Titre: Title:	Deep Learning for Machine Tool Thermal Errors
Auteur: Author:	Ngoc Huy Vu
Date:	2023
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Vu, N. H. (2023). Deep Learning for Machine Tool Thermal Errors [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/54189/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/54189/
Directeurs de recherche: Advisors:	J. R. René Mayer, & Elie Bitar-Nehme
Programme: Program:	Génie mécanique

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Deep learning for Machine tool thermal errors

NGOC HUY VU

Département de génie mécanique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiae Doctor*

Génie mécanique

Juillet 2023

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée :

Deep learning for Machine tool thermal errors

présentée par **Ngoc Huy VU**

en vue de l'obtention du diplôme de *Philosophiae Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Luc BARON, président

René MAYER, membre et directeur de recherche

Elie BITAR-NEHME, membre et codirecteur de recherche

Sofiane ACHICHE, membre

Andrew Peter LONGSTAFF, membre externe

DEDICATION

This PhD dissertation is dedicated to my parents Thang Thao, my wife Linh, my sons Long Minh, my brother Hoang.

Luận án Tiến sĩ này là để dành tặng cho bố mẹ Thăng Thảo, vợ Linh, các con Long Minh, em trai Hoàng

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation towards my advisors, Professor René Mayer and Dr. Elie Bitar-Nehme for their guidance and time throughout my research. With their deep knowledge in Machine tool, I was not fallen to wrong ways and gave some contributions.

I would like to thank the committee members, Professors Luc Baron, Sofiane Achiche and Andrew Peter Longstaff for their time and support in evaluating my thesis.

I especially want to thank technicians of the Virtual Manufacturing Research Laboratory (LRFV) Mr. Guy Gironne, Mr. Vincent Mayer for their works.

I would like to express deep gratitude to my family for their big love and unconditional support. To my parents Vu Ngoc Thang and Vu Thi Thao, thank you for growing me up in the greatest love and teaching me the values of life. To my wife Nguyen Thi Ngoc Linh and my children Vu Ngoc Long and Vu Minh, thank you for being with me in difficult and happy times. To my brother Vu Huy Hoang, thank you for taking care our parents while I am not home.

Finally, I would like to thank Project 911 of Ministry of Education and Training of Vietnam and National Science and Engineering Research Council of Canada, Discovery Grant Program for the financial support.

RÉSUMÉ

Les erreurs induites thermiquement dans les machines-outils ont fait l'objet d'une attention particulière ces dernières années en raison de leur impact considérable sur l'exactitude de l'usinage. De nombreux travaux ont modélisé les comportements thermiques des machines-outils en utilisant les informations disponibles telles que les puissances des moteurs, les courants, les températures et les vitesses. Cette recherche adopte une approche de mesure, de modélisation et de recherche de la robustesse des modèles d'apprentissage profond entre les déformations thermiques et l'alimentation en énergie des axes, la température ambiante et les températures des positions prédéterminées. Les modèles entraînés sont validés dans d'autres processus présentant des activités et des durées différentes des axes.

Des expériences sont menées pour comprendre les comportements thermiques qui varient en fonction des routines de chauffage et de refroidissement différentes des axes rotatifs ou des cinq axes de la machine-outil. Les erreurs volumétriques brutes de la machine-outil sont collectées périodiquement (toutes les 15 minutes) par un ensemble de cinq capteurs capacitifs, ainsi que les puissances réelles des moteurs des cinq axes, les températures aux positions prédéterminées et la température ambiante, qui sont également collectées pour être utilisées en tant qu'entrées des modèles.

L'application de la méthode Stacked LSTM (SLSTM) pour modéliser la relation entre les 10 principales erreurs géométriques induites thermiquement, estimées par la méthode SAMBA, et les entrées correspondant aux puissances d'alimentation des axes rotatifs B et C constitue la première partie de cette thèse. Trois optimiseurs, à savoir SGD, RMSProp et Adam, sont utilisés dans ce travail afin d'améliorer l'efficacité du réseau. Les données utilisées pour l'entraînement et les tests proviennent de deux expériences d'une durée de 132 heures et 40 heures respectivement, avec des activités différentes pour les axes B et C.

Le deuxième travail se rapproche de la compensation des erreurs thermiques des trajets ∂x, ∂y, ∂z afin de déplacer l'outil vers des positions cibles dans l'espace de travail de la machine-outil HU40T. Ces erreurs sont induites par les activités des axes B et C, telles que menées dans le premier travail. L'optimiseur AdamW est utilisé dans ce travail pour soutenir le réseau SLSTM. Les SGRUs sont

également utilisées dans ce travail pour modéliser et prédire les meilleurs et pires cas générés par le SLSTM.

Le dernier travail a étudié le comportement thermique de la machine-outil HU40T en relation avec les activités des cinq axes (la broche est verrouillée) lors de deux expériences de plus longue durée par rapport aux travaux précédents. Ces expériences ont été menées pendant 264 et 84 heures respectivement. SLSTM+Adam est utilisé pour modéliser et prédire les erreurs géométriques thermiques, et Monte Carlo Dropout est combiné avec SLSTM pour déterminer la distribution des sorties du modèle. Les entrées du modèle sont la température ambiante, les températures aux positions prédéterminées et l'alimentation en énergie des cinq axes, et les sorties sont les erreurs géométriques induites thermiquement.

La recherche a apporté des contributions significatives aux modèles des comportements thermiques des machines-outils à cinq axes en utilisant les informations disponibles (les puissances des moteurs et les températures) comme entrées.

ABSTRACT

Thermally induced errors in machine tool have been paid attention in recent years due to their big impacts to the inaccuracy of machining. Many works modelled thermal behaviors of machine tools with the available information like motor powers, currents, temperatures, and speeds. This research adopts to measure, model and find the robustness of the deep learning models between thermal deformations and power supply of axes, ambient temperature, and temperatures of predetermined positions. The trained models are validated in other processes with different activities of axes and times.

Experiments are carried out to understand thermal behaviors altering to different heating and cooling routines of rotary axes or all five axes of machine tool. Raw volumetric errors of machine tool are collected periodically (15 minutes) by a sensor nest of five capacitive sensors, as well as the true motor powers of the 5 axes, temperatures at predetermined positions and ambient temperature are also collected to used as inputs of the models.

Applying Stacked LSTM (SLSTM) for modelling the relationship between 10 main thermal induced geometric errors estimated by SAMBA (Scale and master ball artefact) method and inputs as supply powers of rotary axes B and C is the first work of this thesis. Three optimizers SGD, RMSProp, and Adam are used in this work to improve the efficiency of network. The data for training and testing processes are collected from two experiments with different activities of B and C-axis in 132 and 40 hours, respectively.

The second work reaches closer to the compensation of thermal errors of paths $\partial x, \partial y, \partial z$ to move the tool to target positions in the working space of machine tool HU40T. These errors are induced by activities of axes B and C as conducted in the first work. Optimizer AdamW is used in this work to support SLSTM network. The SGRUs is also used in this work to model and predict the best and worst cases generated by SLSTM.

The last work investigated the thermal behavior of the machine tool HU40T relating to activities of five axes (the spindle is locked) in two longer time experiments comparing to the ones in the first and second works. These experiments are conducted in 264 and 84 hours, respectively. SLSTM+Adam is applied to model and predict thermal geometric errors and Monte Carlo dropout is combined with SLSTM to determine the distribution of the model's outputs. The inputs of the

model are the ambient temperature, the temperatures at predetermined positions and the power supply of five axes and the outputs are thermally induced geometric errors.

The research has made significant contributions towards the models of thermal behaviors of five axis machine tools with available information (the motor powers and the temperatures) as the inputs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Adam	Adaptive Moment Estimation
AI	Artificial intelligence
ANN	Artificial neural network
ARX	Autoregressive models with exogenous inputs
BGD	Batch gradient descent
BNN	Bayesian neural network
BPTT	Back propagation through time
BRNN	Bidirectional recurrent neural network
CNC	Computer Numerical Control
CNN	Convolutional neural network
CSGWO	Cosine and sine gray wolf optimization
FFNN	Feed forward neural network
GRU	Gated recurrent units
SGRU	Stacked GRU
KL	Kullback-Leiber divergence
LASSO	Least absolute shrinkage and selection operator
LRFV	Laboratoire de recherché en fabrication virtuelle
LSTM	Long short term memory
SLSTM	Stacked LSTM
MBS	Multi body system
MC	Monte Carlo
MCD	MC Dropout
MEFCS	A new mist edge fog cloud system

MI	Multiple input
MO	Multiple output
ONT-GCN	Ordered neuron temporal graph convolutional network
RNN	Recurrent neural network
SAMBA	Scale and master ball artefact
SGD	Stochastic gradient descent
SI	Single input
SO	Single output
SRT	Stochastic regularisation technique
ST-CLSTM	Spatiotemporal correlation LSTM
TALC	Thermal adaptive learning control
TFEM	Temporal finite element method
XGBoost	eXtreme Gradient Boosting

CHAPTER 1 INTRODUCTION

1.1 Motivation

Thermally induced deformations in machine tool have been a research topic for years and are the main cause of inaccuracy. They are up to 70% of total errors [1-3]. The behaviors of these deformations with the inputs such as motor powers and temperatures are nonlinear and the latter values are continuous with the preceding values. Modelling these nonlinear deformations with other available parameters (power, temperature, current, speed) is necessary because this does not interrupt manufacturing process. The model can be used to predict and compensate them in advance. Many approaches are used for this purpose such as linear regression methods, machine learning. The challenges in modelling the thermal behavior of the machine tools are the complexity of the thermal error model due to the large amount heat sources, coupling impacts, the nonlinear behavior of thermal errors, the influence of previous states to the current state, and the dependences at various levels of individual thermally induced errors on the inputs... These challenges require advanced methods for thermal error analysis and modeling the thermal behavior of the five-axis machine tools. Recurrent Neural Network (RNN) is one of the most selected artificial neural networks for sequential data. The drawback of RNN is the problem of vanishing gradient. Deep learning LSTM/ GRU with stochastic techniques are as a method to get over the above problems in modelling the thermal errors and have ability to keep prediction in a long period without re-measuring the outputs to be used as references (targets). The objective of this work is to model the relationship between the thermally induced deformations of five-axis machine tool and motor powers, temperatures, linear axis positions and rotary axes' indexations. The trained model does prediction in another process without re-measuring the outputs. Robustness of the model, an important principle to evaluate the reliability of prediction values, is also presented.

1.2 Research Objectives

The work aims at using reliable deep learning models to train and predict thermal errors of machine tool based on various inputs such as motor powers, ambient temperature and temperatures at predetermined positions. To achieve these main goals, three specific objectives need to be done:

1. The first objective is to model the relationship between thermally induced geometric errors with supply powers for B and C axes and predict these errors in another process using SLSTM.

2. The second is to model the relationship between thermally induced volumetric errors (TVEs) with supply powers for B and C axes and test in another process using SLSTMs/SGRUs networks. The reason to directly model and predict TVEs in the second work instead of using kinematic model to calculate volumetric errors from the predicted geometric errors in the first work is the existent of part of volumetric errors which cannot be explained by the kinematic model. For this work the inputs need to have more information of linear axis positions and the rotary axes' indexation besides the power consumptions.

3. The third work investigates the thermal behavior of machine tool HU40T relating to the activities of five axes. SLSTM is used to model the relationship between the thermally induced geometric errors and the inputs motor powers, temperatures and Monte Carlo dropout is applied to SLSTM model to find the distributions of the predictions. To enhance the robustness of deep learning models, a hypothesis test is conducted. Besides, the selection of inputs as temperatures or motor powers and the effect of ambient temperature are also investigated.

1.3 Thesis Outline

This thesis is organized as follows: Chapter 2 reviews recent works for the models of thermally induced deformations in machine tool and presents theories that are using in Chapter 3, 4, and 5, respectively. The goal of Chapter 3 is to model the relationship between outputs as thermally geometric errors and the inputs as supply powers for rotary axes B and C. The trained model is validated in another activity sequence. Chapter 4 goes further than Chapter 3 to reach a practical problem, which focuses on the thermally induced volumetric errors of x, y, and z paths to move a sensor nest to the positions at four artefact balls. In Chapter 5, the focus is on deep learning model between thermal errors caused by activities of five axes and exploring the effects of Dropout applying in deep learning model. The dropout causes the output distributions besides their strong points to make model efficiency. Chapter 6 gives general discussions, and finally Chapter 7 does the conclusion and recommendations for future research.

1.4 Co-Authored papers

- H.V. Ngoc, J.R.R. Mayer, E. Bitar-Nehme. Deep learning LSTM for predicting thermally induced geometric errors using rotary axes' powers as input parameters, CIRP Journal of Manufacturing Science and Technology, 37 (2022) 70-80. DOI: doi.org/10.1016/j.cirpj.2021.12.009

- H.V. Ngoc, J.R.R. Mayer, E. Bitar-Nehme. Deep Learning to Directly Predict Compensation Values of Thermally Induced Volumetric Errors, Machines 2023, 11, 496. DOI: doi.org/10.3390/machines11040496.

- H.V. Ngoc, J.R.R. Mayer, E. Bitar-Nehme. Applying Deep learning and Monte Carlo Dropout for Model's Confidence of Thermally Induced Kinematic Errors of Five Axis Machine Tool. Submitted to CIRP Journal of Manufacturing Science and Technology (3/06/2023).

CHAPTER 2 LITERATURE REVIEW

This chapter provides a comprehensive review of previous literature relating to algorithms to model thermally induced errors in machine tool, deep learning theories, optimizers, techniques that are used in Chapter 3, Chapter 4, and Chapter 5.

2.1 Machine tool thermal errors

Thermally induced errors affect the accuracy of machine tools up to 70% [1-3] of total errors. Two potential methods are typically used to decrease errors in manufacturing: error avoidance and error compensation. Error avoidance can be difficult and costly. Error compensation is an efficient way to get over the limitations of error avoidance. This way has a big way to go by applying existent techniques and theories to model, predict and compensate the machine tool errors (the geometric and volumetric errors) in advance with available inputs obtaining from thermal couples, power sensors, and encoders…

2.1.1 Thermally induced geometric errors

Geometric errors come from the concept of using theory of multi body system (MBS) to build a precision model of 5 axis Computer Numerical Control (CNC) machine tool under the dynamic behavior of rigid bodies. While, the introduction of Chapter 3provides an in-depth review of numerous studies, some additional works not discussed in Chapter 3 are outlined here.

