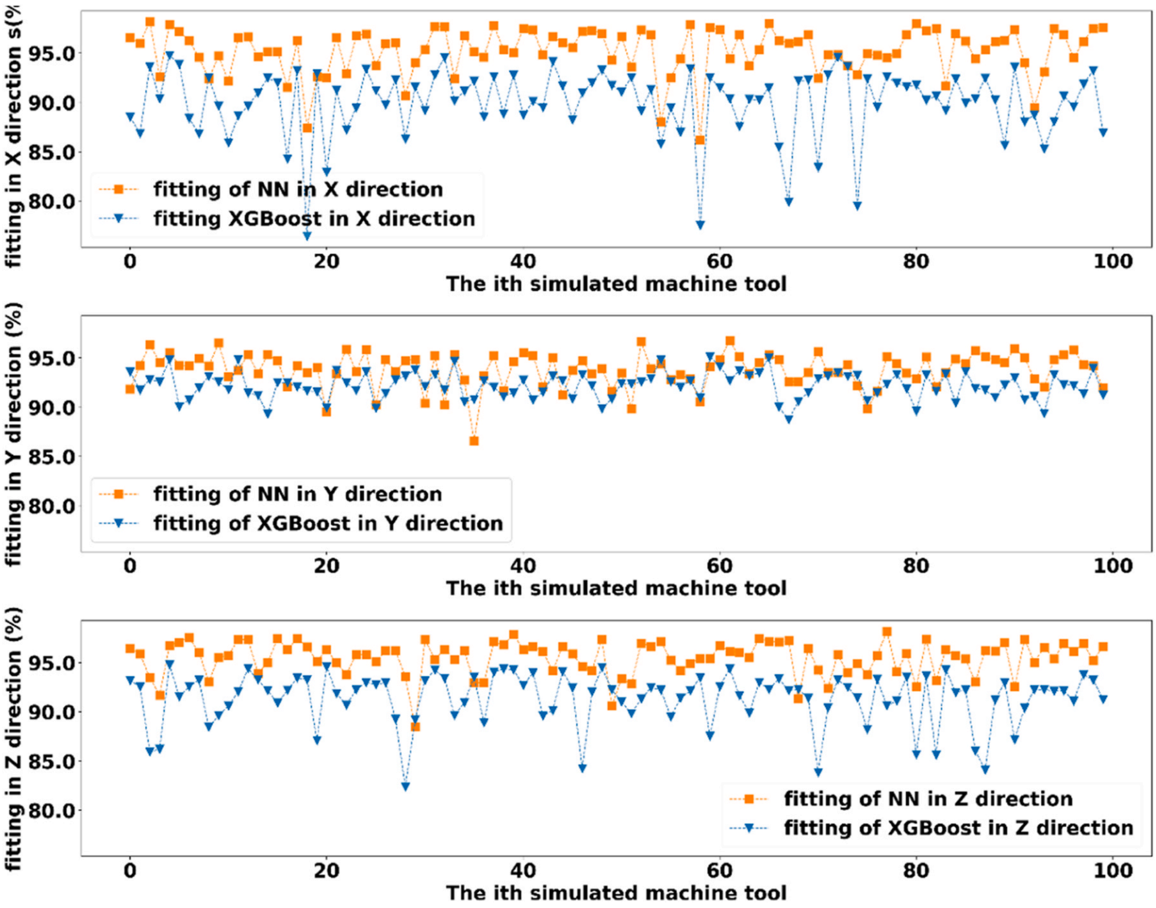

Fig. 18. *fitting* of NN and XGBoost with synthetic data that uses the same measurement strategy as the one to be used for the experimental validation and 100 instances of machine tool's inter- and intra-axis geometric errors.

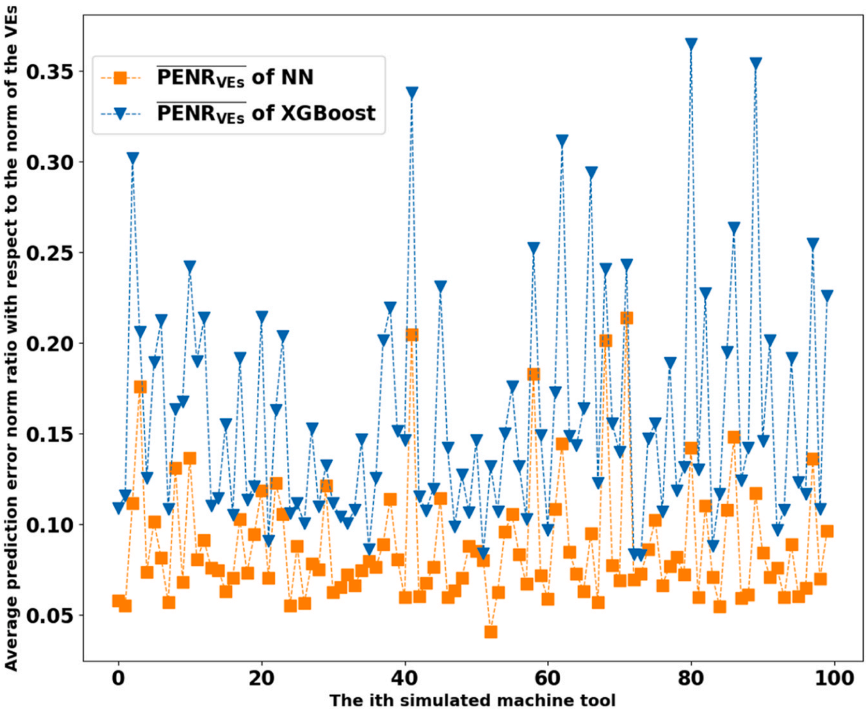

Fig. 19. Average prediction error norm ratio with respect to the norm of the VEs using NN and XGBoost based on synthetic data that uses the same measurement strategy as the one to be used for the experimental validation.

Table 2
Average and standard deviation of different metrics of NN and XGBoost trained with synthetic data that uses randomly generated measurement strategies or the planned experimental strategy, and for all 100 simulated machine error sets.

Average/ Standard deviation of different metrics of NN and XGBoost	Synthetic data that uses randomly generated measurement strategy			Synthetic data that uses the same measurement strategy as the one to be used for the experimental validation		
	Direction					
	X	Y	Z	X	Y	Z
Average *RMSE* of NN (μm)	0.36	0.47	0.41	0.84	0.86	0.95
Average *RMSE* of XGBoost (μm)	0.97	0.73	0.86	1.80	1.10	1.80
RMSE standard deviation of NN (μm)	0.065	0.083	0.047	0.30	0.22	0.26
RMSE standard deviation of XGBoost (μm)	0.240	0.110	0.200	0.52	0.24	0.41
Average *MAE* of NN (μm)	0.28	0.40	0.31	0.68	0.71	0.78
Average *MAE* of XGBoost (μm)	0.76	0.59	0.68	1.50	0.90	1.50
MAE standard deviation of NN (μm)	0.049	0.078	0.036	0.24	0.18	0.21
MAE standard deviation of XGBoost (μm)	0.190	0.100	0.150	0.38	0.17	0.34
Average *fitting* of NN	99.5 %	99.2 %	99.3 %	95.2 %	93.8 %	95.5 %
Average *fitting* of XGBoost	98.6 %	98.8 %	98.6 %	89.9 %	92.2 %	91.4 %
fitting standard deviation of NN	0.09 %	0.14 %	0.08 %	2.4 %	1.8 %	1.7 %
fitting standard deviation of XGBoost	0.36 %	0.20 %	0.32 %	3.5 %	1.4 %	2.7 %
Average of prediction error norm ratio of 100 simulated machine tool of NN	0.04			0.09		
Average of prediction error norm ratio of 100 simulated machine tool of XGBoost	0.08			0.16		
Standard deviation of prediction error norm ratio of 100 simulated machine tool of NN	0.014			0.033		
Standard deviation of prediction error norm ratio of 100 simulated machine tool of XGBoost	0.036			0.063		

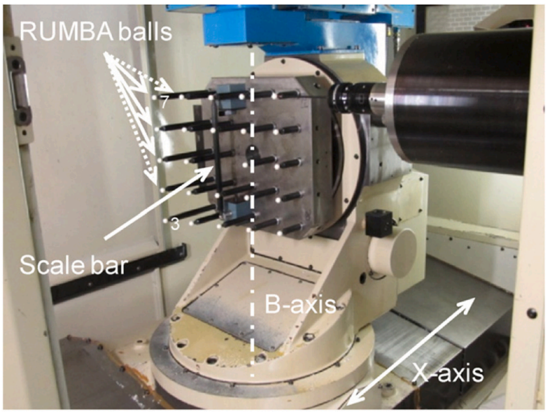

Fig. 20. Scale and master balls artefact (SAMBA) undergoing probing on a wCBXfZY(S)t machine; shown for b = −90°, c = −270°. (reprinted from [26], Copyright (2012), with permission from Elsevier).

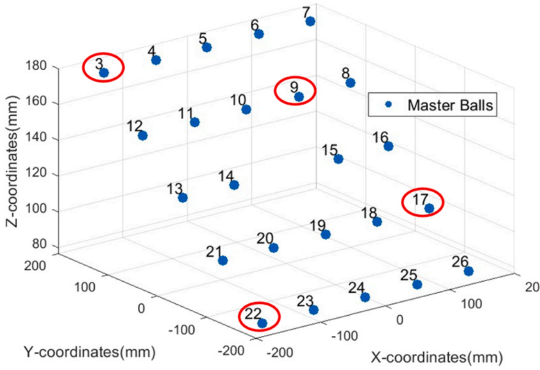

Fig. 21. Nominal positions of 24 master balls in machine table frame when b = 0°, c = 0°; balls in red ellipse are used for testing purpose.