Gui et al. proposed neuron temporal graph convolutional network (ONT-GCN) to conduct the spatial and temporal modelling of tooth surface error and geometric thermal errors with inputs temperatures on worm gear machine tools [4]. Besides [5] presented in Chapter 3, which proposed a new concept about adaptive learning control (ALC) for thermally geometric error compensation of rotary axes of five axis machine tool using a system of differential equations, other works of this adaptive approach were continued in [6-8]. In prediction phase, the errors were measured with intermittent probing to update parameters to match the present situation. Mayr et al. presented a continuous adaptation by applying Weighted Least Squares for parameter updates, on-machine probing to process the diverse sample rates. The efficiency of prediction and compensation of thermal geometric errors improved up to 95% [6]. Zimmermann et al. applied the thermal adaptive learning control (TALC) to improve the precision and robustness of thermal error models by defining the frequency of model updates. TALC used adaptive inputs, which were generated from Group-LASSO and the particle swarm optimization. This work compensated the thermal errors of a swiveling and a rotary axis of five axis machine tool during 350h and the errors were reduced up to 36% and 58% respectively [7]. Blaser et al. presented an extension of the [5] for activating motions of five axis machine tool in four different load cases [8].

2.1.2 Thermally induced volumetric errors

In this part, the models using different algorithms to model thermally induced volumetric errors with possible inputs that are not reviewed in Chapter 4 will be presented.

Li et al. used an Elman neural network optimized by a sparrow search algorithm to model and predict thermal errors in motorized spindles. The positions and number of thermal sensors were determined by pedigree and k-means clustering methods with grey correlation degree [3]. Horejš et al. improved the approximation of thermal error compensation in z direction by developing three separate models for three five axis machining centers of the same type equipped with three different spindle units. To verify the ability of compensation, these models were tested with different spindle speeds [9]. The inverse heat transfer analysis was proposed in [10] for predicting a time varying thermal error of ball screw feed system. A data-driven approach and a dynamic thermal network model were used to estimate the heat source of a ball screw feed system under unsteady thermal effects. The thermodynamic equilibrium equation also presented these heat sources. The optimization was based on Monte Carlo algorithm. Guo et al. presented a static thermal error model using a hybrid CNN-LSTM with spatiotemporal correlation (ST-CLSTM). Temperature data measured from 10 sensors was sampled with spatiotemporal correlation, then CNN was applied to build the relationships between temperature variables and thermally volumetric errors. LSTM was used for predict the change of temperature [11]. Chengyang et al. used CNN to build the relationship between thermally induced angular position errors of rotary axes and temperature images in both heating and cooling conditions [12]. An predictive thermal error model was proposed in [13] for CNC machine tool, which applied the adaptive Least Absolute Shrinkage and Selection Operator (LASSO) for selecting temperature sensitive variables and eXtreme Gradient Boosting (XGBoost) for predictive model. Five-point test method was used in [14] to measure thermal deformations of spindle in x, z directions. The prediction accuracy and robustness of thermal error model were proposed in [15], it took year around to collect the data with the change of ambient temperature intervals. The established model had a higher accuracy with 20.6% and 41.7% comparing with the low and high ambient temperature models. Peng et al. studied on the spindle axial thermal error and thermal bending angles, caused by the rotation of spindle. A physically based model was built to explain thermal bending behaviors in different environmental temperature, spindle temperature and column temperature [16]. Marei and Li introduced an Autoencoder-Augmented LSTM model for predicting the z-axis offset of machine tools based on temperatures, which were selected from input data (52 sensors) the by applying Autoencoder [17]. A new approach based on the principle of heat energy conduction and neural networks for forecasting thermal volumetric error in grinding machines was introduced in [18]. This work firstly collected the heat energy exchange by measuring the temperature deviations between spindle surface and ambient environment. Subsequently, these temperature deviations were employed as inputs of a convolutional neural network. A new mist edge fog cloud system (MEFCS) architecture was proposed in [19] to control and predict thermal error. To validate this method, a finite element method was used to prove the applicability of bidirectional LSTM network. Batch size was optimized by a cosine and sine gray wolf optimization (CSGWO) algorithm. CSGWO-Bi-LSTM network was proposed and prediction accuracy was up to 98.92%.

2.1.3 Uncertainty prediction models

The model’s distribution of prediction is a crucial aspect that needs to be addressed. In this regard, additional models that were not presented in Chapter 5 will be reviewed here.

Liu et al. applied BNN for modelling thermal error of linear motor feed system. Grey relation analysis was used to determine temperature-measuring points and avoid multicollinearity. The BNN’s hyper-parameters were random and calculated through Gauss-Newton approximation [20]. Ahmadi’s work involved identifying the dominant closed loop poles of the machining system from in-process vibration signals. These identified poles were then used to determine probability distributions of model parameters and credibility bounds [21]. Yao et al. proposed the application of a Bayesian network to model uncertain relationship between thermal deformations of spindle and temperatures, spindle speed, feed rate and cooling liquid. The modeling process started with prior knowledge: identifying main factors that contribute to thermal errors, analysing correlation between them and determining prior probability of each variable. The probabilistic inference was used to obtain the probability of the model [22]. Wan et al. quantified uncertainty by performing stochastic temporal finite element method (TFEM) under different conditions with different random uncontrollable variables [23]. Some works did uncertainty quantification by using edge theorem to predict the stability boundary for both single and multiple degree of freedom milling system [24] or MC method to catch the probabilistic characterization of the regenerative chatter in turning [25]. [26] captured uncertainty for estimating streamflow quantiles by using MC dropout in ANN. [27] found uncertainty estimation of soil moisture by applying MC dropout for LSTM.

2.2 RNN

A recurrent neural network (RNN) or Backpropagation through time [28] is a class of neural network that is particularly well-suited for processing time series data. Unlike Feedforward Neural Networks (FFNNs), RNN can handle sequential data by taking into account the influence of previous inputs on the present input and output. To achieve this, RNN incorporate a memory concept that allows them to store information of the previous inputs.

Figure 2.1 provides a detailed illustration of a “rolled” RNN. The “unrolled” visualization represents the time steps of the network. Each layer in the unrolled RNN shows a part of sequence information and contributes partly to the prediction of output in sequence.

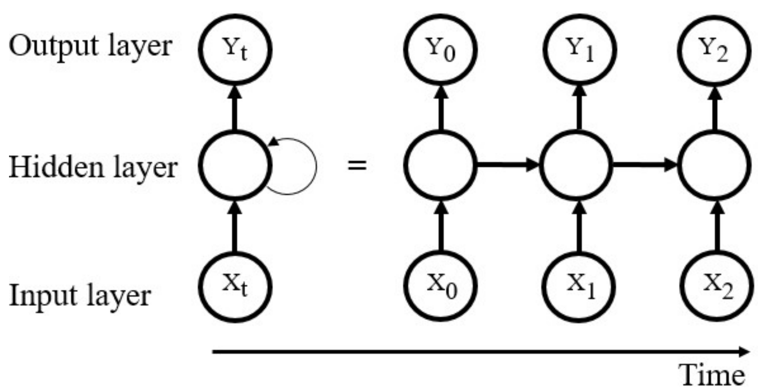

Figure 2.1. Three time step sequence [29]

RNNs uses Back propagation through time (BPTT) to determine the gradients, where the model is trained by calculating differences from its output layer to input layer to adjust and fit the weights and biases approximately.

Activation functions*:* Sigmoid and Tanh

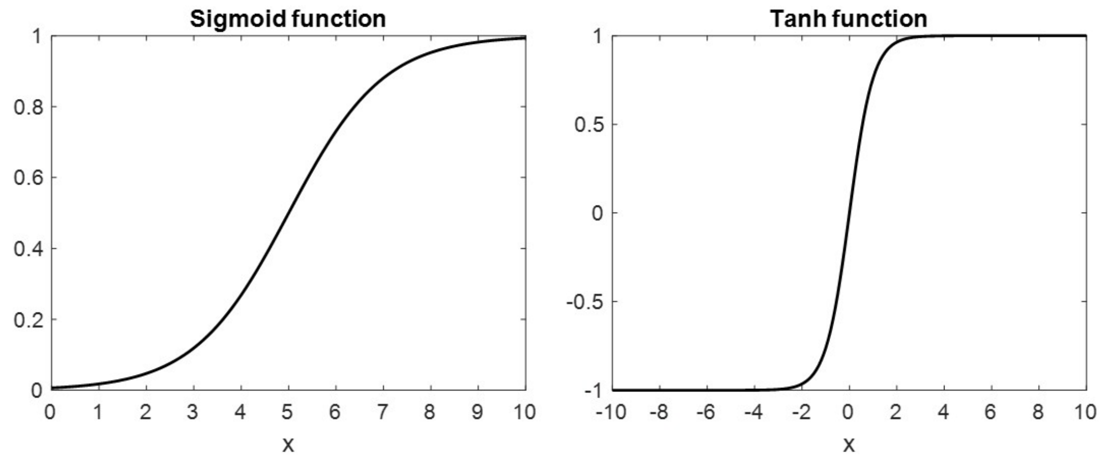

Figure 2.2. Activation function Sigmoid and Tanh [30]

Although RNNs can process time series any length without changing model structure, RNNs suffer from the problem of vanishing gradient, which prevents them from effectively modeling long historical data. Vanishing gradient occurs when gradient is too small, rendering weight updates insignificant and causing the algorithm to stop learning.

Loss function

$$L(\hat{y}, y)=\sum_{t=1}^{T_y} L(\hat{y}^{<t>}, y^{<t>}) \tag{2.1}$$

Backpropagation through time

$$\frac{\partial L^{(T)}}{\partial W}=\sum_{t=1}^{T} \left.\frac{\partial L^{(T)}}{\partial W}\right|_{(t)} \tag{2.2}$$

2.3 Long Short Term Memory (LSTM)

Multi variate time series (MTS) have gained popularity in recent years, owning to advances in sensors and measurement technology that allow collecting data over time [31]. RNNs are often used to address forecasting problems of time series due to possessing internal states or short-term memory, which is enabled by recurrent feedback connections [31, 32]. However, these networks can face vanishing problems when learning long-term dependencies [33]. The long short-term memory (LSTM) [33] is proposed to address this problem. Unlike RNNs, LSTMs can retain long-term memory of previous learning information when processing time series data [34].

LSTM unit is defined as Figure 2.3 presented in [35]. It has an input vector [H(t-1),x(t)], cell state C(t) at time step t. and the output vectors H(t)

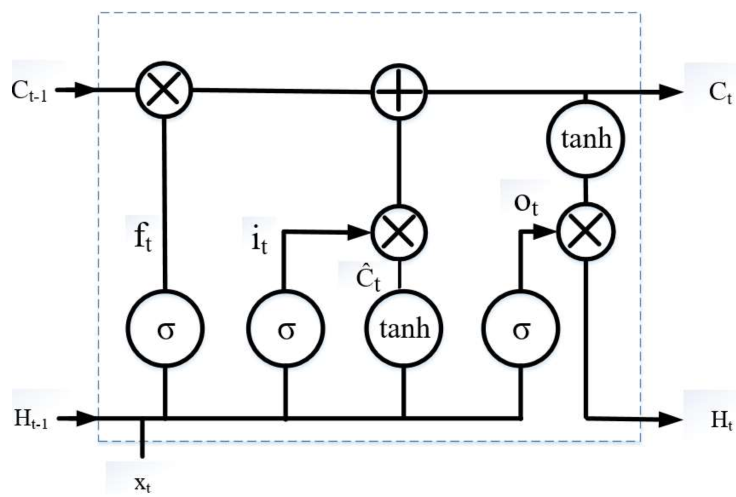

Figure 2.3. LSTM unit [35]

LSTM unit has three gates: the forget gate f_t, input gate i_t, and output gate o_t that update and control the cell state C_t. The gates use hyperbolic tangent (tanh) and sigmoid activation functions.

$$f_t=\sigma_g(W_f x_t+U_f H_{t-1}+b_f) \tag{2.3}$$

Sigmoid function σ_g determines how much information can pass through forget gate f_t

$$i_t=\sigma_g(W_i x_t+U_i H_{t-1}+b_i) \tag{2.4}$$

$$\widehat{C}_t=\tanh(W_c x_t+U_c H_{t-1}+b_c) \tag{2.5}$$

After sigmoid function σ_g let information through the input gate i_t, tanh function gives weightage to these information and regulate their values (-1 to 1). Updating new cell state is:

$$C_t=f_t \odot C_{t-1}+i_t \odot \widehat{C}_t \tag{2.6}$$

The following equations determine the output

$$o_t=\sigma_g(W_o x_t+U_o H_{t-1}+b_o) \tag{2.7}$$

$$H_t=o_t \odot \tanh(C_t). \tag{2.8}$$

Backpropagation through time in LSTMs

$$\frac{\partial E}{\partial W}=\sum_{t=1}^{T}\frac{\partial E_t}{\partial W} \tag{2.9}$$

Preventing the error gradients from vanishing

The LSTM network can decide which information should be discarded by a sigmoid layer known as the "forget gate layer". This allows the LSTM to control the gradients at each time step, which prevents the error gradient from vanishing.

$$\frac{\partial E_{k+1}}{\partial W} \nrightarrow 0 \tag{2.10}$$

The shallow LSTM architecture does not represent the complex features of sequential data efficiently [31]. A deep LSTM model (DLSTM or Stacked LSTM) [36] has capability of overcoming the limitations of the shallow LSTM network. The structure of Stacked LSTM (SLSTM) contains LSTM layers and each layer contains multiple units as Figure 2.4.

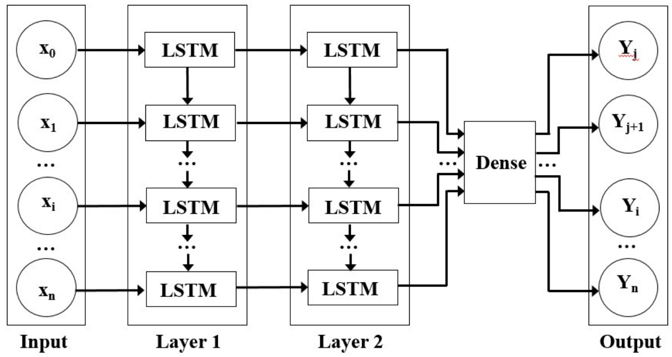

Figure 2.4. A SLSTM model

2.4 Gated Recurrent Unit (GRU)

GRU [37] is a variant of the RNN, which has ability to hold on long term dependencies from the computations within the GRU unit. GRUs were developed later than LSTM by using two gates instead of using three gates and memory cell as LSTM.

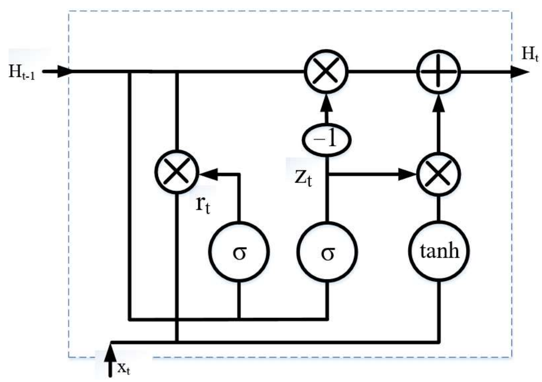

Figure 2.5. A GRU unit

Two gates of GRU are reset gate r_t and update gate z_t, which are give sigmoid activations to transform their values to interval [0, 1]

$$r_t=\sigma(x_t W_{xr}+H_{t-1} r_{hr}+b_r), \tag{2.11}$$

$$z_t=\sigma(x_t W_{xz}+H_{t-1} r_{hz}+b_z), \tag{2.12}$$

where W_{xr}, $W_{xz} \in R^{d\times h}$ and W_{hr}, $W_{hz} \in R^{h\times h}$ are weight matrices and b_r, b_z are biases

The reset gate r_t is responsible for capturing short-term dependences in sequences, while the update gate has ability to capture long-term dependences in sequences. The candidate-hidden state

$$\tilde{H}_t=\tanh(x_t W_{xh}+(r_t \odot H_{t-1})W_{hh}+b_h) \quad (2.13)$$

The hidden state

$$H_t=(z_t \odot H_{t-1})+(1-z_t)\odot \tilde{H}_t \quad (2.14)$$

2.5 Optimizers

Optimizers are mathematical functions, which commonly use Gradient descent [38, 39]. They minimize an objective function $\nabla_\theta J(\theta)$ with respect to model's parameters (weights and biases) and model's parameters θ_s are updated during this process.

2.5.1 Batch gradient descent (BGD) [40]

In Batch gradient descent, the gradient of the cost function w.r.t. to the parameters θ is computed for the entire training dataset:

$$\theta=\theta-\eta.\nabla_\theta J(\theta) \quad (2.15)$$

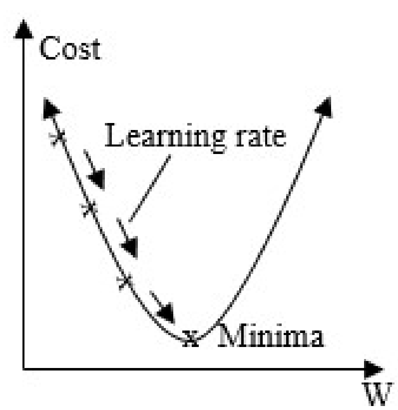

Figure 2.6. Batch gradient descent

One disadvantage of using batch gradient descent is that the gradient is calculated for the entire dataset before updating the model's parameters. This approach can be slow and is not suitable for updating model's parameters online with new information.

2.5.2 SGD [40]

Unlike BGD, Stochastic gradient descent (SGD) updates model's parameters every training example $x^{(i)}$ and label $y^{(i)}$:

$$\theta=\theta-\eta.\nabla_{\theta}J(\theta;x^{(i)};y^{(i)}) \tag{2.16}$$

SGD is a modification of gradient descent, where gradient is calculated by random selecting a part of all observations. This reduces computation cost. SGD can estimate the gradient of the cost function for each observation and update decision parameters accordingly.

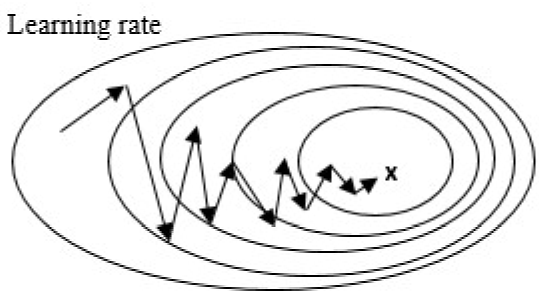

Figure 2.7. Stochastic gradient descent

SGD cost function has a severe fluctuation, which is from random observations and updates frequently. However, this is capable to fall to new and potentially better local minima. Learning rate can be adjusted decrease when overshooting appears.

Another disadvantage is that it can converge to a local minimum because whole data is considered at one time.

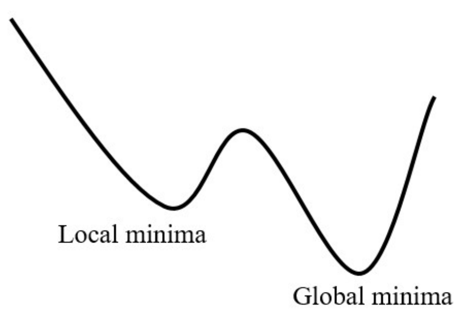

Figure 2.8. Local and global minima

2.5.3 Mini-batch gradient descent [40]

Mini-batch gradient descent is a variation of gradient decent algorithm, which splits the training dataset into small batches. It performs an update for every mini-batch of n training examples:

$$\theta=\theta-\eta.\nabla_{\theta}J(\theta;x^{(i:i+n)};y^{(i:i+n)}) \tag{2.17}$$

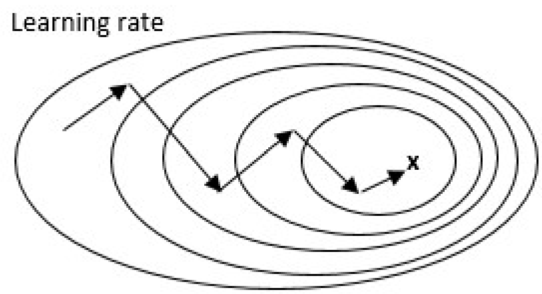

Figure 2.9. Mini batch gradient descent

This approach can help to reduce the variance of the parameter updates and make convergence more stable.

2.5.4 RMSprop [40]

RMSprop is an adaptive learning rate method proposed by Geoff Hinton [41], inspired from Rprop [42]. Rprop uses the sign of the gradient and individually adapt learning rate for each weight. Rprop has problem with large datasets and need to perform mini-batch weight updates.

RMSprop is

$$\theta_{t+1}=\theta_t-\frac{\eta}{\sqrt{E\left[\nabla_{\theta}J^2\right]_t+\epsilon}}\nabla_{\theta}J_t \tag{2.18}$$

where: $E\left[\nabla_{\theta}J^2\right]_t$ is moving average of squared gradient. RMSprop divides the gradient by the moving average of the squared gradients for each weight. The moving average is done on a successive set of n values rather than the whole set of values. The reason for keeping the moving average of squared gradient is to balance the step size and prevent exploding gradients for large gradient values and vanishing gradients for small gradient values by decreasing and increasing step size, respectively. RMSprop can work with stochastic objectives and can be applied to mini batch learning.

2.5.5 Adam [40]

Adaptive Moment Estimation (Adam) [43] is a algorithm that computes adaptive learning rates for each parameter.

$$\theta_{t+1}=\theta_t-\frac{\eta}{\sqrt{\hat{v}_t+\epsilon}}\widehat{m}_t \tag{2.19}$$

where $\widehat{m}_t$, $\hat{v}_t$ are the bias-corrected first and second moment estimates

$$\widehat{m}_t=\frac{m_t}{1-\beta_1^t} \tag{2.20}$$

$$\hat{v}_t=\frac{v_t}{1-\beta_2^t} \tag{2.21}$$

$\widehat{m}_t$, $\hat{v}_t$ are from an exponentially decaying average of past squared gradients v_t like RMSprop, and an exponentially decaying average of past gradients m_t, similar to momentum.

$$m_t=\beta_1 m_{t-1}+(1-\beta_1)\nabla_\theta J_t \tag{2.22}$$

$$v_t=\beta_2 v_{t-1}+(1-\beta_2)\nabla_\theta J_t^2 \tag{2.23}$$

β_1, β_2: are the decay rate of the first moment (momentum) and the second moment (sum of gradient squares). Normally β_1=0.9, β_2=0.999.

Adam has the advantages of the momentum (the first moment or sum of gradients) and RMSprop (the second moment or sum of gradient squared) and gives a better quality of optimizing process for neural networks. Chapter 3 will provide evidence to support this claim.

2.5.6 AdamW [44]

L_2 regularization is added to the loss function as a solution for having the effect of weight shrinking during backpropagation. This partly prevents the network from overfitting the training data.

$$\text{Loss}=\text{Error}(y,\hat{y})+\lambda\sum_{t=1}^{N}\theta_t^2 \tag{2.24}$$

where $\lambda\sum_{t=1}^{N}\theta_t^2$ are L_2 terms.

However, it is not effective for adaptive gradient algorithms as Adam, because the weights do not decay multiplicatively as expected. The reason is the gradient of loss function has the L_2 norms as follows

$$\nabla_\theta J_t=\nabla_\theta J_{t-1}+\lambda\theta_t \tag{2.25}$$

Then m_t and v_t are updated as formulas (2.22) and (2.23), which means the decay averages of gradient and its square keep track not only the gradient of the loss function but also the L_2 term. To improve, [44, 45] proposed a simple and effective way by decoupling weight decay from the gradient based update. This means the optimal choice of weight decay factor is independent to adaptive learning rate. [46] described the weights θ decay exponentially as

$$\theta_{t+1}=(1-\lambda)\theta_t-\alpha\nabla_\theta J_t \tag{2.26}$$

where λ is the rate of the weight decay per step, $\nabla_\theta J_t$ is the t-th batch gradient, and α is learning rate.

Only the gradient of loss function is updated, weight decay is kept same rate λ, then m_t, v_t, $\widehat{m}_t$, $\hat{v}_t$ are calculated as in equations from (2.20) to (2.23) without gradient of L2 norm, then the weight is updated with

$$\theta_{t+1}=\theta_t-\eta_{t+1}\left(\frac{\alpha\widehat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}}+\epsilon}+\lambda\theta_t\right) \tag{2.27}$$

2.5.7 Bayesian probability modelling [47]

A Bayesian network is a probabilistic model that provides a mathematic technique for diagnosing and making decision in problems that involve inherent uncertainty [48, 49]. Bayesian probability theory provides a mathematical structure to implement inference. In Bayesian probability theory, posterior distribution follows Bayes' rule.

$$p(\theta|x,y)=\frac{p(y|x,\theta)p(\theta)}{p(y|x)} \tag{2.28}$$

where: $p(\theta)$ is the prior distribution which reflects the probability of the hypothesis before having observed data.

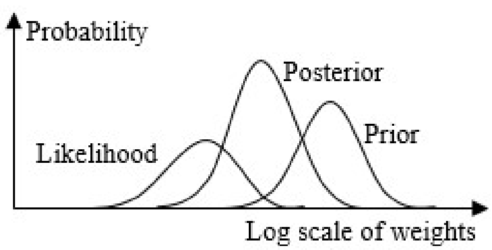

Figure 2.10. Maximum likelihood

$p(\theta|x,y)$ is the posterior distribution which expresses the probability of the hypothesis after having observed data

$p(y|x,\theta)$ is the likelihood function which represents probability of the data under the hypothesis

$p(y|x)$ is the evidence which represents probability of the data under any hypothesis

$$p(\theta|x,y)\propto p(y|x,\theta)p(\theta) \tag{2.29}$$

This distribution covers most of probable parameters and in the prediction phase with a new input point x^* we have a new prediction bound

$$p(y^*|x^*,x,y)=\int p(y^*|x^*,\theta)p(\theta|x,y)d\theta \tag{2.30}$$

This procedure is an inference, which is a concept of integration over model parameters involving optimisation at training time.

2.5.8 Variational inference

Variational inference is an approximate solution to the inference problem, which can not be intractable in probabilistic models [50]. The aim of this approach is to optimize a problem over a class of tractable distributions Q with $q \in Q$ most close to p. Kullback-Leiber (KL) divergence [51] is known as a measure to minimize the difference between two distributions

$$D_{KL}(q(\theta)||p(\theta|x,y))=\int q(\theta)(\log q(\theta)-\log p(\theta))=\int q(\theta).\log\frac{q(\theta)}{p(\theta)} \qquad (2.31)$$

Where $p(x_i)$ is probability of outcome x_i and $q(x_i)$ is a replaced probability

$q(\theta)$ is the solution of the minimisation and the predictive distribution is:

$$p(y^*|x^*,x,y)=\int p(y^*|x^*,\theta)q(\theta)d\theta \qquad (2.32)$$

This process is considered as the variational inference (VI).

2.5.9 Dropout

Deep Neural Networks is a highly effective tool in machine learning due to their ability to leverage a large number of parameters and multiple hidden layers to facilitate the learning process. However, overfitting is a common challenge that can lead to unexpected prediction outcomes. To address this issue, dropout [52] is employed. During training process, random units are dropped to resist excessive co-adapting of parameters and improve the quality of predicted results. Figure 2.11 illustrates the temporarily removing units from the network. The chosen units to drop are random.