Overall, the performance of both NN and XGBoost trained with synthetic data that uses a randomly generated measurement strategy shows that ML models can be applied to predict VEs for the kinematics of the target five-axis machine tool with different machine error parameters by using axis commands as input. NN slightly outperforms XGBoost.

2.7. Comparison of NN and XGBoost results for synthetic data that uses the same measurement strategy as the one to be used for the experimental validation

A simulation will now be conducted that closely follows the conditions of experimental validation to be presented in Section 3. Twenty-four master balls plus a scale bar are used and eleven (b_i, c_i) pairs. The one-hundred simulated machine tool error instances are re-used as presented in Section 2.3. The process is as illustrated in Fig. 4. The specific balls are selected randomly to divide the data into a training set

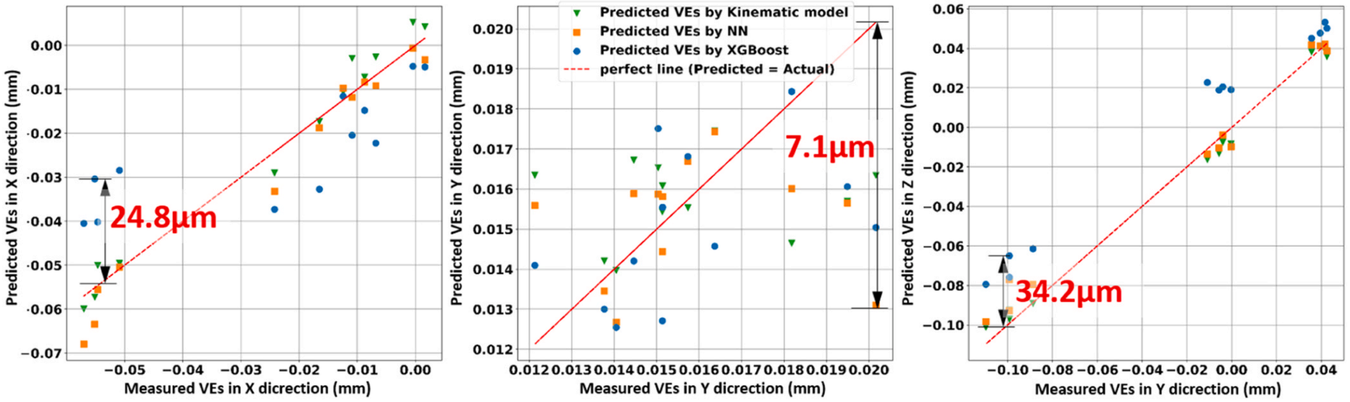

Fig. 22. Comparison of predicted VEs by NN, XGBoost and kinematic model in X; Y; and Z with different plot scale based on 264 experimental data.

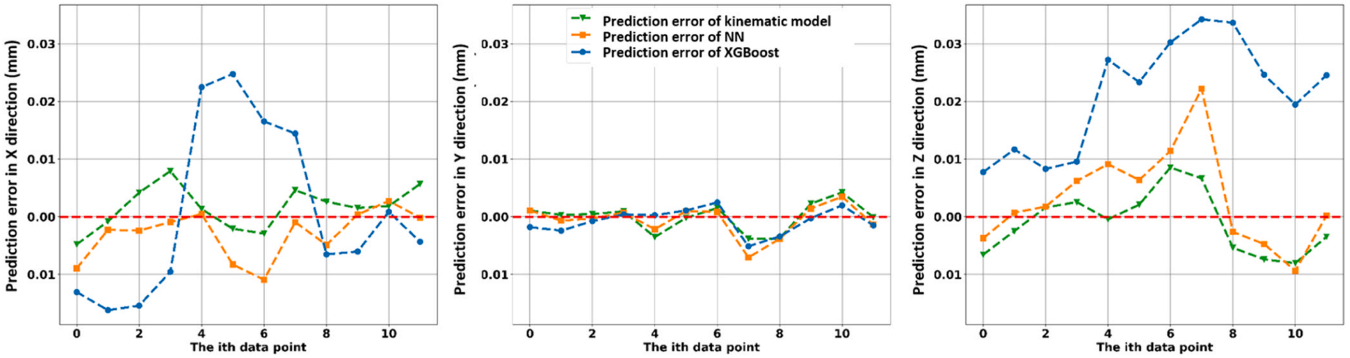

Fig. 23. Prediction error of NN, XGBoost and kinematic model in X, Y; and Z directions based on 264 experimental data.

Table 3
Pearson correlation coefficient of prediction error between ML models and the kinematic model based on 264 experimental data.

	Direction		
	X	Y	Z
Correlation between NN and kinematic model	0.69	0.90	0.89
Correlation between XGBoost and kinematic model	−0.21	0.65	0.24

(with twenty balls) and a testing set (with four balls). Moreover, the eleven (b_i, c_i) pairs are randomly divided into eight (b_i, c_i) pairs for training and three (b_i, c_i) pairs for testing. Predicted VEs by NN and by XGBoost, for synthetic data emulating a real experimental procedure are shown in Fig. 15. The orange and blue dots indicate the VEs predicted by NN and XGBoost, respectively, while the red dotted line represents a perfect prediction. The width of the blue region of the XGBoost predicted VEs is slightly larger than the width of the orange zone of the NN predicted VEs in X and Z direction, which indicates some of the predicted VEs by XGBoost are further away from the perfect prediction than the predicted VEs by NN in those directions. However, in the Y direction, NN and XGBoost perform similarly.

To better evaluate and compare the performance of NN and XGBoost, *RMSE*, *MAE* and *fitting* are calculated and presented in Fig. 16, Fig. 17 and Fig. 18, respectively. All the results show that, NN performs slightly better than XGBoost in the X and Z direction. The performance of these two models is quite close in Y direction.

Fig. 19 shows the average prediction error norm ratio with respect to the norm of the VE (Eq. (8)) of each simulated machine tool error instance using NN and XGBoost. The performance of NN is better than XGBoost with the worst-case prediction error norm ratio of 0.21 and 0.36 respectively.

Overall, by using synthetic data based on the experimental strategy, similar performance, with a slight advantage for NN, is obtained for both NN and XGBoost with the same structure and hyper-parameters used in the previous sections.

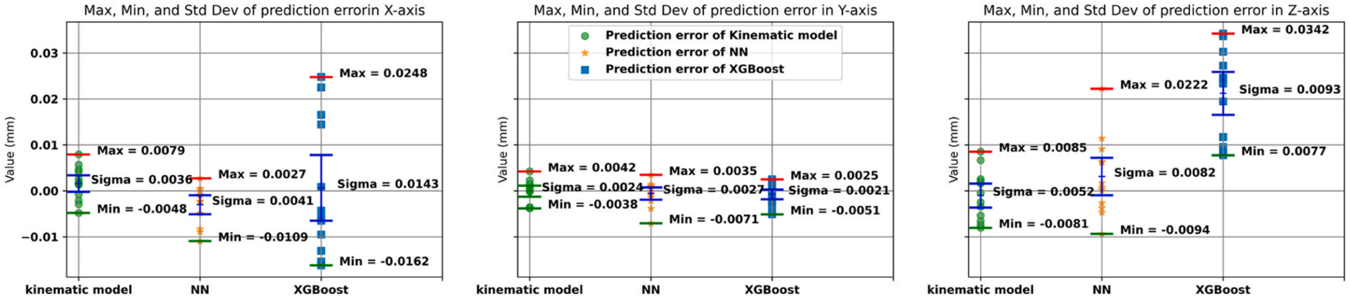

Fig. 24. Plot with Max, Min and Std Dev of prediction error using kinematic model, NN and XGBoost based on 264 experimental data.

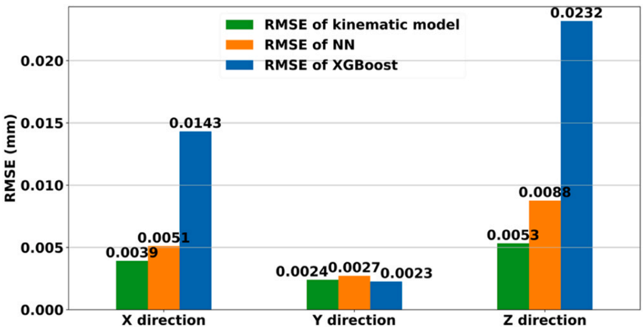

Fig. 25. *RMSE* of kinematic model, NN and XGBoost based on 264 experimental data.

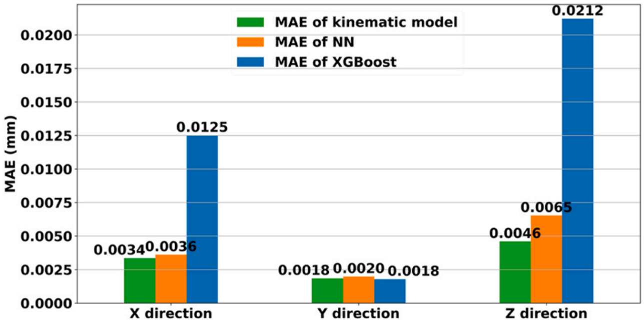

Fig. 26. *MAE* of kinematic model, NN and XGBoost based on 264 experimental data.

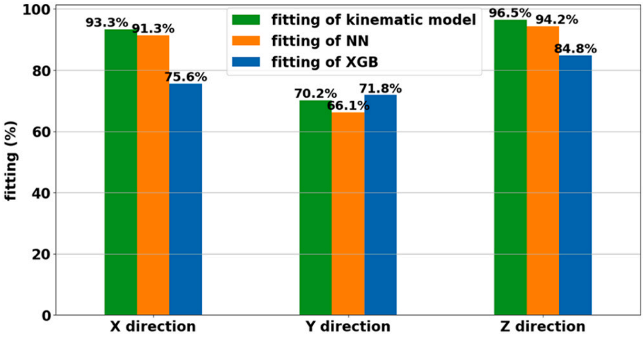

Fig. 27. *fitting* of kinematic model, NN and XGBoost based on 264 experimental data.

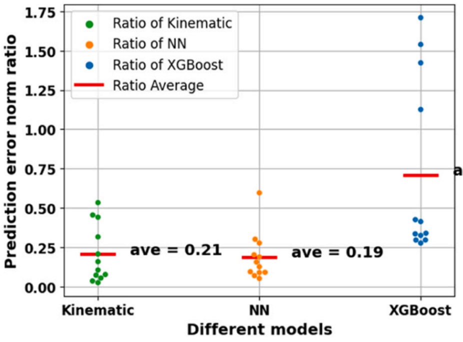

Fig. 28. *$PENR_{VE}$* of kinematic model, NN and XGBoost based on 264 experimental data.

2.8. Discussion

To better compare the performance of NN and XGBoost trained with synthetic data that uses different measurement strategies, the average and standard deviation of the four metrics are calculated and presented in Table 2.

When NN is trained with synthetic data that uses a randomly generated measurement strategy, the performance of different simulated machine tool error instance is quite similar with a standard deviation of *RMSE, MAE and fitting* less than 0.09 μm, 0.08 μm and 0.14 % in all three directions, respectively, while the standard deviation of *RMSE*, *MAE* and *fitting* of XGBoost is slightly higher than those of NN. Using NN, the performance in Z direction is more stable and less sensitive to variations in the data, while using XGBoost there is a better performance in Y direction. Regarding the standard deviation of prediction error norm ratio of 100 simulated machine tool error instances, there is a better performance and less fluctuation between different simulated machine tool error instances using NN than using XGBoost with averages of 0.04 and 0.08, and standard deviations of 0.014 and 0.036, respectively.

NN trained with the synthetic data that uses the same measurement strategy as the one to be used for the experimental validation, the standard deviation of *RMSE, MAE and fitting* is higher than for the training with synthetic data that uses randomly generated measurement strategy. Using NN, the performance in all directions is quite similar, while using XGBoost there is a better prediction in the Y direction. Regarding the standard deviation of prediction error norm ratio of 100 simulated machine tool error instances, there is more fluctuation of different simulated machine using XGBoost than using NN with the standard deviation of 0.063 and 0.033, respectively.

The performance of both NN and XGBoost trained with synthetic data that uses different measurement strategies shows that ML models can be applied to predict VEs caused by the 13 machine error parameters of the target five-axis machine tool for different machine error parameter values by using axis commands as input. NN slightly outperforms XGBoost for data from both measurement strategies. The predictive performance of NN and XGBoost trained by four balls with 180 (b_i, c_i) pairs is better than that of twenty balls with eight (b_i, c_i) pairs, which indicates that the measurement strategy has an impact on the performance of both NN and XGBoost. Moreover, given that the number of samples in the randomly generated measurement set is 720 while the one in the same measurement strategy as the one to be used for the experimental validation is only 160, we can also conclude that, NN is more sensitive to the number of training samples than XGBoost. For a five-axis machine tool, many of the parametric errors propagate to the tool and workpiece through the kinematic chain. The ML models appear to be able to perform nearly as well as the kinematic error propagation model. In Section 3, a set of experimental data is used to further validate the possibility of using ML models to predict VEs.

3. Experimental validation

A previous experimental investigation of the SAMBA method (Scale and master balls artefact) was conducted on a Mitsui-Seiki HU40T horizontal machining center with wCBXfZY(S)t topology [26]. The experimental setup is shown in Fig. 20, in which, a fixed length magnetic double ball bar, and a 24 ceramic balls reconfigurable uncalibrated master ball artefacts (RUMBA) with different nominal lengths of 75, 100, 125, 150 and 175 mm, are mounted on the machine pallet at only nominally known positions. A Renishaw® MP 700 touch-trigger probe with a 100 mm stylus is installed in the spindle. During the experiment, all 24 balls were measured by probing eleven (b_i, c_i) pairs. The ball bar is measured once horizontally, parallel to the machine X-axis, at b=c= 0°, and once vertically, parallel to the machine Y-axis, at b= 0° and c= 90°.

A total of 272 ball center measurements were obtained, of which eight outliers were removed leaving 264 samples for processing. The dataset consists of VEs at different positions throughout the machine

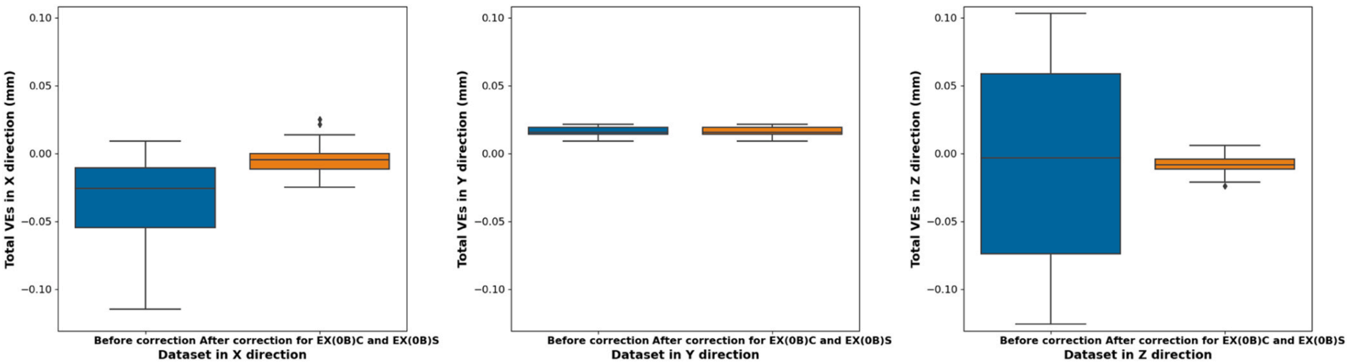

Fig. 29. Comparison of total VEs before (left most and blue on each plot) and after correction for EX(0B)C and EX(0B)S.

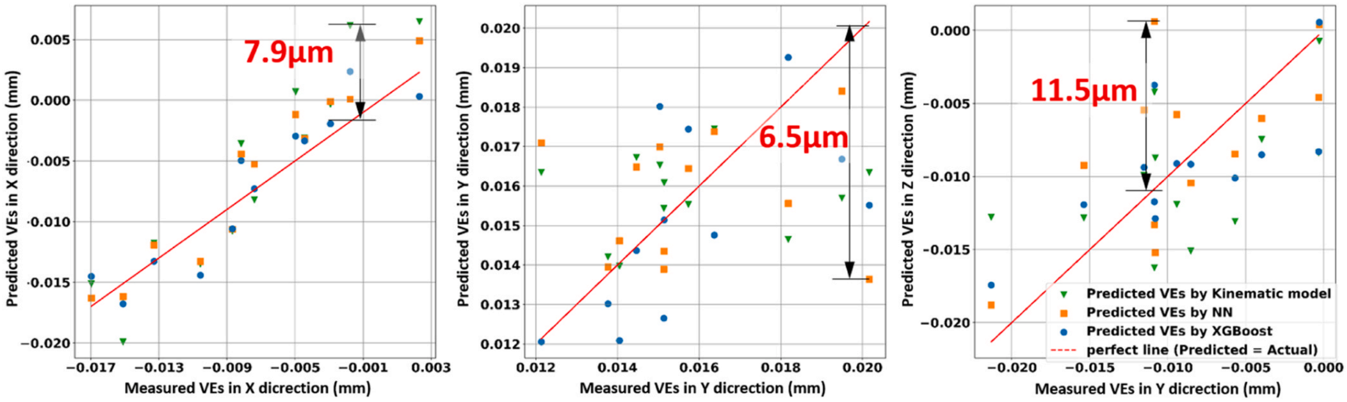

Fig. 30. Comparison of predicted VEs by NN, XGBoost and kinematic model in X, Y; and Z directions with different plot scale based on 264 experimental data after correction for EX(0B)C and EX(0B)S.