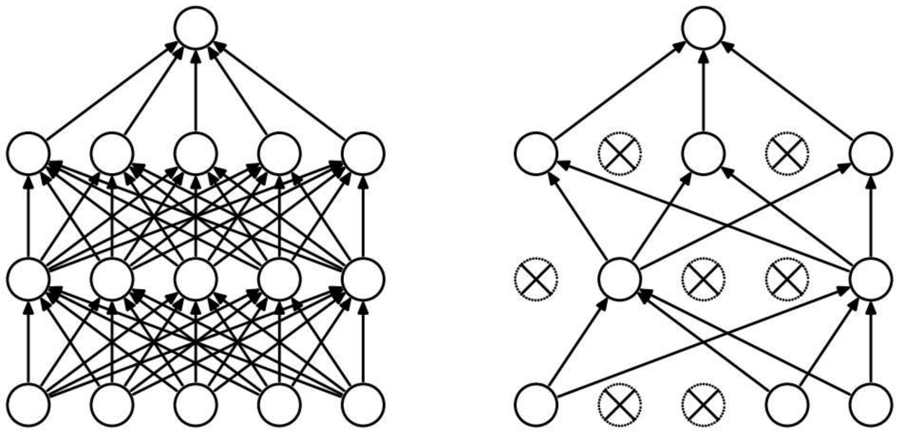

Figure 2.11. Dropout [52]

In neural network, dropout can be used in a particular way by sampling two binary $\hat{\epsilon}_1$,$\hat{\epsilon}_2$. The $\hat{\epsilon}_1$, $\hat{\epsilon}_2$ are the Bernoulli distributions [53] with probability p_i. The parameters are:

$$\widehat{W}_1 := \mathrm{diag}(\hat{\epsilon}_1) W_1 \text{ and } \widehat{W}_2 := \mathrm{diag}(\hat{\epsilon}_2) W_2$$

The objective is:

$$\widehat{L}_{dropout}(W_1, W_2, b) := \frac{1}{M} \sum_{i \in S} E^{\widehat{W}_1^i, \widehat{W}_2^i, b}(x_i, y_i) + \lambda_1 \|W_1\|^2 + \lambda_2 \|W_2\|^2 + \lambda_3 \|b\|^2 \quad (2.33)$$

The objective's derivative w.r.t parameters $\theta=\{W_1, W_2, b\}$

$$\frac{\partial}{\partial\theta} \widehat{L}_{dropout}(\theta) = -\frac{1}{M_T} \sum \frac{\partial}{\partial\theta} \log p(y_i | f^{g(\theta,\hat{\epsilon}_i)}(x)) + \frac{\partial}{\partial\theta} (\lambda_1 \|W_1\|^2 + \lambda_2 \|W_2\|^2 + \lambda_3 \|b\|^2) \quad (2.34)$$

2.5.10 Bayesian neural networks

NNs have illustrated their effectiveness in solving various real world problems. However, they cannot provide a probability distribution of their predictions due to pointwise optimization [54]. Bayesian NN (BNN) is a solution to train the model parameters as a distribution $p(\theta)=p(W_i)=N(0, I)$ rather than searching for a optimal value [55-60]. Recent works in BNNs focus on variational inference based on techniques to approximate intractable posterior distribution $p(\omega|X,Y)$. Gal [61] proposed a practical method to do variational inference by applying Monte

Carlo (MC) dropout technique, which involves conducting multiple stochastic forward passes (T passes) through the network and then averaging the results

The output average is:

$$\widetilde{E}[y^*] := \frac{1}{T}\sum_{t=1}^{T} f^{\hat{\theta}_t}(x^*) \tag{2.35}$$

The predictive variance is obtained as

$$\mathrm{std}_{q(y^*|x^*)}(y^*) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} (\hat{y}^* - \bar{\hat{y}}^*)^2} \tag{2.36}$$

2.6 Z-scores and p-values

A statistical measurement called Z-score can be used to describe how a value is related to the mean of a set of values. Z-score is calculated by measuring the number of standard deviations a data point is away from the mean. A Z-score of zero indicates that the data point has the same score as the mean. If the Z-score is one, it means that the value is one standard deviation away from the mean. Z-scores can be either positive or negative. A positive Z-score indicates that the score is above the mean, while a negative score indicates that it is below the mean.

$$z = \frac{x\text{-mean}}{\sigma} \tag{2.37}$$

Where z: z-scores, x the value being evaluated, σ the standard deviation

The p-value is mathematically computed using integral calculus by calculating the area under the probability distribution curve for all statistics values that are equal to or greater than the observed value, in comparison to the total area under the probability distribution curve. This process involves calculating the probability of obtaining results as extreme as or more extreme than the observed results, assuming that the null hypothesis is correct. The p-value is then compared to the level of significance to determine whether to reject or fail to reject the null hypothesis. If p-value is small, the null hypothesis can be rejected.

Formula of p value

$$\text{Left tailed test: p-value=cdf(x)} \tag{2.38}$$

$$\text{Right tailed test: p-value=1-cdf(x)}$$

$$\text{Two tailed test: p-value=}2\times\min\{\text{cdf(x)},1-\text{cdf(x)}\}$$

CHAPTER 3 ARTICLE 1: DEEP LEARNING LSTM FOR PREDICTING THERMALLY INDUCED GEOMETRIC ERRORS USING ROTARY AXES' POWERS AS INPUT PARAMETERS

Huy Vu Ngoc*, J.R.R. Mayer, Elie Bitar-Nehme

Department of Mechanical Engineering, Polytechnique Montreal

Published in CIRP Journal of Manufacturing Science & Technology, Vol.37, Pages 70-80, 2022
Date of publication: 20/01/2022

Abstract

Temperature changes within a machine tool, which are partly due to internal heat sources related to the power consumption at the machine drives, result in thermally induced geometric errors of the machine. Predicting the thermally affected machine geometry facilitates the prediction of volumetric errors throughout the 5-axis machine volume. Measuring power instead of temperature is easier to implement in practice but causes drift issues. Stacked Long Short Term Memory (LSTM) with two hidden LSTM layers is applied to determine the relationship between powers and the thermal variation of geometric error parameters by remembering previous states and learning complex non-linear relationships through optimized generalized parameters of the predicting model. Optimizers Adam, RMSprop, and SGD based on exponential moving average of first order and second order moments of gradients are combined with Stacked LSTM for faster learning convergence to an optimal solution. Validation trials with various B- and C-axis motion sequences demonstrate a capability of predicting multi time steps ahead over a 40 hours period without re-measuring the geometric error parameters of the model. Stacked LSTM with the Adam optimizer has the best capability by predicting up to 93% of the main geometric error compared to the other optimizers.

Keywords: Geometric errors; deep learning LSTM; machine tool; thermal errors

3.1 Introduction

Thermally induced geometric errors cause inaccuracy in machining estimated at 65% of all geometric errors [62], [63], and [64]. Predicting them using readily available information in real time has been an active subject of research in recent years in order to reduce their effects by finding the relationship as physics based or data-driven models [63] between thermal deformation and inputs like temperatures, powers etc.

Physics based models were built based on differential formulas [63]. Uhlmann and Hu presented a 3D Finite element method (FEM) model to predict the thermal behavior of a high speed motor spindle through tool centre point (TCP) displacement in the x, y and z directions [65]. Wu and Kung applied FEM to analyze the change in temperature and the deformations of a ball screw feed drive system [66]. Dos Santos et al. combined artificial neural network (ANN) and (FEM) on a three axes machine. FEM is used to determine the thermal errors while ANN is used to generate the relationship between temperature changes and thermal errors. The validation is carried out separately in ball screw system, linear guides and spindle [67]. Mayr et al. used a dynamic gray box model which consists of a system of differential equations describing the relationship between geometric error parameters induced by rotary axes and spindle and inputs: cooling powers and spindle speed [62]. Blaser et al. built differential equations with adaptive parameters to predict thermal position and orientation errors over the long term for 5-axis machine tools [5]. Zhang et al. generated thermal transfer function of machine tool from first order equations of major components (spindle, column, table…). The thermal deformations are caused by time varying environmental temperature. Validating model is carried out in the Z axis and compared with FEM simulation [68]. Kang et al. developed two separate thermal models for ball screw expansion and spindle thermal growth based on a heat transfer mechanism and experimental data using as inputs the feed rate and spindle speed variations [69]. Liu et al. analyzed the thermal behavior of the machine-workpiece system and built a thermal error prediction model based on heat transfer theory [70].

Data-driven models based on collected data of the system without using physics based modelling [63] is an alternative approach. Data-driven models are adaptable to different context but they provide no physical insight into the physical mechanisms at work. Huang used multiple regression analysis to develop a relationship between thermally induced position error of the feed drive system and temperatures of the front bearing, nut and back bearing [71]. Yu et al. explained that a first order linear model cannot reflect actual thermal errors so additional items from polynomial, interaction and cyclical series were proposed to model the relationship between temperatures and deformations [72]. Brecher et al. determined the transfer function with the first and second delay terms between the output volumetric errors and the input load parameters such as rotational speeds and motor currents of spindle and feed drives and environmental temperature [73]. El Ouafi et al. proposed an ANN based quasi optimal thermal error model with the selection of the best combination of temperature variables [74]. Abdulshahed et al. used ANFIS architecture for model correlation between measured temperature changes inside a small CNC milling machine structure as well as ambiant temperature and the thermal deformations [75]. Fujishima et al. used Ridge regression and Convolutional Neural Network (CNN) to predict the thermal displacement error in the X-axis of a turning center from ambiant temperature variation [76].

The efficiency of prediction using physics based and data driven models are shown in the above works. A limitation of the physics based model is that its accuracy depends highly on mathematical descriptions of heat sources (loss of drives through friction, machining heat, heat irradiation etc.). These descriptions are not easy to obtain exactly [63] due to the complex effects of machine working cycles and environmental conditions. Data driven model using manual analysis is used for a specific system. This is not generalized for other systems and is not efficient for complicated structures and vast amounts of data [77]. Recently, computers and algorithms have become far more powerful to analyze large datasets and adjust model parameters. Thermal geometric errors in machine-tool are continuous nonlinear data sequences [78, 79], and previous events are affecting current events. A more sophisticated and flexible model with memory mechanisms to handle sequential data and updating parameters continuously is desired [80]. Recurrent Neural Network (RNN) with cyclic connections are powerful for modeling sequences [81]. However, the vanishing gradient limits RNN widespread use for sequential problems. LSTM [33] typically incorporate memory mechanisms which improve the problem of vanishing gradient and are successful to learn long term dependencies [82]. Liang et al. showed that the LSTM network has better performance for thermal error problem compared to RNN and CNN [83]. Recent works continued to demonstrate the applicability of LSTM to train models between thermal errors and spindle speed [84, 85], spindle thermal displacement in Z direction and key temperature points [86], thermal errors of ball screws and temperature of feed drive system [87]. Although LSTM has the property of selectively storing information and can train non-linear data [33, 84, 88, 89], LSTM, in the above works, performs efficiently with regular updates of the actual values of the output variables, i.e. actual machine thermal errors, obtained from measurements in testing phase. However, access to such data in an industrial setting requires stopping production shortly to conduct a calibration sequence, which reduces productivity. Finding LSTM networks with multiple layers combined with optimizing algorithms is necessary to forecast outputs over longer periods. Although stacking multiple layers is known to provide better learning ability for longer term prediction of patterns [90, 91] with more flexible parameters to improve the prediction capability, Stacked LSTM networks are more complicated and require more computational time. Optimizers are necessary to help the network converge faster and update parameters more frequently. Gradient descent with momentum-based algorithm is used to strengthen LSTM [83, 92].

In this work, three optimizers i.e. Adam, RMSprop, and SGD are used to improve the efficiency of predicting multi time steps ahead in time series thermal problems. This article models the change in machine geometry under thermal effects caused by rotary axes' activities in the workpiece branch on a 5-axis machine tool (HU40-T). The experimental test sequence is described in Section 3.2. Section 3.3 introduces Stacked LSTM. Training and validation using Stacked LSTM models are shown in Section 3.4. Section 3.5 presents the results and discussions, and is followed by a conclusion.

3.2 Experimental process

The experimental work is carried out on a Mitsui-Seiki HU40-T horizontal machining centre with a wCBXfZY(C1)t topology. Figure 3.1 shows the machine workspace with the metrological artefact mounted on the table, the sensor's nest mounted in the tool holder and the added sensors to measure the power to the axes.

Figure 3.2 (left) shows the training process of activity as power level of the B and C rotary axes against time which is aimed at generating heat within the machine and cause thermal errors. The motions of rotary axes follow a reciprocating motion between their position limits (from -90° to +90° for the B axis and from -180° to +180° for the C-axis, respectively). The training process lasts

132 hours (528 cycles). The raw volumetric deviations, induced by the running axes, are measured as in [64].

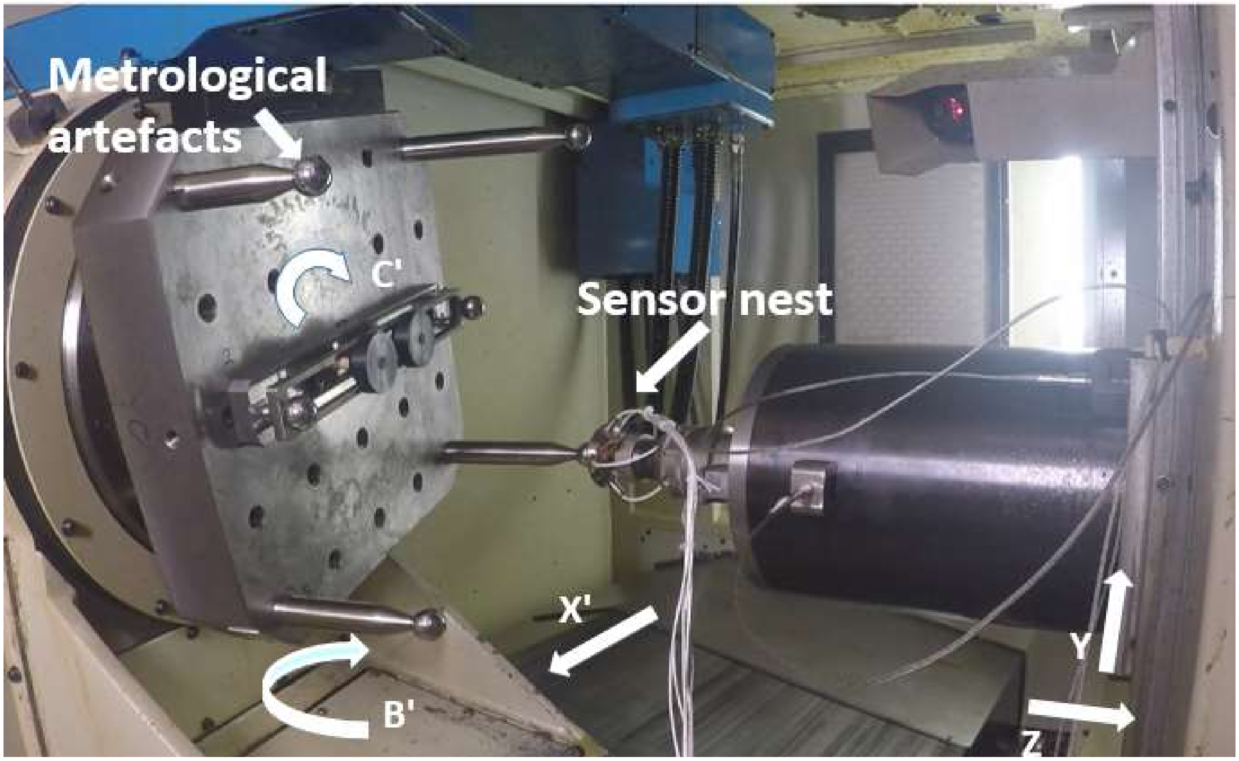

Figure 3.1. Sensor's nest measuring one of the metrological artefacts

An R-test type device with five non-contact precision capacitive proximity sensors form a sensors' nest (Capball) mounted in the tool holder that can detect the position of a precision ball. On the machine table, four precision balls with uncalibrated positions and a reference scale bar made of two balls separated by a calibrated distance altogether form the scale and master ball artefact (SAMBA). In the training process, each power level of heating cycles lasts 12 hours. One heating/measurement cycle lasts 15 minutes. The heating activity takes 12 minutes, followed by the measuring cycle, which takes only 3 minutes in order to limit changes in the machine's geometric thermal state. The power values are average Power consumptions of axes B and C for each 15 minutes activity. They are used as the inputs of the prediction model. So each power level will cover 12 hours × 4 cycles = 48 cycles.