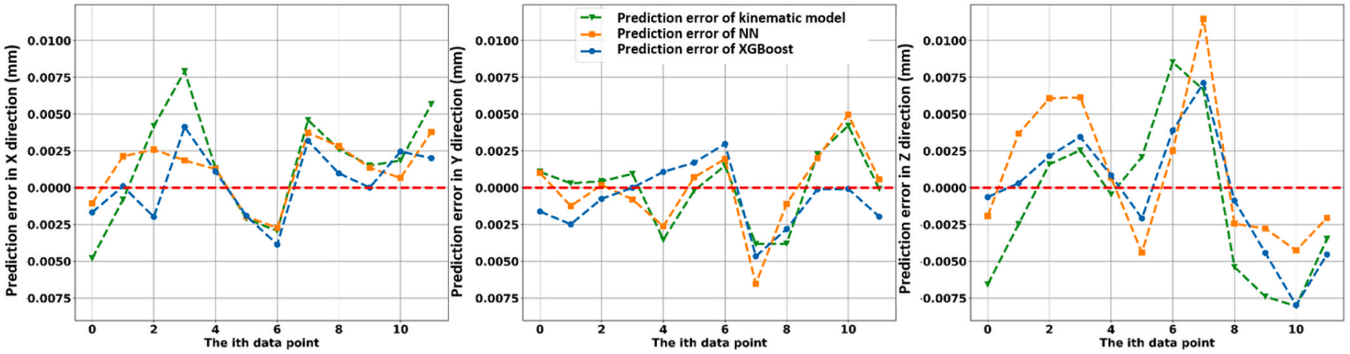

Fig. 31. Prediction error of NN, XGBoost and kinematic model in X, Y and Z directions using 264 experimental data after correction for EX(0B)C and EX(0B)S.

Table 4
Pearson correlation coefficient of prediction error between ML models and kinematic model based on 264 experimental data after correcting EX(0B)C and EX (0B)S.

	Direction		
	X	Y	Z
Correlation between NN and kinematic model	0.79	0.86	0.69
Correlation between XGBoost and kinematic model	0.76	0.43	0.82

work envelop, corresponding to different X-, Y-, Z-, B- and C-axis positions that corresponds to the stylus tip of the touch probe being virtually positioned at the center of the ball.

The same structure of NN and XGBoost, with the same parameters and hyper-parameters used in Section 2, are evaluated using the experimental datasets. Data from randomly selected master balls No. 3, 9, 17 and 22 (Fig. 21), with three (b_i, c_i) pairs is used to test the performance of the two ML models. Data from the two scale bar balls, ball No. 26 when only the spindle is rotated, and all of the remaining twenty balls, with the other eight (b_i, c_i) pairs are used for training.

The machine tool used in this study suffers from large EX(0B)C and EX(0B)S errors of the order of 100 µm. In order to avoid having two error sources dominate the others and so potentially unfairly improving the performance of the ML models, models are evaluated based on the experimental data and re-evaluated after correcting the data for the VEs due to EX(0B)C and EX(0B)S. The results are presented in Sections 3.1 and 3.2 respectively.

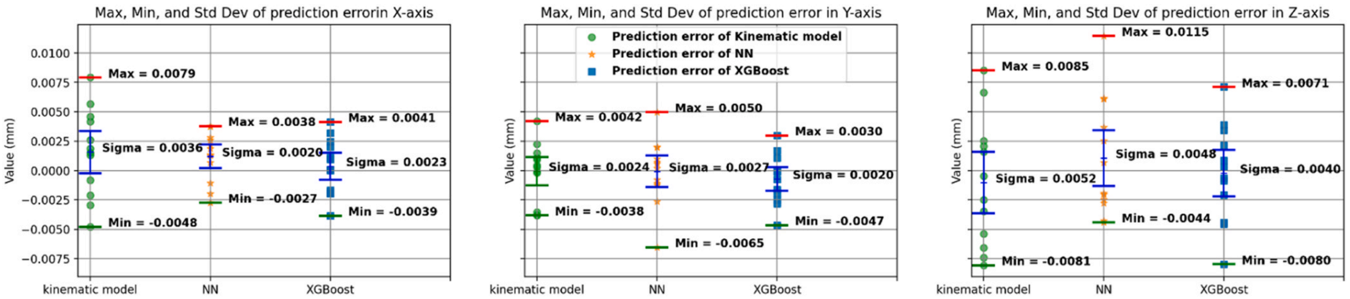

Fig. 32. Plot with Max, Min and Std Dev of prediction error using kinematic model, NN and XGBoost based on 264 experimental data after correction for EX(0B)C and EX(0B)S.

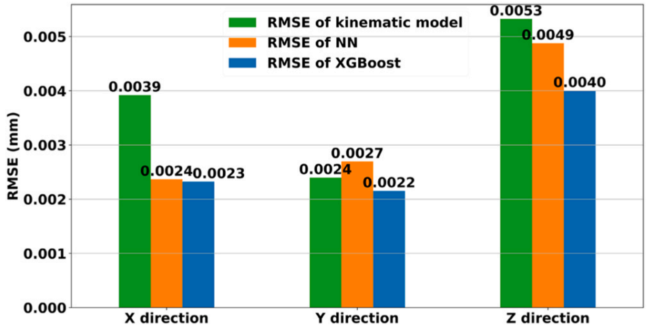

Fig. 33. *RMSE* of kinematic model, NN and XGBoost based on 264 experimental data after correction for EX(0B)C and EX(0B)S.

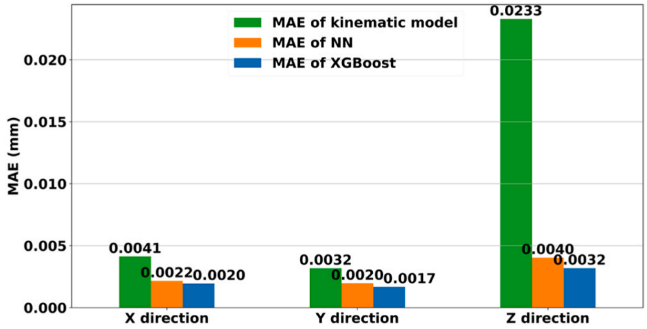

Fig. 34. *MAE* of kinematic model, NN and XGBoost based on 264 experimental data after correction for EX(0B)C and EX(0B)S.

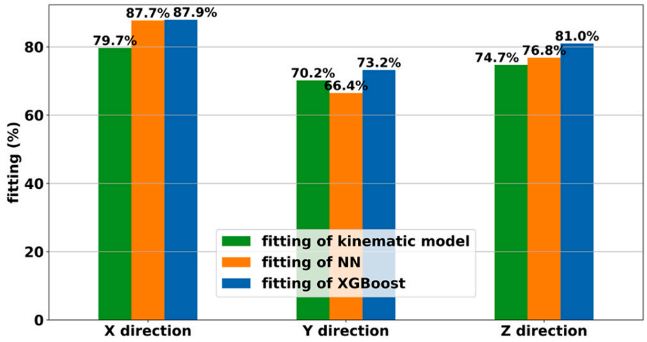

Fig. 35. *fitting* of kinematic model, NN and XGBoost based on 264 experimental data after correction for EX(0B)C and EX(0B)S.

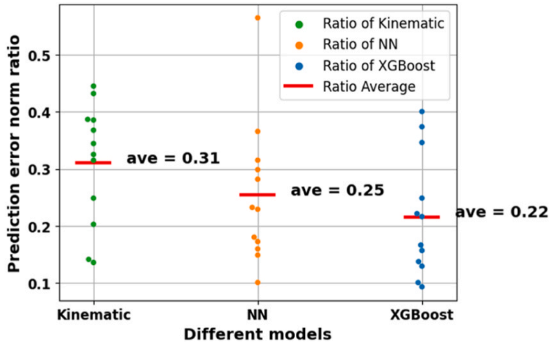

Fig. 36. Prediction error norm ratio of testing data for kinematic model, NN and XGBoost based on 264 experimental data after correction for EX(0B)C and EX(0B)S.

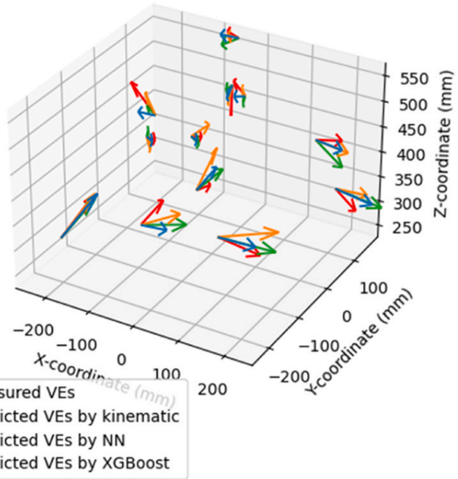

Fig. 37. Vectors of predicted VEs by different models based on 264 experimental data after correction for EX(0B)C and EX(0B)S.

3.1. Comparison results using NN and XGBoost based on 264 experimental data

Fig. 22 and Fig. 23 compare the predicted VEs and prediction error obtained using the kinematic model, NN and XGBoost. The graphs show that even with a limited number of training samples all models predict the VEs in X, Y and Z meaningfully.