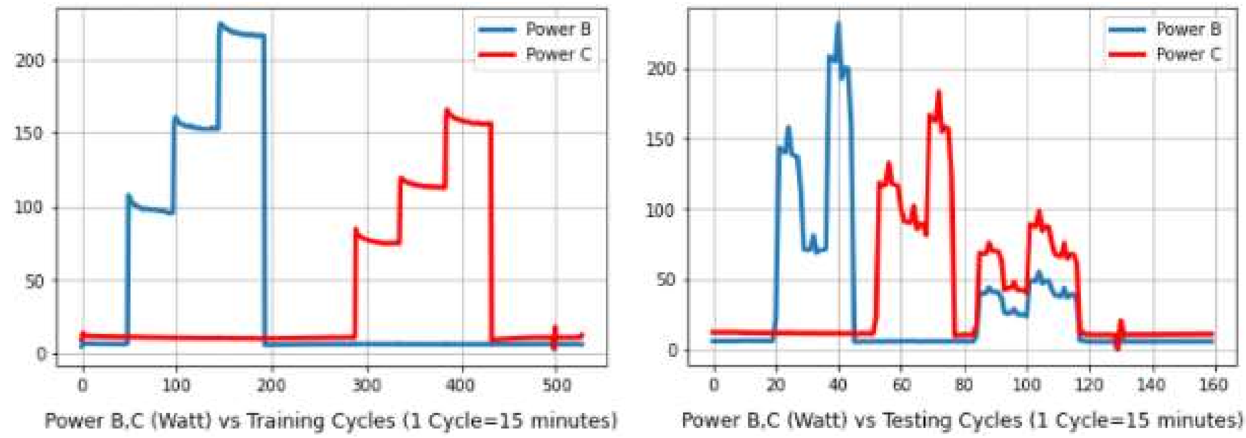

Figure 3.2. Power B and C in training and testing processes

Figure 3.2 (right) shows the testing process of activity, which lasts 40 hours (160 cycles), with individually and simultaneously operated rotary axes. Speed setpoints and the time at each speed setpoints are significantly different from the training data and also include periods when both axes are active.

3.3 Stacked LSTM

LSTM was proposed by Hochreiter and Schmidhuber [33] to solve the vanishing gradient issue in RNN when learning long data sequence through backpropagation in the training process. The LSTM cell in Figure 3.3 (left) shows the components of LSTM: the input, output, gates (f_t, o_t, and i_t), cell state (C_t) and activation functions sigmoid σ and tanh. Each LSTM cell stores information over time using three non-linear gate units. The three gates can learn and in turn decide how much information passes. By doing that, they can keep relevant information which is used for prediction.

$$\text{gates: } \begin{bmatrix} \tilde{c}_t \\ i_t \\ f_t \\ o_t \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(\begin{bmatrix} W_c \\ W_i \\ W_f \\ W_o \end{bmatrix} I_t + \begin{bmatrix} U_c \\ U_i \\ U_f \\ U_o \end{bmatrix} h_{t-1} + \begin{bmatrix} b_c \\ b_i \\ b_f \\ b_o \end{bmatrix} \right) \tag{3.1}$$

The depth structure of LSTM such as Stacked LSTM can improve the quality of the trained model [82, 93] by putting layers of LSTM where the output of one layer becomes the inputs of others. Each LSTM layer outputs a sequence of vectors, which will be used as an input to a subsequent LSTM layer, as shown in Figure 3.3 (right).

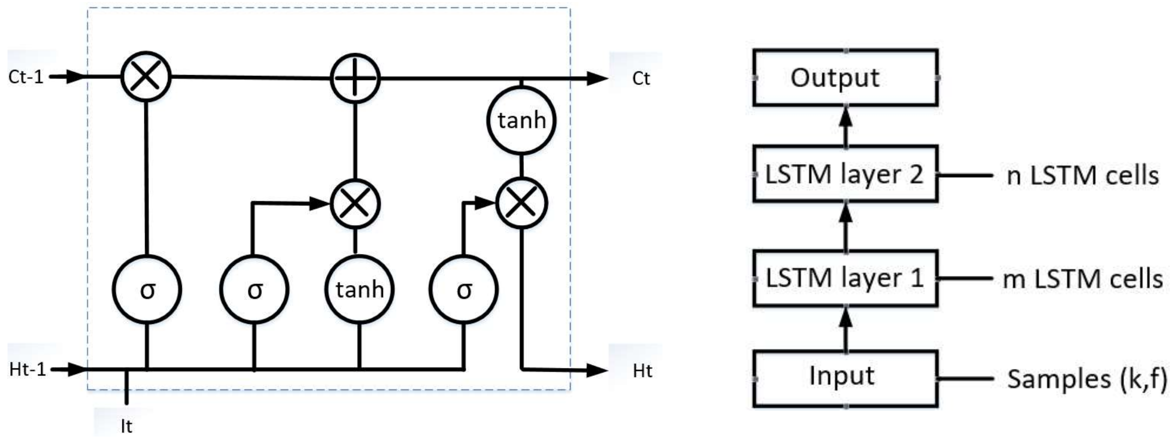

Figure 3.3. LSTM cell (left) and Stacked LSTM with two hidden layers (right)

Dataset fed to the Stacked LSTM divides into many samples containing f inputs (I_t) with k consecutive states. This feature is an improvement to keep previous data longer in order to forecast new output more accurately [94]. Stacked LSTM has two parts, the encoder that outputs a context vector (encoding) of the input sequence, which is then passed to the second part, the decoder to decode and predict the targets.

3.4 Training and Prediction models

Each set of collected raw volumetric deviations is used to estimate the geometric error parameters of the machine tool by a linearised pseudoinverse kinematic approach [64]. Figure 3.4 shows the machine's nominal kinematic model and Table 3.1 lists the measured geometric errors.

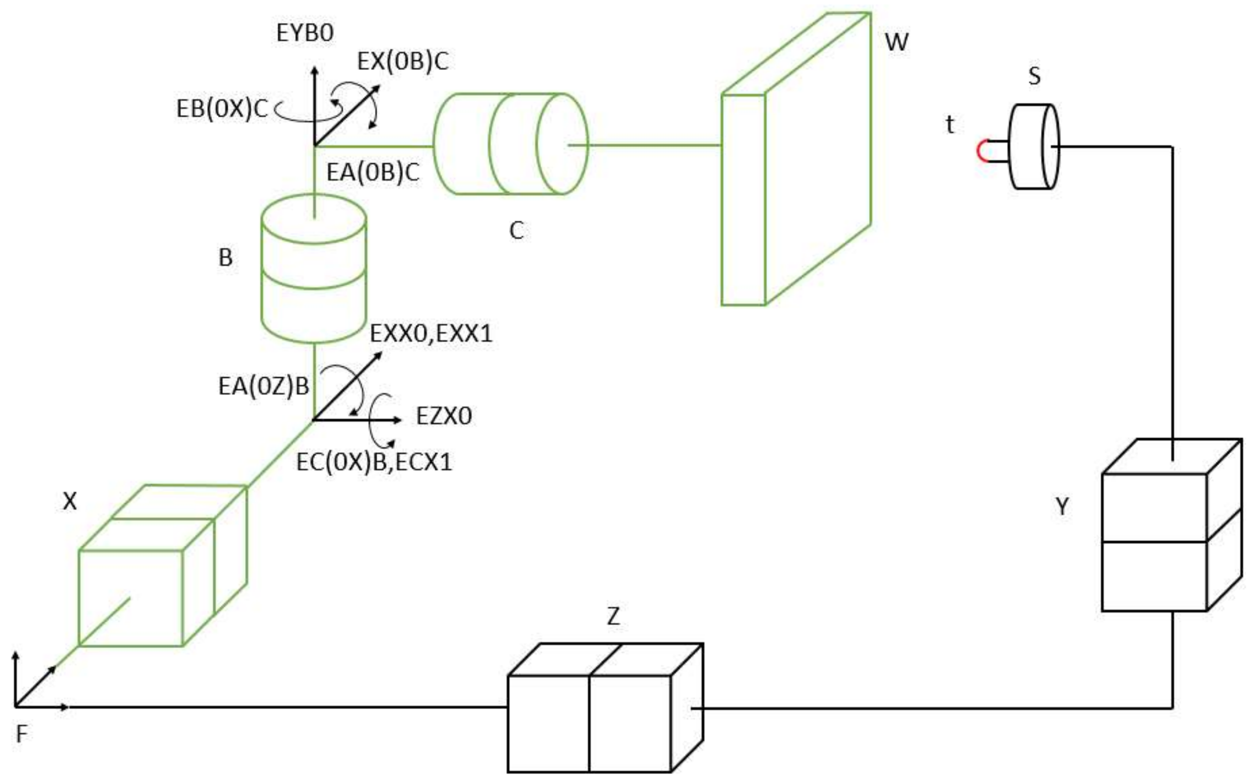

Figure 3.4. Machine schematic with thermally induced geometric errors in the work-piece branch (green color)

$$EP_{nx1}=J^{\dagger}_{mxn}EV_{mx1} \tag{3.2}$$

where: EP_{nx1}: geometric error parameters

EV_{mx1}: measured raw volumetric deviations

$J^{\dagger}_{mxn}$: pseudoinversed Jacobian sensitivity matrix

Table 3.1. Thermally induced geometric error parameters

Geometric error parameters	Description
EXX0	Zero degree (constant) term of linear positioning error motion in X-axis
EXX1	1st degree term of linear positioning error motion in X-axis; equivalent to a scale gain error
EYB0	Zero degree term of axial error of B axis motion in Y-axis; equivalent to a translational offset in Y of the B-axis
EC(0X)B	The out-of-squareness of the B-axis relative to the X-axis
ECX1	1st degree term of angular error of X axis motion around Z-axis; equivalent to a pitch error
EA(0Z)B	The out-of-squareness of the B-axis relative to the Z-axis
EA(0B)C	The out-of-squareness of the C-axis relative to the B-axis
EZX0	Zero degree term of straightness error motion of X axis in Z-axis, equivalent to a translational offset in Z of the X-axis
EB(0X)C	The out-of-squareness of the C-axis relative to the X-axis
EX(0B)C	The translational offset in X of the C-axis relative to the B-axis

In the training phase, each measured geometric error parameter is used as target and the measured motor powers of rotary axes, are used as inputs. Stochastic Gradient Descent (SGD), Root Mean Square propagation (RMSprop), and Adaptive moment (Adam), are the algorithms used to minimize the objective functions. In the testing phase, the optimized weights of LSTM obtained from the training phase process combine with the recorded powers to predict the thermal induced geometric error parameters as shown in Figure 3.5. Training and testing processes are programmed with Pytorch library in Python using torch.nn.LSTM recurrent layer and torch.optim. (SGD, RMSprop, Adam) optimizers.

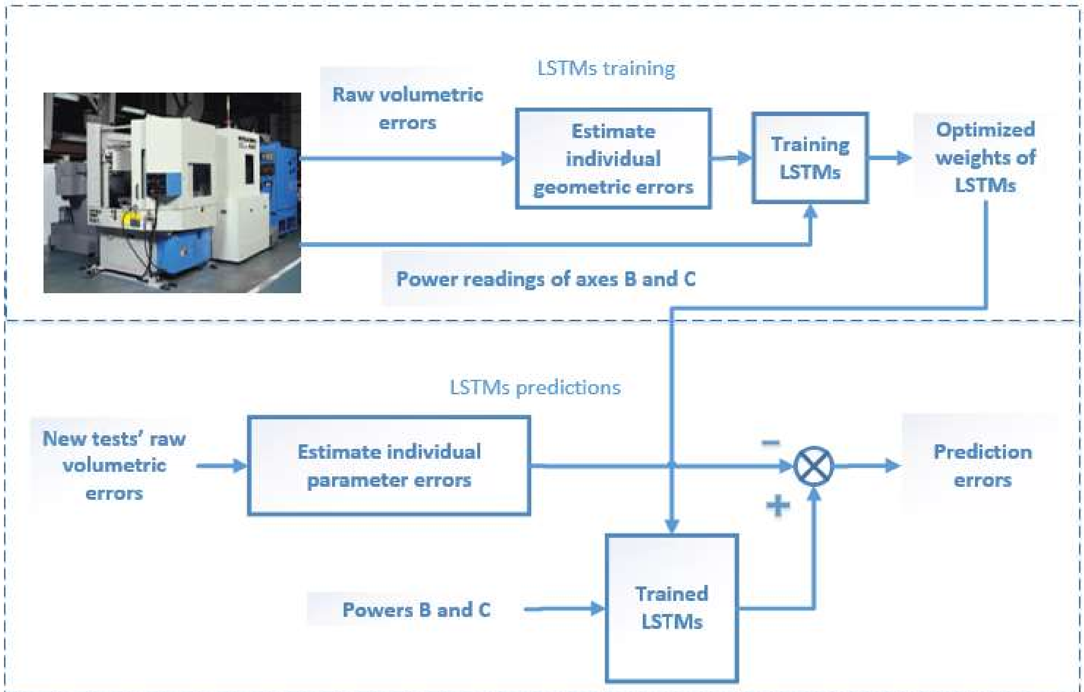

Figure 3.5. Optimized LSTM in training and testing phases

Figure 3.6 (left) shows the general diagram of the training process. The data sequences of power B or C are fed to the Stacked LSTM as samples. H_t is the predicted individual geometric error parameters (outputs) of the LSTM network. The optimizers consider the differences between H_t and $Target_t$ (measured error parameters) to update the parameters of the network.