Fig. 23 shows that the prediction error of the kinematic, NN and XGBoost models have a similar trend in Y and Z, which suggests that the predictive capacity of the three models is quite similar in those two directions. However, although the prediction error results in X are of the

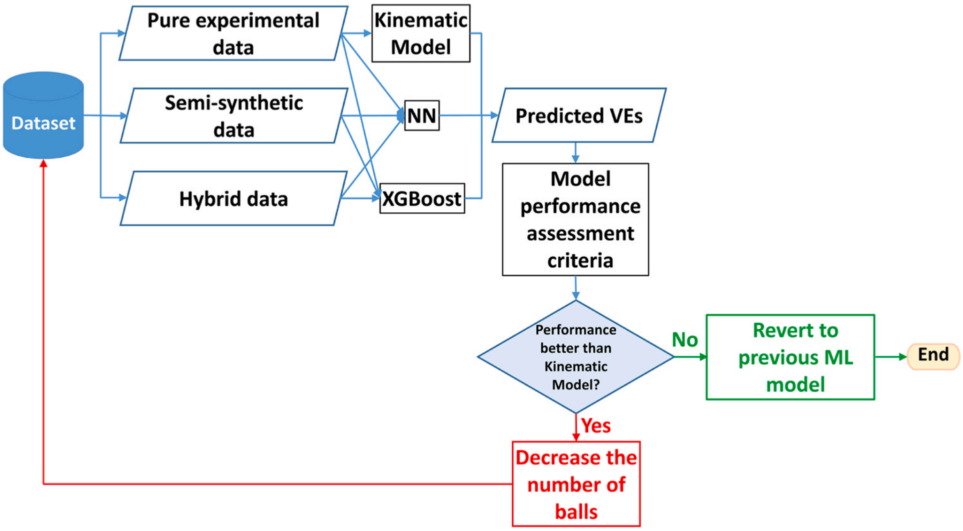
Dataset
Pure experimental data
Semi-synthetic data
Hybrid data
Kinematic Model
NN
XGBoost
Predicted VEs
Model performance assessment criteria
Performance better than Kinematic Model?
No
Revert to previous ML model
End
Yes
Decrease the number of balls

Fig. 38. Procedure to determine the number of balls needed for training to achieve better performance than the kinematic model.

Table 5
Comparison of computational resources and time consumption of different ML models during training process.

Machine learning models	NN	XGBoost
Training size	161	161
Testing size	12	12
Training time (s)	40	1
Memory usage (MB) during training	425.52(peak)	370.61
CPU usage (%) during training	120.1(peak)	451.6
GPU usage (%) during training	22(peak)	1
Predicting time (s)	0.04	0.002

same order of magnitude as for Y and Z, they show much less correlation between the kinematic and XGBoost, since there is no clear resemblance between the prediction error of the kinematic and XGBoost in that direction. In order to quantify the similarity of the prediction error between NN and kinematic, and XGBoost and kinematic the respective Pearson correlation coefficient is listed in Table 3. The calculated Pearson correlation coefficient indicates a lower correlation in the X direction.

Fig. 24 presents the maximum, minimum and standard deviation of the prediction error using the kinematic, NN and XGBoost models. The standard deviations of the prediction error using kinematic model are the smallest in all X and Z directions. The prediction error range (Eq. (9)) using XGBoost is smaller than using the kinematic and NN models in Y direction.

$$\boldsymbol{range} = \quad \boldsymbol{error}_{\boldsymbol{max}} - \boldsymbol{error}_{\boldsymbol{min}} \tag{9}$$

where $error_{\max}$ is the maximum of the prediction error, $error_{\min}$ is the minimum of the prediction error.

Based on the assessment criteria: *RMSE* (Fig. 25), *MAE* (Fig. 26), *fitting* (Fig. 27), and $\overline{PENR}_{VE}$ (Fig. 28), NN performed as good as kinematic model, with similar *RMSE, MAE,* $\overline{PENR}_{VE}$ and *fitting*. According to the $\overline{PENR}_{VE}$ results, XGBoost performed the worst. However, the performance of XGBoost in Y direction is the best.

These results suggest that NN trained with 264 experimental data is a promising ML model for predicting VEs based on axis commands.

3.2. Comparison results using NN and XGBoost based on 264 experimental data after correcting EX(0B)C and EX(0B)S

The total VEs before and after correcting the EX(0B)C and EX(0B)S effects using simple trigonometric functions is shown in Fig. 29. EX(0B)C

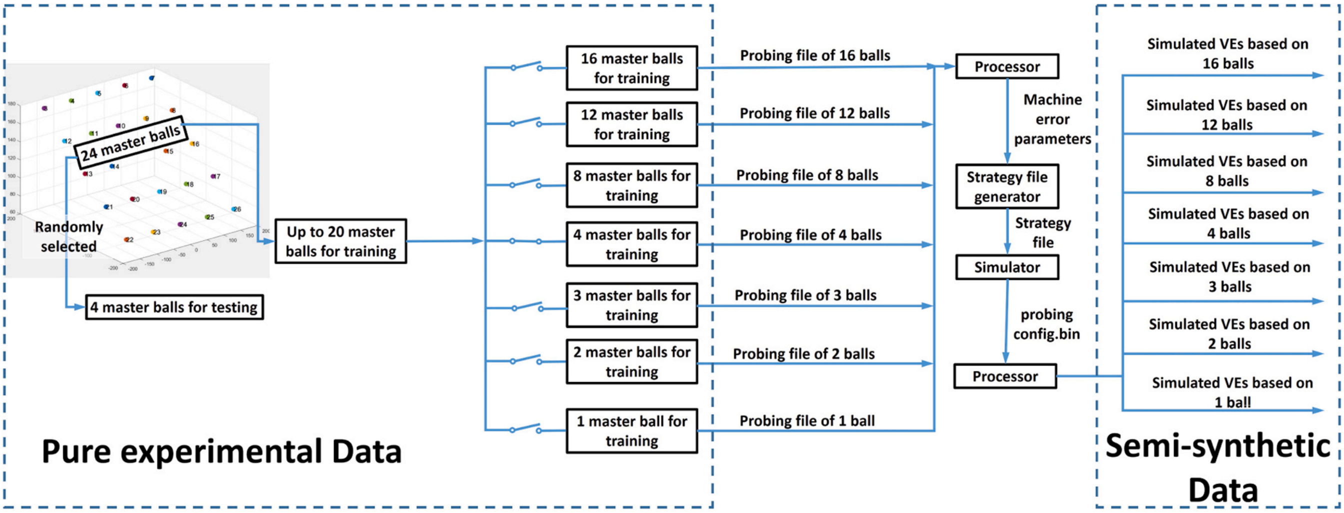
24 master balls
Randomly selected
4 master balls for testing
Up to 20 master balls for training
16 master balls for training
12 master balls for training
8 master balls for training
4 master balls for training
3 master balls for training
2 master balls for training
1 master ball for training
Pure experimental Data
Probing file of 16 balls
Probing file of 12 balls
Probing file of 8 balls
Probing file of 4 balls
Probing file of 3 balls
Probing file of 2 balls
Probing file of 1 ball
Processor
Machine error parameters
Strategy file generator
Strategy file
Simulator
probing config.bin
Processor
Simulated VEs based on 16 balls
Simulated VEs based on 12 balls
Simulated VEs based on 8 balls
Simulated VEs based on 4 balls
Simulated VEs based on 3 balls
Simulated VEs based on 2 balls
Simulated VEs based on 1 ball
Semi-synthetic Data

Fig. 39. Process of generating different types of datasets.

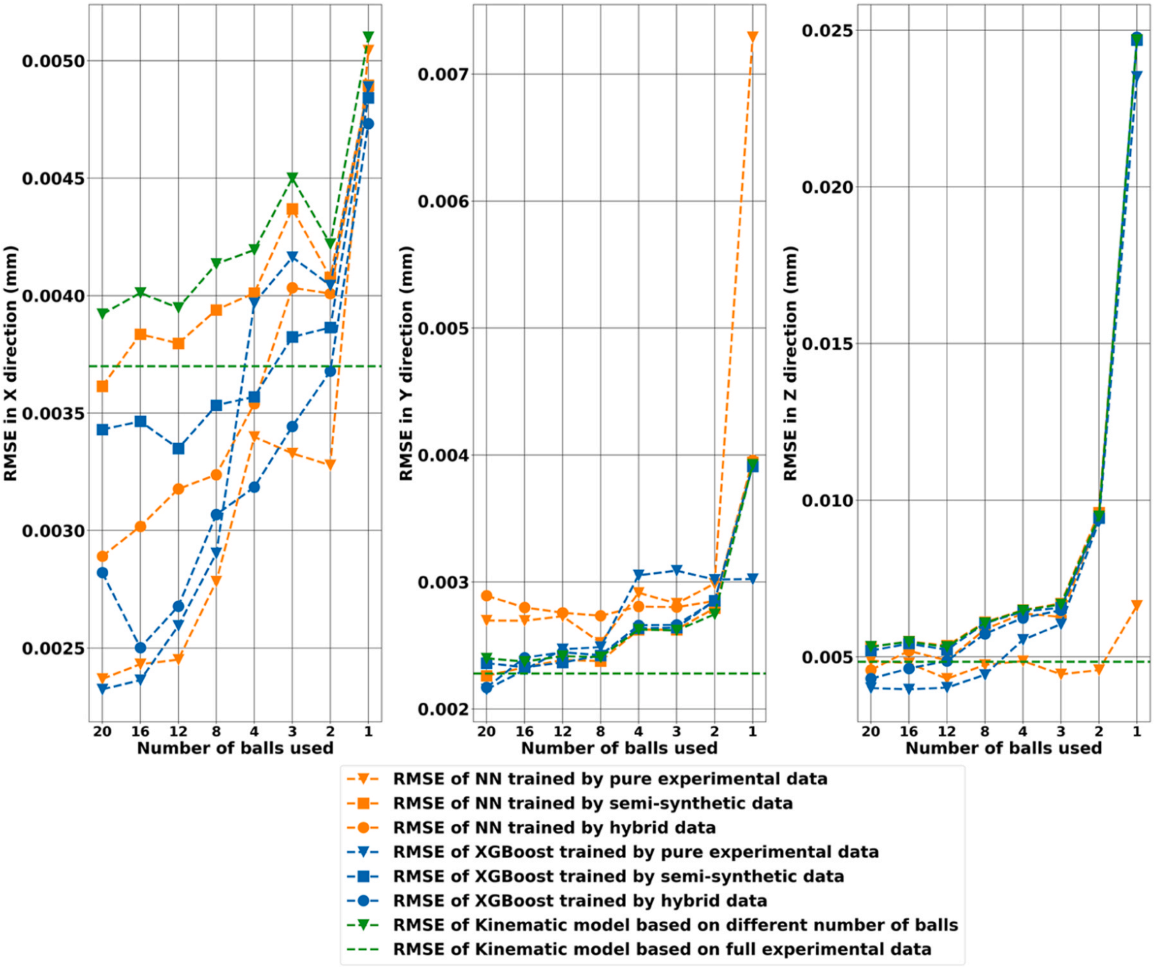

Fig. 40. *RMSE* of different models trained by different types of data based on experimental data after correction for EX(0B)C and EX(0B)S with different numbers of balls (from sixteen, twelve, eight, four, three, two balls, and one ball) of up to eight (b_i, c_i) pairs.

is the lack of concurrency, or offset distance in the X direction, between the B- and C- axes, which causes VEs in the x-z plane as can be seen very clearly from Fig. 29. EX(0B)S describes a similar offset distance of the spindle relative to the B-axis, also in the X direction, which will only affect the VEs in X direction. The total VEs in the Y direction are expectedly unchanged after the correction due to the nature of these two geometric errors, including the range and the distribution. However, the range and the spread of VEs in both X, and Z direction, decrease dramatically, especially in the Z direction. In addition, before the correction, the VEs in X direction are skewed slightly in the negative direction. After the correction, the total VEs are balanced in both positive and negative directions.

Fig. 30 shows the predicted VEs using the different models. The VEs predicted by XGBoost are slightly better than for the kinematic and the NN models. Although many VEs predicted by the three models appear relatively far from the actual values, the prediction errors are smaller than those before correction, especially for the X and Z directions.

The prediction errors of the different models after correction are shown in Fig. 31. Different to the results obtained by models trained before correcting the EX(0B)C and EX(0B)S, where XGBoost performs the worst, the prediction error of XGBoost trained after correction becomes the smallest in the X and Z directions. The prediction error of NN and kinematic model in all three directions show similar trends, as the peaks and throughs of orange and green colors correspond to each other. To quantify the similarity of the prediction error between NN and kinematic, and XGBoost and kinematic, Pearson correlation coefficient (Eq. (9)) is calculated and showed in Table 4. The calculated Pearson correlation coefficient indicates a lesser correlation in the Y direction when using XGBoost to predict VEs.

Fig. 32 shows the maximum, minimum and standard deviation of prediction error for the three models. For standard deviation, XGBoost is the smallest in both the Y and Z directions, with values of 2.0 and 4.0 μm. The range (Eq. (9)) of prediction error using XGBoost is also the smallest in those two directions.

According to the results based on assessment criteria *RMSE* (Fig. 33) *MAE* (Fig. 34) and *fitting* (Fig. 35)), XGBoost performs the best with the lowest *RMSE*, *MAE* and the highest *fitting*.

Although NN does not perform better than XGBoost, in this situation, with respect to *RMSE*, *MAE*, and *fitting*, it performs better than kinematic model, except for *RMSE* and *fitting*, in the Y direction. It indicates that both NN and XGBoost are better able to fine tune when there are no dominant errors with XGBoost performing slightly better.

All the prediction error norm ratios (Eq. (8)) of XGBoost are less than 0.5, and nearly 75 % of them are smaller than 0.25. For both kinematic and NN, the ratio is slightly higher than XGBoost (Fig. 36). The average prediction error norm ratio of XGBoost is 0.22, which is smaller than NN of 0.25 and kinematic of 0.31 (Fig. 36).

The vectors of VEs predicted by different models (Fig. 37) show that although the vectors are not exactly in the same direction, they are similar. The lengths of vectors of VEs predicted by XGBoost are closest to the measured ones. In most cases, the lengths of VE vectors predicted by NN are longer than those of measured VE vectors.

To further study the efficiency of different ML models, we analyze the computational resources and time consumption of NN and XGBoost

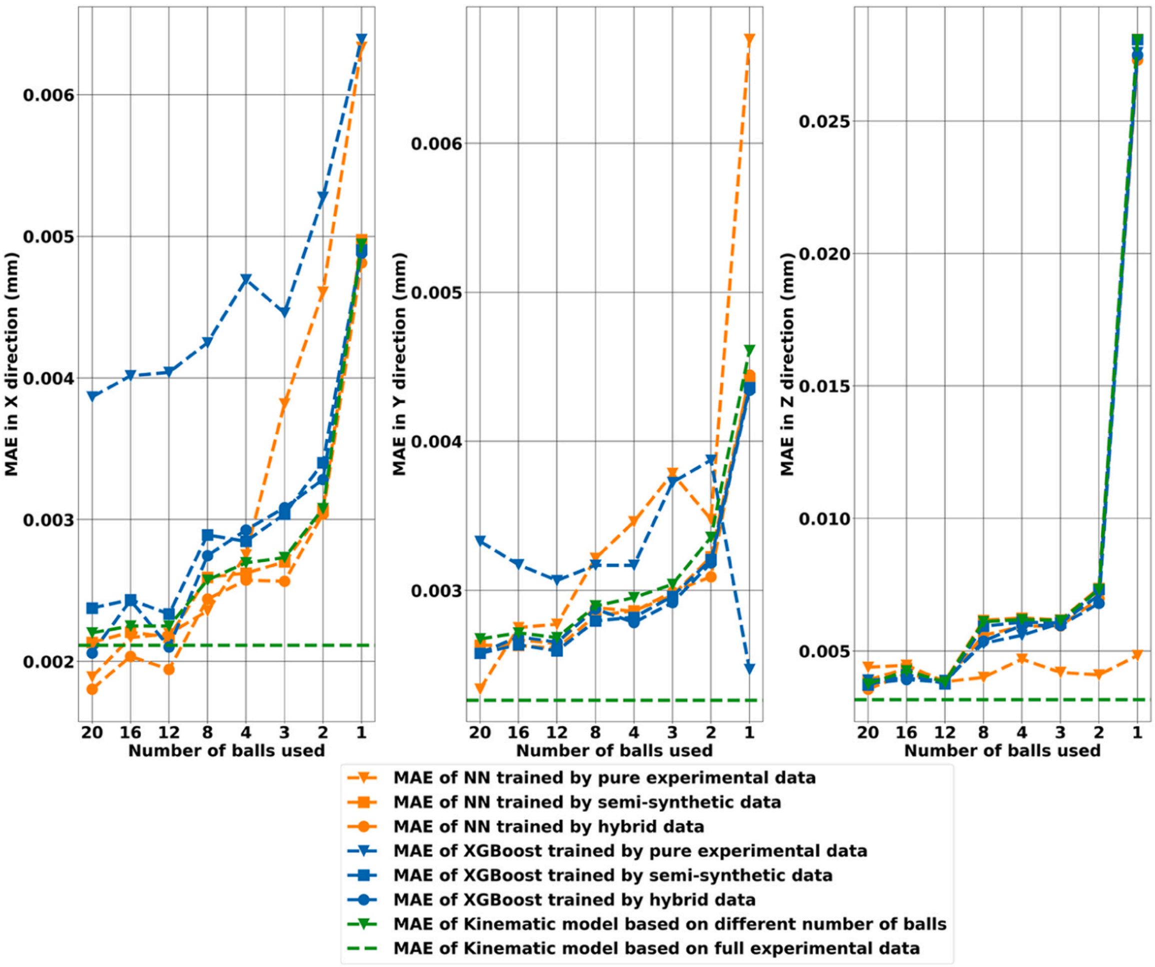

Fig. 41. *MAE* of different models trained by different types of data based on experimental data after correction for EX(0B)C and EX(0B)S with different numbers of balls (from sixteen, twelve, eight, four, three, two balls, and one ball) of up to eight (b_i, c_i) pairs.

when use 161 experimental data to train (plotted in Table 5). The results show that XGBoost trains 40 times faster than NN, and could predict 20 times faster. Training with XGBoost uses less memory, GPU, while with very high CPU usage.

Based on all the results we have discussed, both NN and XGBoost are able to predict VEs by using axis commands as inputs. Moreover, XGBoost trained with the measurement data after correcting for the dominant EX(0B)C and EX(0B)S error parameter has a better performance than model trained with the measurement data before correcting.

4. Selection of the best type of data with the minimum number of balls for training

From an industrial perspective, VEs cannot currently be measured while machining and so the minimum number of VEs measurements to achieve good results is sought for productivity reasons. Although based on simulation, the predictive performance of XGBoost trained with data from twenty balls with eleven (b_i c_i) pairs is close to that of four balls with 180 (b_i c_i) pairs, conducting a 24-ball SAMBA test (in which 20 balls are for training, and 4 balls are for testing) to collect data is very time consuming as each ball position measurements takes between 30 s and one minute. On the other hand, it is also worth investigating whether synthesized data can help improve performance of ML models. Fig. 37 illustrates the procedure for determining the number of balls of different data type needed to generate training data to achieve a better performance than the kinematic model.

4.1. Different types of data

Fig. 39 illustrates the process to generate the three types of datasets, which are only (pure) experimental data after correcting the EX(0B)C and EX(0B)S, semi-synthetic data and hybrid data.