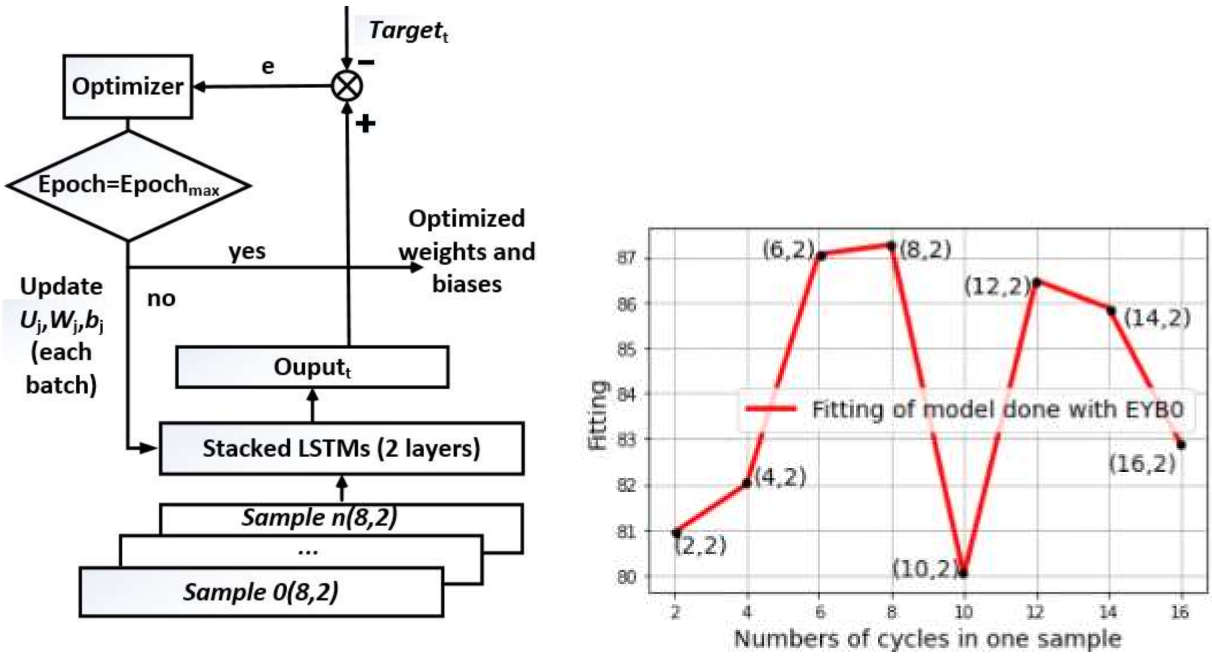

Figure 3.6. Training process with optimizers (left) and fitting of model with different size of sample (right)

The dimension (k, f) of samples are chosen to match with the number of LSTM cells in the two layers of the Stacked LSTM, batch size and number of epochs and keep as many previous information as possible. In this work, two hidden layers are chosen with m=48 and n=8 LSTM cells in layer 1 and 2, respectively. The optimal dimension of sample, obtained by trial and error, is (8, 2) leading to higher fitting of the model as compared with the other dimensions (Figure 3.6 (right)). This means using measured data of eight previous states to predict the output of the ninth state. Figure 3.7 and Figure 3.8 present the way to create the features from the data sequence in the training and testing processes.

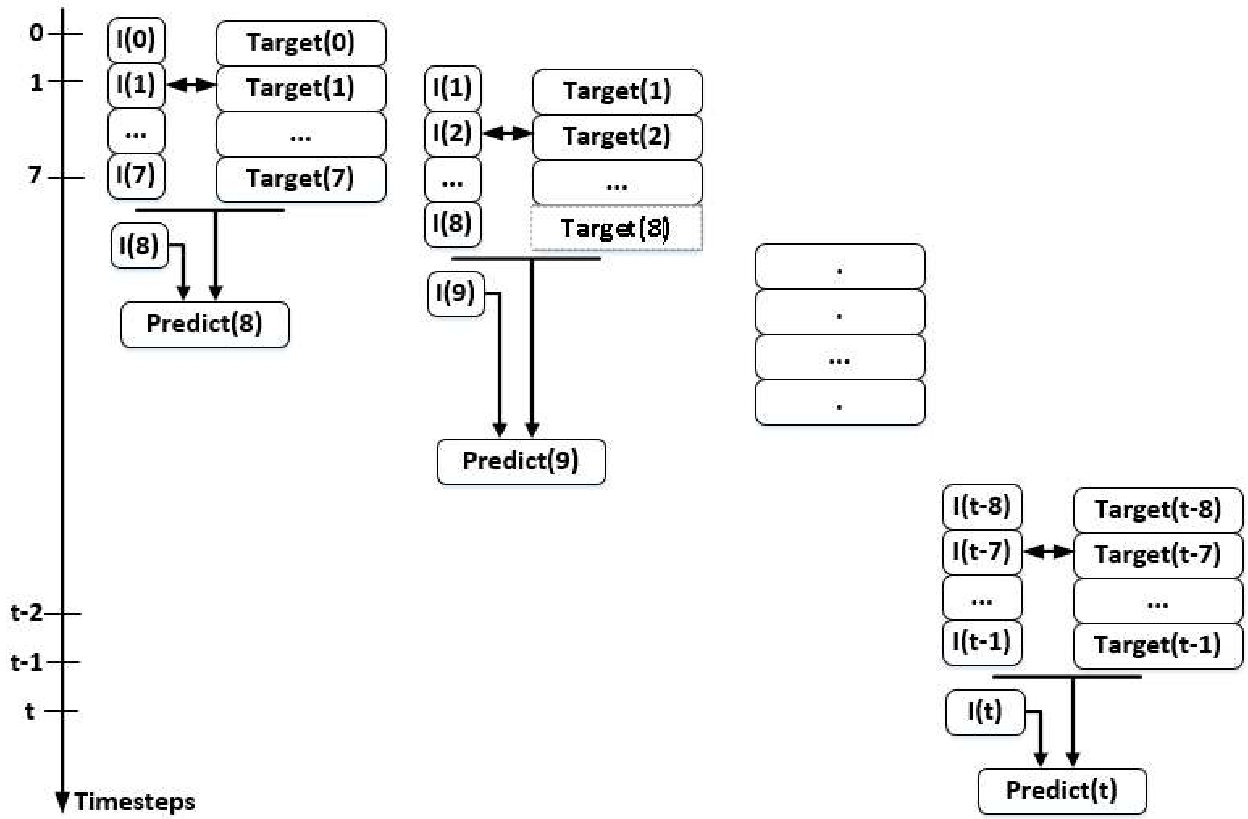

Figure 3.7. Multi features with inputs and targets in the training phase

Because the measured error parameters are not used as targets in the 40 hours (160 measurement cycles or time steps) of the testing process while using the same structure, the prediction values of previous cycles are used as shown in Figure 3.8. The first sample is set to zero to match the situation of the starting state. In the testing phase, the weights are not updated.

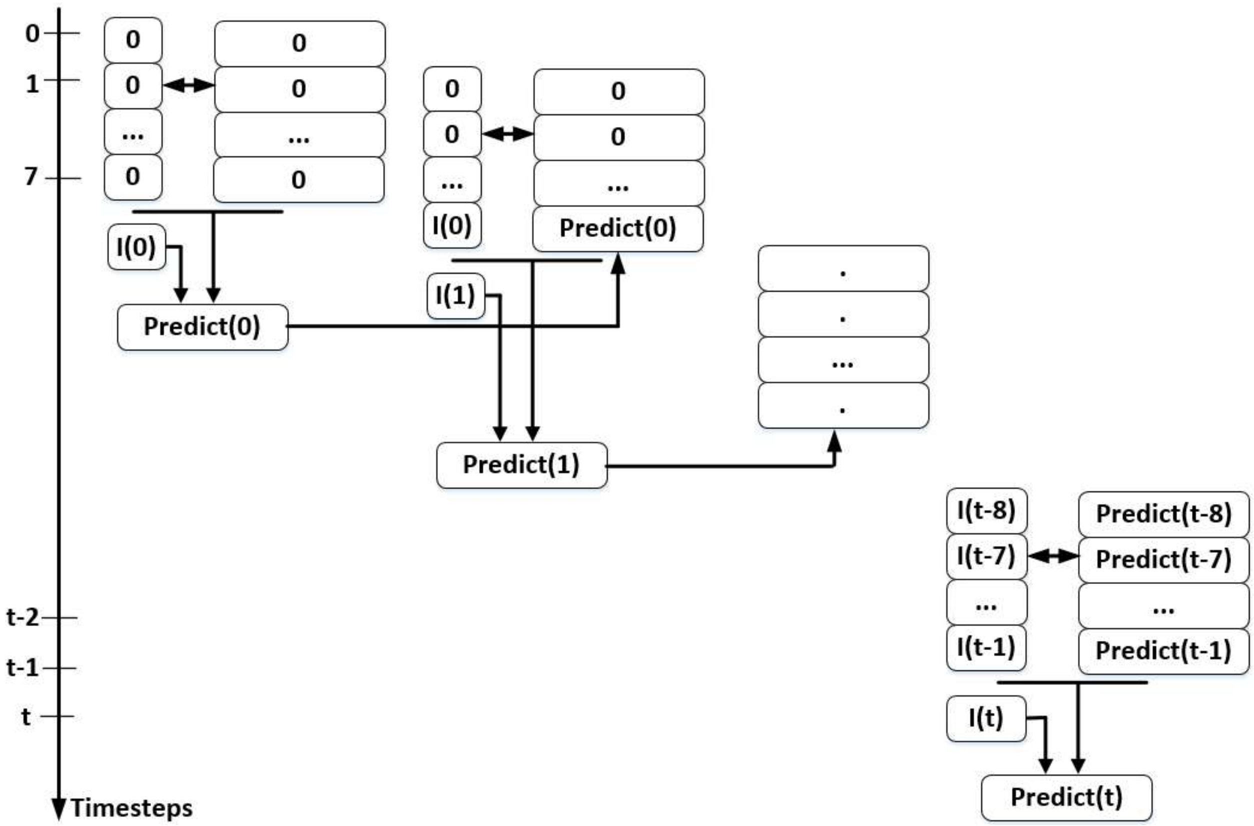

Figure 3.8.Multi features with inputs and outputs in the testing phase

For the back propagation, the internal parameters h_t, δW, δU, δb are calculated for updating the next state.

The optimizers are SGD, RMSprop and Adam, which update the following exponential moving averages of the gradient and squared gradient. The hyper-parameters $\beta_1,\beta_2 \in (0, 1)$ control the rates of exponential decay of these moving averages.

$$f_t = f(\beta, m_{t-1}, v_{t-1}, \delta, \delta^2) \tag{3.3}$$

Table 3.2 lists the updating weights and biases based on the SGD and RMSprop

Table 3.2. Updating weights and biases following algorithms SGD and RMSprop

	SGD	**RMSprop**
1	$W^{new}=W^{old}-\eta.\delta W^{old}$	$W^{new}=W^{old}-\frac{\eta}{\sqrt{E[\delta W^{old^2}]_{wt}+\epsilon}}\delta W^{old}$
2	$U^{new}=U^{old}-\eta.\delta U^{old}$	$U^{new}=U^{old}-\frac{\eta}{\sqrt{E[\delta U^{old^2}]_{ut}+\epsilon}}\delta U^{old}$
3	$b^{new}=b^{old}-\eta.\delta b^{old}$	$b^{new}=b^{old}-\frac{\eta}{\sqrt{E[\delta b^{old^2}]_{bt}+\epsilon}}\delta b^{old}$

with ϵ:smoothing term=$1e^{-8}$ and E_{wt}, E_{ut},E_{bt}:exponential moving average of squared gradients

Accumulating squared gradients in training can decrease the effective learning rate very fast and the training will stop at local optimization. Adam uses an exponentially decaying average of past squared gradients v_t like RMSprop and of past gradients m_t, which is same as momentum. Adam accumulates squared gradients in a window like RMSprop, instead of from the beginning of training. In addition, Adam can find the separate adaptive learning rates for each parameter from estimations of first and second moments of gradients.

The bias-corrected first and second moment estimate are calculated as follows:

$$\widehat{m}_t=\frac{f(\delta W,\delta U,\delta b,\beta_1)}{1-\beta_1^t} \tag{3.4}$$

$$\hat{v}_t=\frac{E(\delta W^2,\delta U^2,\delta b^2,\beta_2)}{1-\beta_2^t} \tag{3.5}$$

weights and biases are updated

$$W^{new}=W^{old}-\frac{\eta_w}{\sqrt{\hat{v}_{wt}+\epsilon}}\widehat{m}_{wt} \tag{3.6}$$

$$U^{new}=U^{old}-\frac{\eta_u}{\sqrt{\hat{v}_{ut}+\epsilon}}\hat{m}_{ut} \tag{3.7}$$

$$b^{new}=b^{old}-\frac{\eta_b}{\sqrt{\hat{v}_{bt}+\epsilon}}\hat{m}_{bt}. \tag{3.8}$$

The number of LSTM cells, the learning rates, the decay rates and the batch size are chosen after many simulations. In addition, Root Mean Square Error (RMSE) is used to determine the optimized weights of the networks when the deviations reach the preset values.

The RMSE is given by

$$RMSE=\sqrt{\frac{\sum_{j=1}^{n}(\text{Prediction values}_j-\text{Measured values}_j)^2}{n}} \tag{3.9}$$

Figure 3.9 shows the 30 RMSEs for the 30 training repetitions.

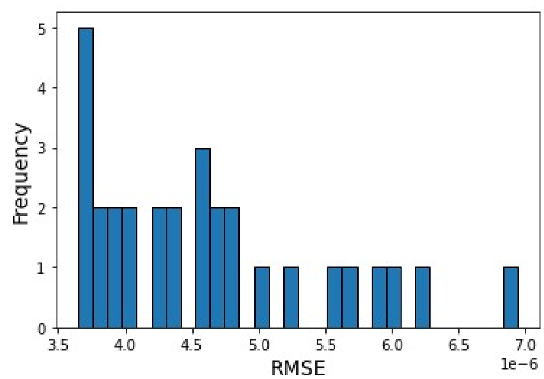

Figure 3.9. Training repetition RMSEs

The optimized weights, which are saved from the training network with the smallest RMSE, are applied for predicting the error parameters in the testing phase. The experiment for the testing process follows different heating and cooling sequential activities as shown in Figure 3.2.

3.5 Results and discussion

Figure 3.10, Figure 3.11 and Figure 3.12 show results for error parameter EXX1 in the training and testing processes.

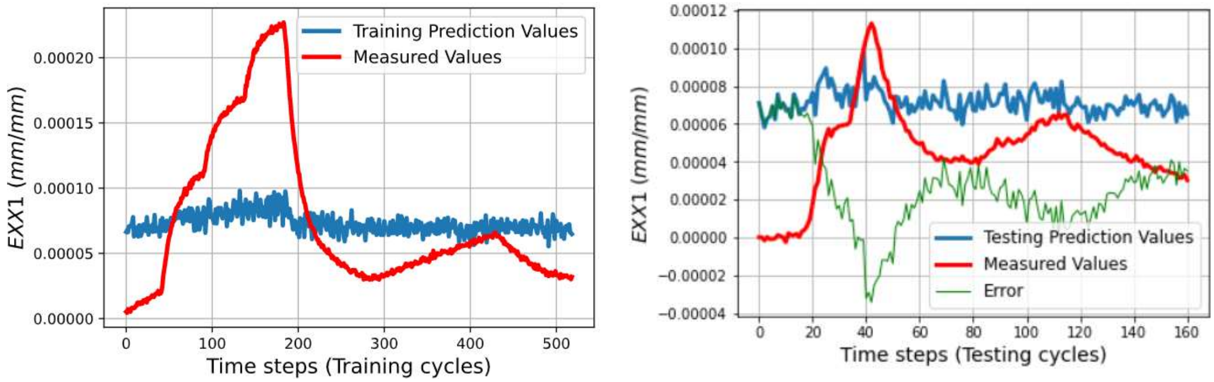

Figure 3.10. Training (left) and testing (right) processes for EXX1 by Stacked LSTM+SGD (1 cycle=15 minutes)

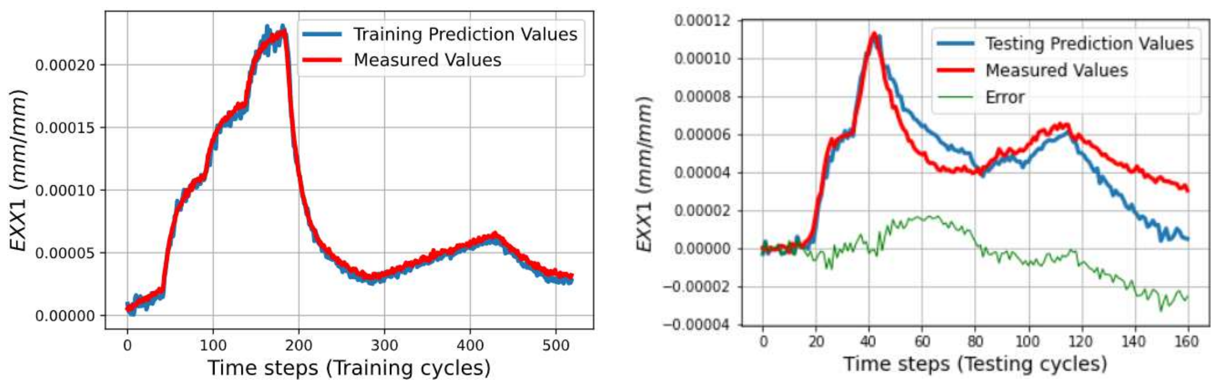

Figure 3.11. Training (left) and testing (right) processes for EXX1 by Stacked LSTM+RMSprop (1 cycle=15 minutes)

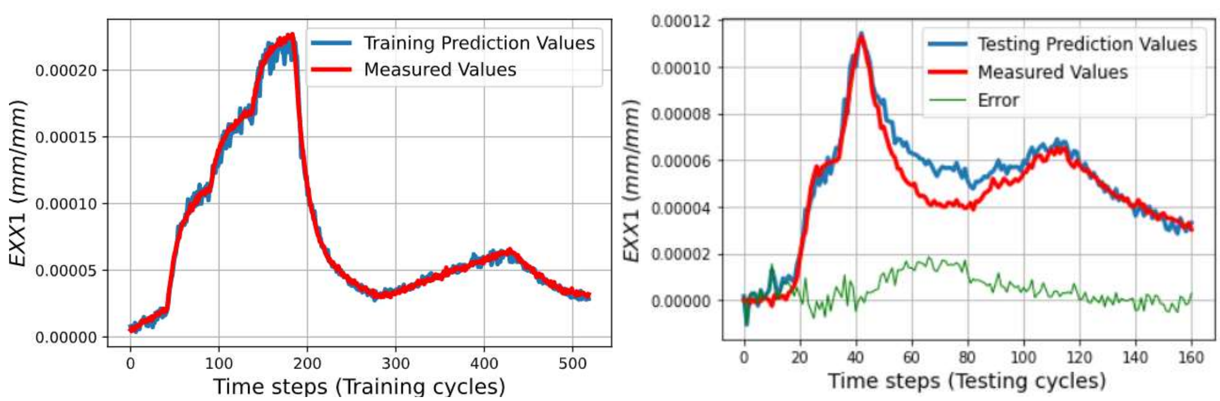

Figure 3.12. Training (left) and testing (right) processes for EXX1 by Stacked LSTM+Adam (1 cycle=15 minutes)

On the left of each figure is the result of the training process in 520 cycles (time steps) from cycle 9 to cycle 528. The reason the prediction values start from cycle 9 is mentioned in Figure 3.7. The result of prediction compared with measured values in the testing process is shown on the right and includes 160 time steps (measurement cycles). To compare the results between the Stacked LSTM+optimizers, the structure of Stacked LSTM and learning rate are similar. The differences between the three combinations of Stacked LSTM+optimizers are apparent. The quality of training and testing of LSTM+SGD is the worst. LSTM+RMSprop keeping the same learning rate for weights and biases but using past squared gradient improve the results. LSTM+Adam has the best capability with RMSE = $8.2*10^{-6}$ (fitting 93%). One key advantage is its ability to follow the measured values over longer periods without the need for updated measured values in the testing phase. The quality of training and predicting improves from the SGD to the RMSprop to the Adam optimizer.

The fitting is calculated as follows:

$$\text{fitting (\%)}=\left(1-\frac{\text{RMSE}}{\left|\text{Measured values}_{\text{jmax}}-\text{Measured values}_{\text{jmin}}\right|}\right)*100 \tag{3.10}$$

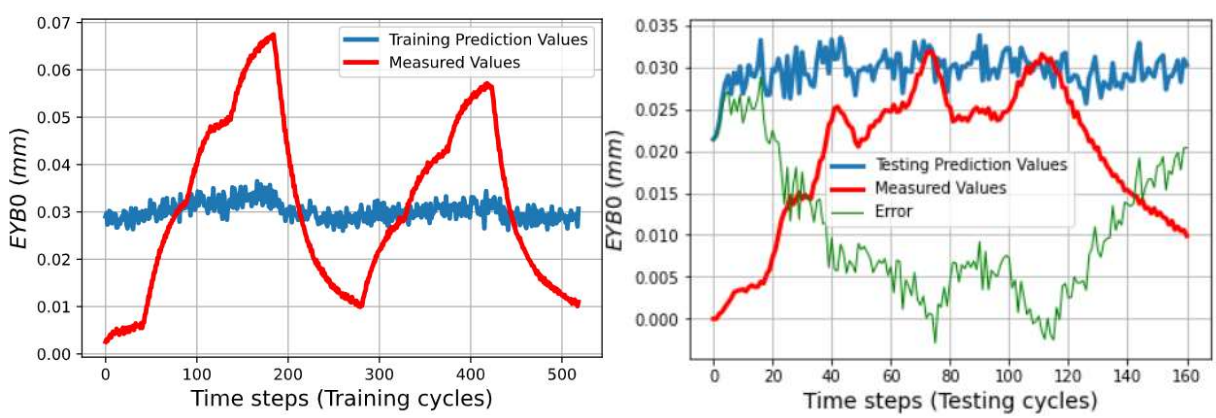

Figure 3.13. Training (left) and testing (right) processes for EYB0 by Stacked LSTM+SGD (1 cycle=15 minutes)

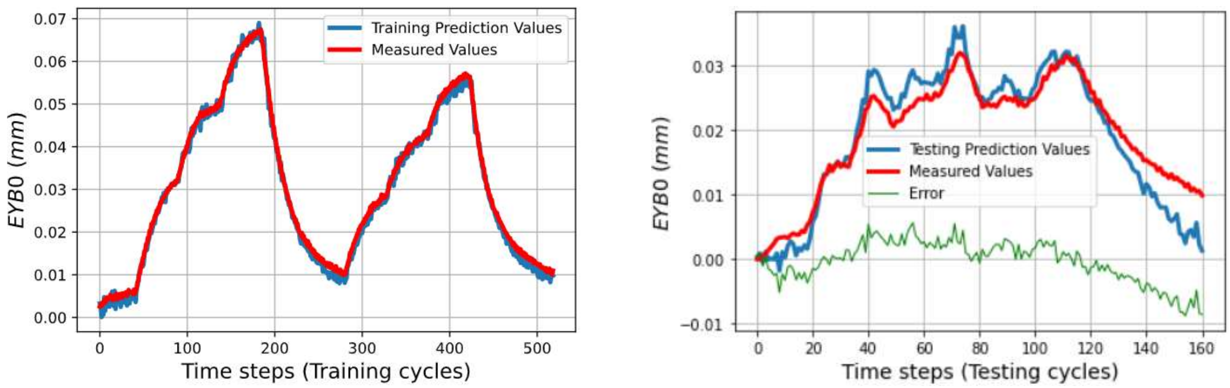

Figure 3.14. Training (left) and testing (right) processes for EYB0 by Stacked LSTM+RMSprop (1 cycle=15 minutes)

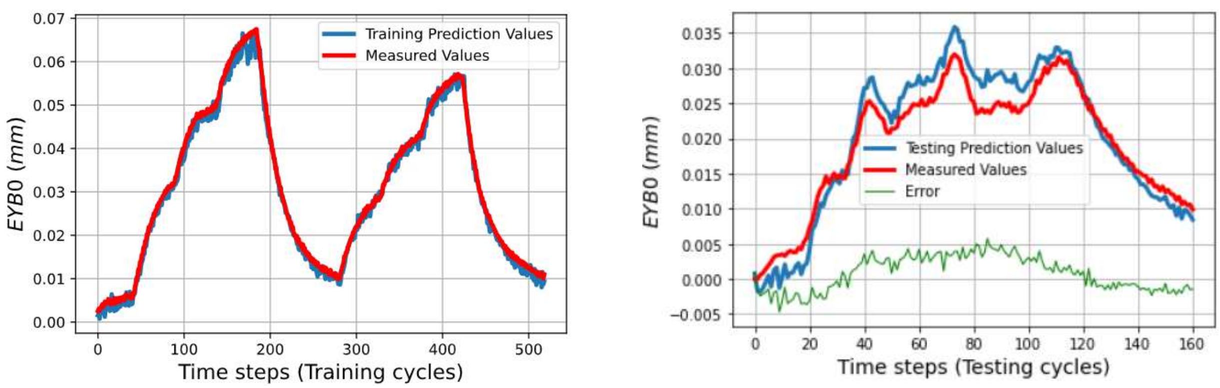

Figure 3.15. Training (left) and testing (right) processes for EYB0 by Stacked LSTM+Adam (1 cycle=15 minutes)

EYB0 was the parameter most affected by rotary axes activities [64]. The training and testing processes shown in Figure 3.13, Figure 3.14 and Figure 3.15 have similar initial conditions in ambiant temperatures and thermal error values. The model predicts the general trend of the error parameter both for separated and simultaneous B and C activities. In the testing phase, the predicted values can track the general trend of the measured values with the Adam and RMSprop optimizers. However, the drift comes sooner with RMSprop. The RMSE, calculated using Equation (3.9), of the testing phase with LSTM+Adam is $2.8*10^{-3}$ mm of the measured values (fitting=91%). In the following, only the results for the Adam optimizer are shown. The results for the others will be succinctly provided in Table 3.3 and Table 3.4.

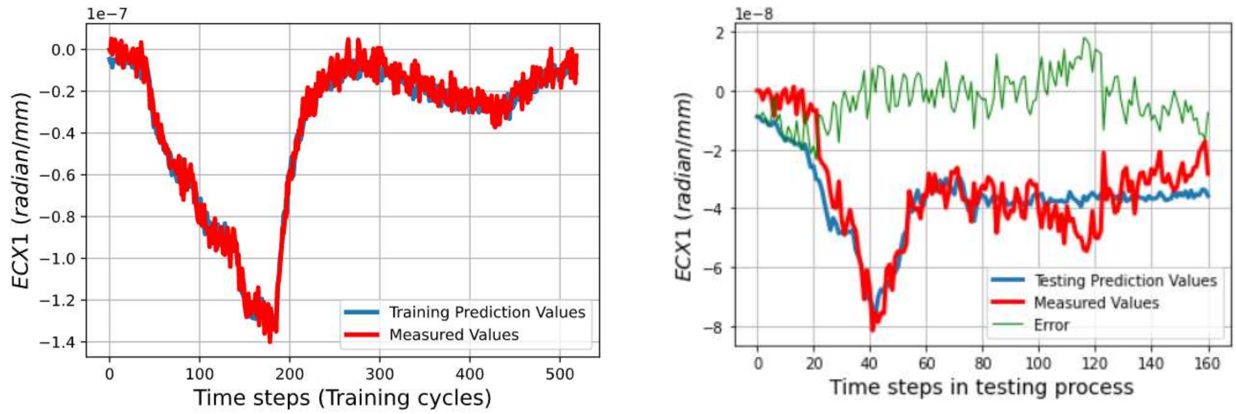

Figure 3.16. Training (left) and testing (right) processes for ECX1 by Stacked LSTM+Adam (1 cycle=15 minutes)

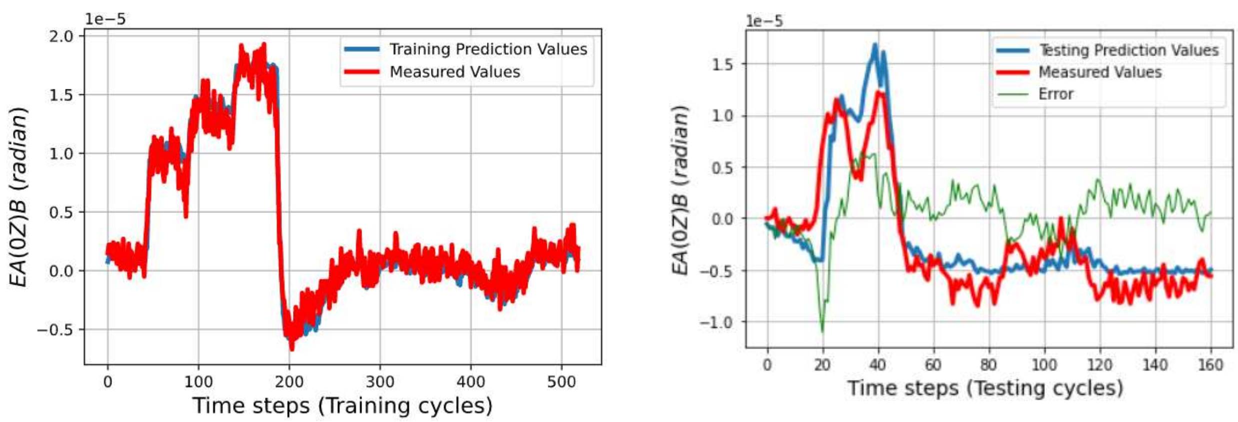

Figure 3.17. Training (left) and testing (right) processes for EA(0Z)B by Stacked LSTM+Adam (1 cycle=15 minutes)

Predicted values of EA(0Z)B in Figure 3.17 (right) tracked the output with RMSE=$2.6*10^{-6}$ (fitting=87%) for the testing process. EA(0Z)B is strongly influenced by activity of rotary axis B, while motions of axis C have much less effect on this error. This also happens for ECX1 (Figure 3.16).