1. Only (pure) experimental data after correcting the EX(0B)C and EX(0B)S

   Axis commands and translational VEs, as 3D vectors, of randomly selected ball No. 3, 9, 17 and 22 (Fig. 21) with three (b_i c_i) pairs are test data. The remaining twenty master balls with eight (b_i c_i) pairs plus one scale bar dataset and an additional master ball (when only the spindle rotates) are used for training purpose. However, to find the minimum number of balls for training, we randomly select the amount of balls in a decreasing strategy: from sixteen, twelve, eight, four, three, two balls, and one ball. For each quantity of balls, we randomly select five different ball combinations and then average the values of each metric to mimic a five-fold cross-validation.
2. Semi-synthetic data

   The semi-synthetic data is generated using a kinematic model with error parameters estimated using the experimental data, corrected to remove the effect of EX(0B)C and EX(0B)S. The estimation is repeated with decreasing number of balls. Thirteen machine kinematic error parameters are estimated. Then based on these machine error parameters, semi-synthetic data is generated kinematically according to the process shown in Fig. 39.
3. Hybrid data

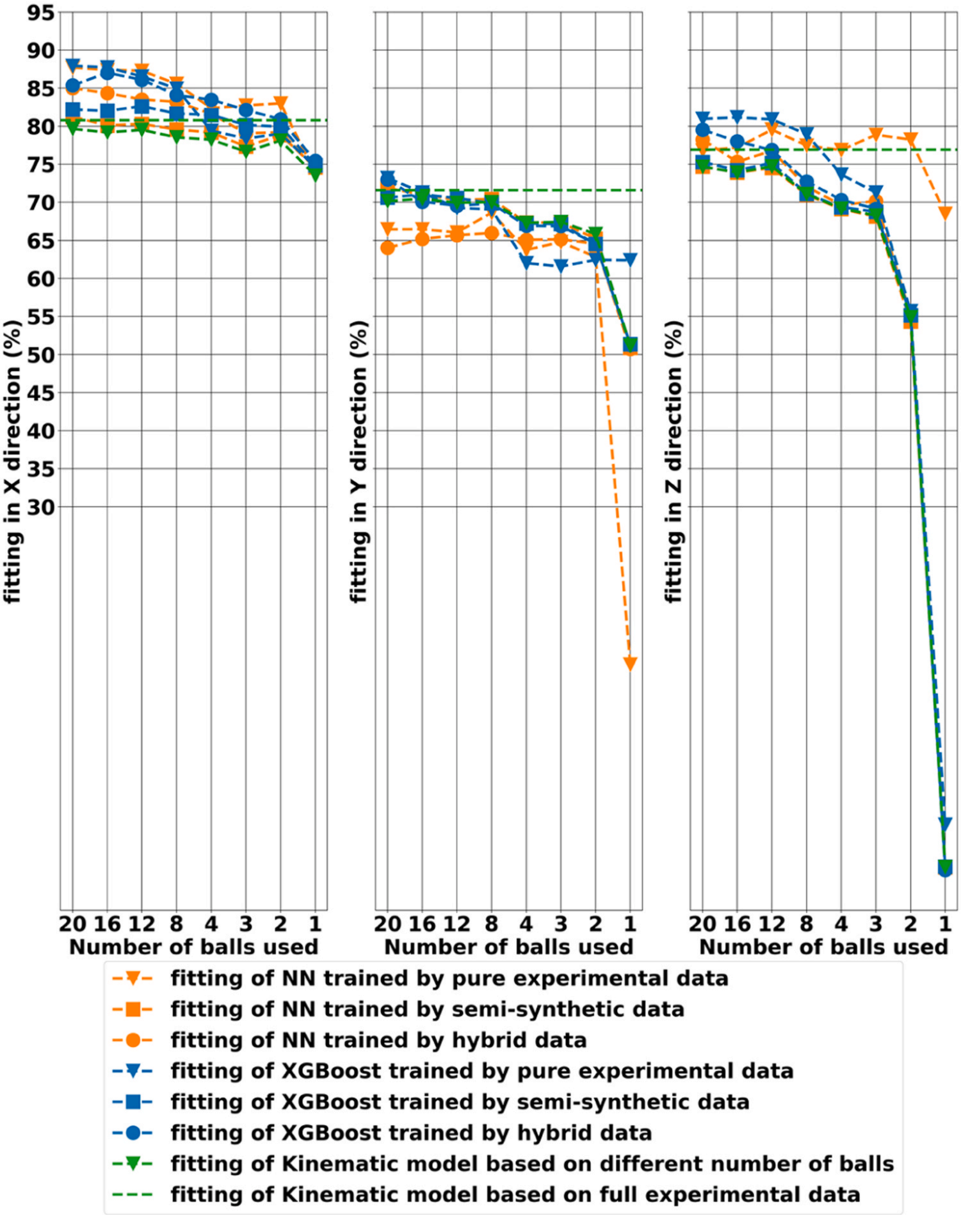

Fig. 42. *fitting* of different models trained by different types of data based on experimental data after correction for EX(0B)C and EX(0B)S with different numbers of balls (from sixteen, twelve, eight, four, three, two balls, and one ball) of up to eight (b_i, c_i) pairs.

We combine the pure experimental data, after correcting the EX(0B)C and EX(0B)S, with the corresponding semi-synthetic data to generate the hybrid data.

NN and XGBoost are trained using the different types of data. The performances of the model and training dataset combinations are evaluated using the same test dataset and are compared to each other and to the kinematic model. If any model trained by any type of data performs better than the kinematic model, the number of balls is reduced, all three types of datasets are regenerated, and training and testing are performed, until the performance is worse than the kinematic model. The whole process will then stop, and the number of balls used in the previous step recorded as the minimum number of balls used for training (the results are explained in Section 4.2). After comparing the performance of each model trained using the minimum number of balls, the best model will be identified for further application.

4.2. Comparison of different types of training datasets based on different number of balls

Based on the process proposed in Fig. 38, the 264 experimental data, after correction for EX(0B)C and EX(0B)S, from up to twenty balls with up to eight (b_i, c_i) pairs plus one scale bar dataset and an additional master ball (when only the spindle rotates) is used for training, and the remaining four balls used as test data with the other three (b_i, c_i) pairs to assess the performance of XGBoost and NN. The number of balls for training is gradually reduced until the performance using different models is worse than that of the kinematic model. The performances of kinematic, NN and XGBoost models are shown in Fig. 40 to Fig. 42. Each colour/symbol combination corresponds to a specific model/dataset combination. For all three models, the *RMSE* (Fig. 40) and *MAE* (Fig. 41) overall increase, and the *fitting* (Fig. 42) overall decreases as the number of training balls decreases, except for the performance of XGBoost trained by only experimental data in the Y direction. When the number of training balls is reduced from two to one, the *RMSE* and *MAE* increase

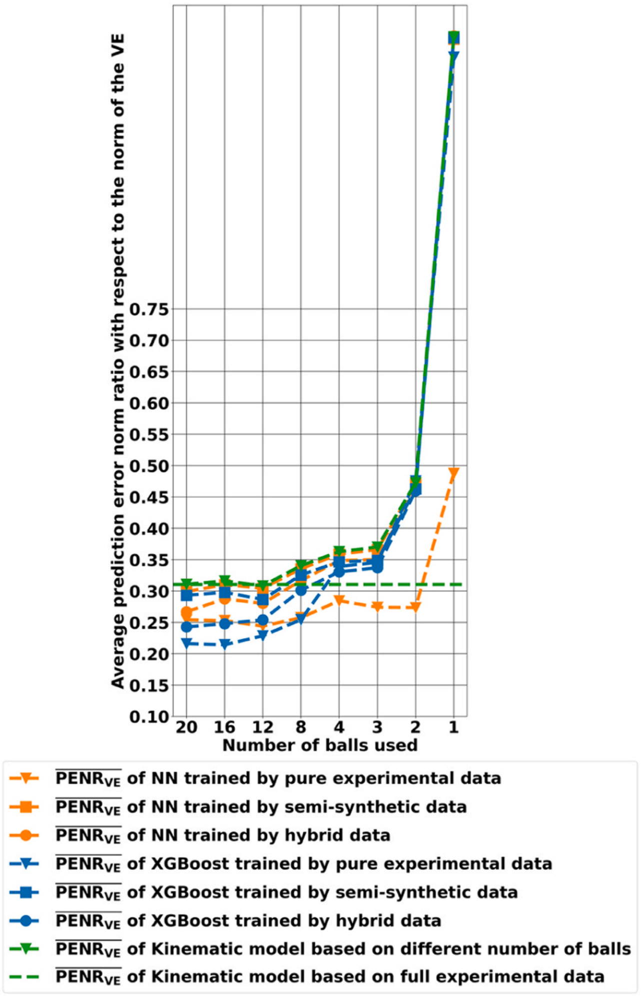

Fig. 43. Average prediction error norm ratio with respect to the norm of the VEs of different models trained by different types of data based on experimental data after correction for EX(0B)C and EX(0B)S with different numbers of balls (from sixteen, twelve, eight, four, three, two balls, and one ball) of up to eight (b_i, c_i) pairs.

rapidly, and the fitting decreases sharply.

From *RMSE* and *fitting*, the performance of NN and XGBoost trained with only experimental or hybrid data is better than that of NN and XGBoost trained with semi-synthetic data and kinematic models in the X and Z directions, suggesting that ML models can model errors not accounted for by the kinematic model, except for those in the Y direction.

The performances of the kinematic, NN and XGBoost models trained with semi-synthetic data are very similar. This makes sense because semi-synthetic data are generated purely from estimated kinematic errors thus filtering out other error sources. It also means that NN and XGBoost can model the VE impact of such error sources given sufficient data.

In addition, there are not many improvements in using hybrid data. This is assumed to be because when using hybrid data, there is a competition between satisfying the pure data with all errors included in it and the semi-synthetic data which are filtered to only account for modelled kinematic error sources. Moreover, adding semi-synthetic data to pure data means developing a kinematic model, which is not desirable as it requires extra time and expertise.

None of the machine learning models trained on any type of data consistently outperformes the kinematic model in all three directions, and machine learning models are not sensitive to different types of training data. Each model demonstrates strengths in some directions but fails to demonstrate superior performance in all cases.

Results show, as suspected, that ML model performance improves with the increase of the number of balls in the training sets, as seen from the average prediction error norm to VE norm ratio decreasing (Fig. 43). Using only experimental data based on two balls to train NN, the average prediction error norm ratio is 0.27 (Fig. 43). So, for the machine and experimental data used, a good compromise between accuracy and experiment time is to use only experimental data of two balls with eight (b_i, c_i) pairs plus one scale bar dataset and an additional master ball (when only the spindle rotates) to train NN to get a more accurate prediction compared to the kinematic model.

5. Conclusions

In this study, two hundred sets of synthetic volumetric translational error vector (VE) output data (one hundred for training purpose and one hundred for testing purpose), eight master balls (four for training, and the other four for testing), 180 and 45 (b_i, c_i) rotary axis positioning pairs were used for training and testing, respectively, to evaluate the possibility of applying ML models to predict VEs from axis commands. Based on this synthetic data that simulates a randomly generated measurement strategy, the results show that using only axis commands as input, ML models Neural Networks (NN) and eXtreme Gradient Boosting (XGBoost) are equally able to model the VEs due to the effect of inter-axis kinematic errors and 1st order linear scale errors, for the wCBXfZYt topology five-axis machine tool studied, independently of the specific values of these error parameters, with the worst-case prediction error norm ratio of 0.08 and 0.19, respectively, and overall average error norm ratios of 0.04 and 0.08, respectively.

Simulation of the specific test strategy that is used for the experimental validation indicated that by using twenty master balls but only eight (b_i, c_i) pairs good performance is obtained for both NN and XGBoost with the same structure and hyper-parameters as used in the models trained with synthetic data that uses a randomly generated measurement strategy. For these simulations, the performance of NN is slightly better than that of XGBoost in all directions, with the worst-case prediction error norm ratio of 0.21 and 0.36 respectively, and overall averages of the ratio of 0.09 and 0.16, respectively.

Experimental results show that, after correcting the data for the effect of two dominant geometric errors (EX(0B)C and EX(0B)S), both NN and XGBoost perform generally better than the kinematic model in predicting the VEs with twenty balls and eight (b_i, c_i) pairs for training and four balls with three (b_i, c_i) pairs for testing, except for *RMSE* and *fitting* of NN, in the Y direction. One potential explanation is that the ML models can model error sources that the kinematic model did not include. Moreover, XGBoost performs better than NN. The study conducted to determine the minimum test conditions and data to achieve good results showed that using two master balls to train NN with just pure experimental data could obtain the average prediction error norm ratio of 0.27. Enriching the experimental data with semi-synthetic data based on estimated kinematic errors did not improve the VE prediction performance by the ML. Thus, it is suggested that an efficient way is using only experimental data of two master balls with eight (b_i, c_i) pairs, plus one scale bar dataset and an additional master ball (when only the spindle rotates).

Limitation and future works

The small size of the dataset limits our ability to reliably evaluate the performance of the machine learning models against the kinematic models. In order to draw more accurate and reliable conclusions, we plan on using a larger dataset such as can be gathered using a sensor nests of three non-contact proximity sensor, as in [28] to further validate the generalization performance of NN and XGBoost.

In the future, the existing model should be improved to make it more efficient and applicable to more uncertain problems, such as predicting positioning uncertainty and thermal deformation.

CRediT authorship contribution statement

Bitar-Nehme Elie: Methodology, Conceptualization. **Mayer J. R. R.:** Writing – review & editing, Software, Resources, Methodology, Investigation, Funding acquisition, Conceptualization. **Duong Xuan Truong:** Conceptualization. **Feng Miao:** Formal analysis, Data curation. **Zeng Min:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: J. R.R. Mayer reports financial support was provided by Natural Sciences and Engineering Research Council of Canada. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank technicians Guy Gironne and Vincent Mayer for their help with the experimental work. Authors’ research is funded by NSERC’s Discovery grant RGPIN-2022-04092.

References

[1] Li J, Mei B, Shuai C, Liu X-j, Liu D. A volumetric positioning error compensation method for five-axis machine tools. Int J Adv Manuf Technol 2019;103:3979–89.

[2] Gao W, Ibaraki S, Donmez MA, Kono D, Mayer JRR, Chen Y-L, et al. Machine tool calibration: Measurement, modeling, and compensation of machine tool errors. Int J Mach Tools Manuf 2023;187:104017.

[3] Yuan J, Ni J. The real-time error compensation technique for CNC machining systems. Mechatronics 1998;8:359–80.

[4] Donmez MA, Blomquist D, Hocken R, Liu C, Barash MM. A general methodology for machine tool accuracy enhancement by error compensation. Precis Eng 1986;8:187–96.

[5] Kim K, Kim M. Volumetric accuracy analysis based on generalized geometric error model in multi-axis machine tools. Mech Mach Theory 1991;26:207–19.

[6] Cong DC, Chinh BB, Jooho H. Volumetric error model for multi-axis machine tools. Procedia Manuf 2015;1:1–11.

[7] Díaz-Tena E, Ugalde U, López de Lacalle LN, de la Iglesia A, Calleja A, Campa FJ. Propagation of assembly errors in multitasking machines by the homogenous matrix method. Int J Adv Manuf Technol 2013;68:149–64.

[8] Olvera D, de Lacalle LNL, Compeán FI, Fz-Valdivielso A, Lamikiz A, Campa FJ. Analysis of the tool tip radial stiffness of turn-milling centers. Int J Adv Manuf Technol 2011;60:883–91.

[9] Iñigo B, Colinas-Armijo N, López de Lacalle LN, Aguirre G. Digital twin-based analysis of volumetric error mapping procedures. Precis Eng 2021;72:823–36.

[10] Xiang S, Altintas Y. Modeling and compensation of volumetric errors for five-axis machine tools. Int J Mach Tools Manuf 2016;101:65–78.

[11] Rahman MM, Mayer JRR. Five axis machine tool volumetric error prediction through an indirect estimation of intra- and inter-axis error parameters by probing facets on a scale enriched uncalibrated indigenous artefact. Precis Eng 2015;40:94–105.

[12] Slamani M, Mayer JRR, Cloutier GM. Modeling and experimental validation of machine tool motion errors using degree optimized polynomial including motion hysteresis. Exp Tech 2011;35:37–44.

[13] Araie I, Akemura O. Method and apparatus for correcting positioning errors on a machine tool. Google Pat 1996.

[14] Fines JM, Agah A. Machine tool positioning error compensation using artificial neural networks. Eng Appl Artif Intell 2008;21:1013–26.

[15] He Z, Yao X, Fu J, Chen Z. Volumetric error prediction and compensation of NC machine tool based on least square support vector machine. Adv Sci Lett 2011;4:2066–70.

[16] Chapman MA. Limitations of laser diagonal measurements. Precis Eng 2003;27:401–6.

[17] Wan A, Song L, Xu J, Liu S, Chen K. Calibration and compensation of machine tool volumetric error using a laser tracker. Int J Mach Tools Manuf 2018;124:126–33.

[18] Nguyen V-H, Le T-T, Truong H-S, Duong HT, Le MV. Predicting volumetric error compensation for five-axis machine tool using machine learning. Int J Comput Integr Manuf 2023:1–28.

[19] Burkov A. The hundred-page machine learning book, 1. City, Can: Andriy Burkov Quebec; 2019.

[20] Chen T, Guestrin C. Xgboost: Reliable large-scale tree boosting system. Proceedings of the 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA. 2015. p. 13–7.

[21] T. Chen, T. He, Higgs boson discovery with boosted trees, in *NIPS 2014 workshop on high-energy physics and machine learning*, 2015, pp. 69–80.

[22] Guo R, Zhao Z, Wang T, Liu G, Zhao J, Gao D. Degradation state recognition of piston pump based on ICEEMDAN and XGBoost. Appl Sci 2020;10:6593.

[23] James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning, 112. Springer; 2013.

[24] Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput Sci 2021;7:e623.

[25] Ngoc HV, Mayer JRR, Bitar-Nehme E. Deep learning to directly predict compensation values of thermally induced volumetric errors. Machines 2023;11:496.

[26] Mayer JRR. Five-axis machine tool calibration by probing a scale enriched reconfigurable uncalibrated master balls artefact. CIRP Ann 2012;61:515–8.

[27] Müller AC, Guido S. Introduction to machine learning with Python: a guide for data scientists. O’Reilly Media, Inc.; 2016.

[28] Zargarbashi SHH, Mayer JRR. Single setup estimation of a five-axis machine tool eight link errors by programmed end point constraint and on the fly measurement with Capball sensor. Int J Mach Tools Manuf 2009;49:759–66.