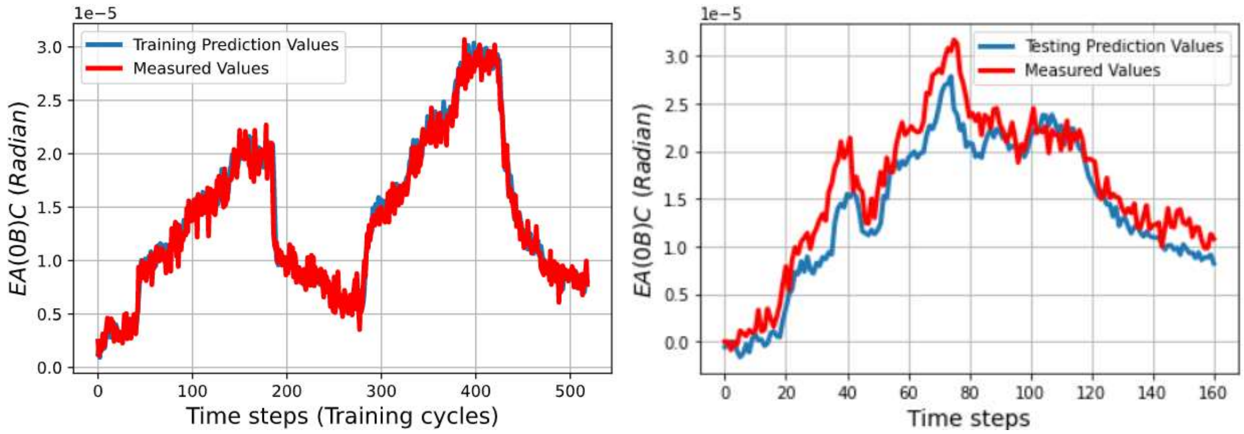

Figure 3.18. Training (left) and testing (right) processes for EA(0B)C by Stacked LSTM+Adam (1 cycle=15 minutes)

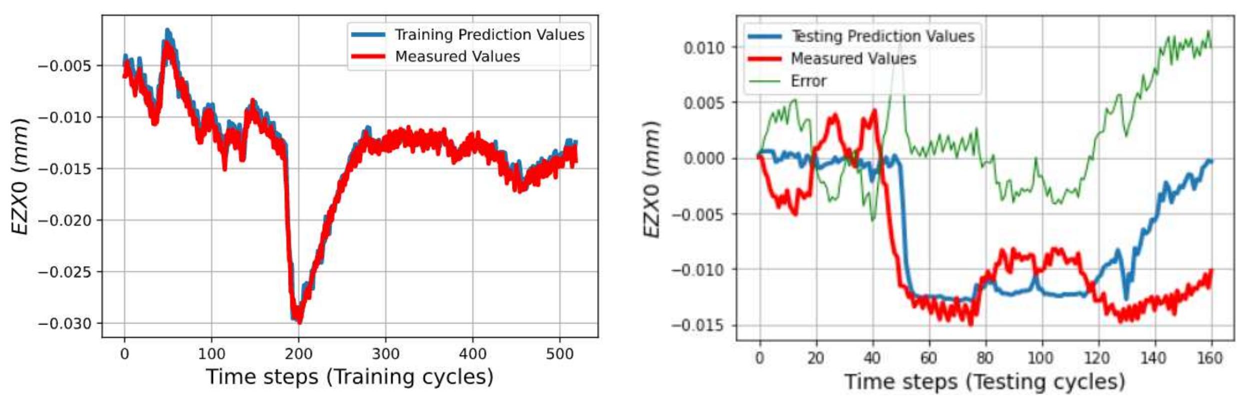

Figure 3.19. Training (left) and testing (right) processes for EZX0 by Stacked LSTM+Adam (1 cycle=15 minutes)

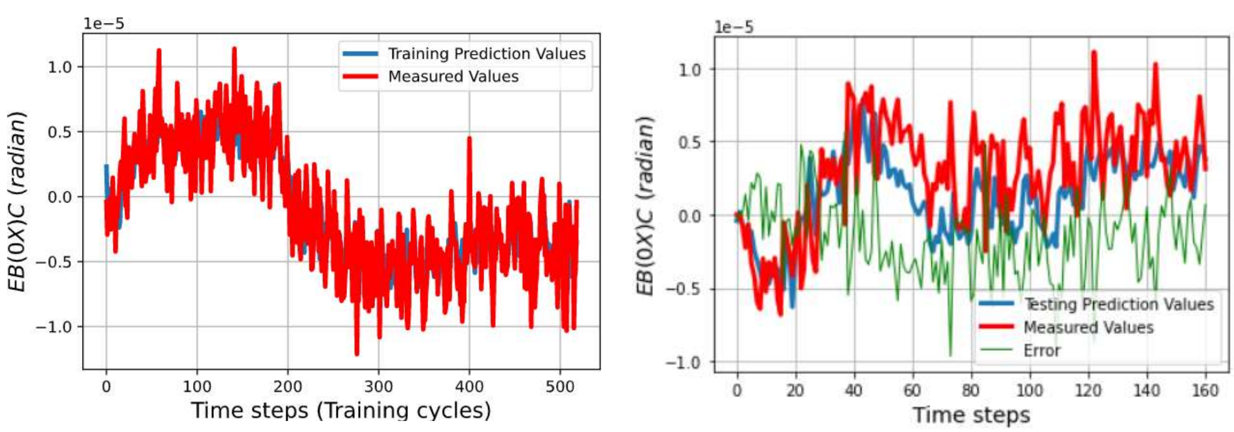

Figure 3.20. Training (left) and testing (right) processes for EB(0X)C by Stacked LSTM+Adam (1 cycle=15 minutes)

In the testing phase of EA(0B)C (Figure 3.18), EZX0 (Figure 3.19) and EB(0X)C (Figure 3.20), results still show that the model followed the changes of thermal error parameters, which supports the prediction capability of Stacked LSTM + Adam.

Table 3.3. Root Mean Square Errors (RMSE) for the testing process

Geometric error parameter	RMSE with LSTM+SGD	RMSE with LSTM+RMSprop	RMSE with LSTM+Adam
EXX0 (mm)	$2.6*10^{-3}$	$2.4*10^{-3}$	$2.2*10^{-3}$
EXX1(mm/mm)	$36.7*10^{-6}$	$12.6*10^{-6}$	$8.2*10^{-6}$
EYB0 (mm)	$14.1*10^{-3}$	$3.1*10^{-3}$	$2.8*10^{-3}$
EC(0X)B (radian)	$3.0*10^{-6}$	$2.7*10^{-6}$	$2.6*10^{-6}$
ECX1 (radian/mm)	$15.6*10^{-9}$	$11.2*10^{-9}$	$8.8*10^{-9}$
EA(0Z)B (radian)	$5.5*10^{-6}$	$4.8*10^{-6}$	$2.6*10^{-6}$
EA(0B)C (radian)	$8.5*10^{-6}$	$5.8*10^{-6}$	$3.0*10^{-6}$
EZX0 (mm)	$7.5*10^{-3}$	$5.5*10^{-3}$	$4.4*10^{-3}$
EB(0X)C (radian)	$5.1*10^{-6}$	$3.5*10^{-6}$	$3.1*10^{-6}$
EX(0B)C (mm)	$3.3*10^{-3}$	$3.2*10^{-3}$	$2.7*10^{-3}$

Table 3.4. The fitting values for the testing process

Geometric error parameter	Fitting with Stacked LSTM+SGD	Fitting with Stacked LSTM+RMSprop	Fitting with Stacked LSTM+Adam
EXX0 (%)	78	79	81
EXX1 (%)	68	89	93
EYB0 (%)	55	90	91
EC(0X)B (%)	79	80	81
ECX1 (%)	81	86	89
EA(0Z)B (%)	73	76	87
EA(0B)C (%)	73	82	90
EZX0 (%)	60	71	77
EB(0X)C (%)	71	81	83
EX(0B)C (%)	74	75	79

The range of model fitting varies from 55% (EYB0) to 81% (ECX1) with the LSTM+SGD, from 71% (EZX0) to 90% (EYB0) with the LSTM+RMSProp and from 77% (EZX0) to 93% (EXX1) for the LSTM+Adam, which has the best performance. The quality of prediction is maintained even after 40 hours of the testing process. The drift of prediction in the testing process of the main error parameter EYB0 comes later than in [64]. The fitting quality varies significantly between the optimizers. Adam has the best fitting performance. However, the training cycles is simpler than testing cycles, which may contribute to these differences. Future work will focus on finding other optimizers and more combined power training cycles.

3.6 Conclusion

A Stacked LSTM is investigated to model thermally induced geometric errors of a machine tool caused by activities on the workpiece side of rotary axes B and C. The input and output variables are the drive powers to the rotary axes and the machine geometric error parameters, respectively. Drive power is readily available from machine tool controllers whereas temperature sensors within the machine structure, which are the preferred input in thermal error prediction, are not so common. Three different optimizers, i.e. SGD, RMSprop, and Adam, are used with the Stacked LSTM. The trained LSTM is evaluated for their ability to predict the machine errors for a testing process significantly different from the training process. The Stacked LSTM+Adam significantly outperforms the other combinations to build model in the training process and to forecast error parameters in the testing process. The prediction capability without using measured error parameters as targets in 40 hours (160 cycles) exceeds 77% for all thermal error parameters, and reaches 93% for the dominant error EXX1. One limitation of this method is that there is no general rule to determine the optimal structure and parameters for Stacked LSTM except the trial and error method. Future works will focus on improving the prediction capability by applying other optimizers and training network with more sophisticated heating cycles, including simultaneous activity of all considered axes.

CHAPTER 4 ARTICLE 2: DEEP LEARNING TO DIRECTLY PREDICT COMPENSATION VALUES OF THERMALLY INDUCED VOLUMETRIC ERRORS

Huy Vu Ngoc*, J.R.R. Mayer, Elie Bitar-Nehme

Department of Mechanical Engineering, Polytechnique Montreal
Published in Journal Machines MDPI, Vol.11, Issue 4, 2023. Date of publication: 18/04/2023

Abstract: The activities of the rotary axes of a five-axis machine tool generate heat causing temperature changes within the machine that contribute to tool center point (TCP) deviations. Real time prediction of these thermally induced volumetric errors (TVEs) at different positions within the workspace may be used for their compensation. A Stacked Long Short Term Memories (SLSTMs) model is proposed to find the relationship between the TVEs for different axis command positions and power consumptions of the rotary axes, machine's linear and rotary axis positions. In addition, a Stacked Gated Recurrent Units (SGRUs) model is also used to predict some cases, which are the best and the worst predictions of SLSTMs to know the abilities of their predictions. Training data come from a long motion activity experiment lasting 132 h (528 measuring cycles). Adaptive moment with decoupled weight decay (AdamW) optimizer is used to strengthen the models and increase the quality of prediction. Multistep ahead prediction in the testing phase is applied to seven positions not used for training in the long activity sequence and 31 positions in a different short activity sequence of the rotary axes lasting a total of 40 h (160 cycles) to test the ability of the trained model. The testing phase with SLSTMs yields fittings between the predicted values and measured data (without using the measured values as targets) from 69.2% to 98.8%. SGRUs show performance similar to SLSTMs with no clear winner.

Keywords: volumetric errors; deep learning; machine tool; thermal errors

4.1 Introduction

Machine tool accuracy may degrade significantly due to thermal effects resulting partly from heat generated by the drives and the motions of rotary axes [64]. One way to re-duce these errors is real time compensation to cancel TCP deviation at an arbitrary point in the work space by adjusting its command position [95]. Mayr et al. stated that precise pre-dictions of thermal errors through models are fundamental for effective real time compensation [2]. The higher the predictability of the thermal error models, the more accurate the compensation will be.

Brecher et al. developed first and second order transfer functions between the thermal volumetric errors (TVEs) of a three-axis machine tool as outputs and the internal control data (rotational speeds and motor currents) and environment temperature as inputs [73]. Horejs et al. compensated thermal displacements in the y and z directions of the TCP through thermal transfer functions that were achieved using Matlab's System Identification Toolbox. The inputs are the temperatures from inside the main spindle, ram and base of the horizontal milling machine, ambient temperature and spindle speed and the out-puts are the thermal displacements [96]. Liu at al. modelled thermally induced volumetric errors based on the 21 geometric error components and nine thermal drift errors for milling and boring machine using rigid body kinematics. Only volumetric error in the z direction identified as the main error was compensated with Programmable Logic Controller and the machine's computer numerical controller (CNC) [97]. Mayr et al. found compensation values for individual thermal errors of rotary axis and main spindle from a gray box model using cooling power as input [62]. Yu et al. investigated the dependence of volumetric errors on spindle speed and temperature at pre-defined points through polynomial regression model. The model with closest predictability to the measured outputs is chosen [72]. Bitar-Nehme and Mayer developed first order transfer functions with delay terms to predict individual machine geometric errors as output from power consumptions as input. The machine table thermal expansion was also predicted [64]. Baum et al. modeled thermally induced volumetric error from the 21 geometric errors components for a three-axis machine tool through rigid body kinematics. Error components were measured with integral deformation sensors in the training process and the R-test based procedure in the testing process [98]. Liu et al. established a mechanism-based error model to identify three thermally induced error terms of linear axes X, Y, and Z as functions of time and positions. With spindle system, they modelled relationship between two angular thermal errors yaw and pitch, two thermally translational errors in x and y directions, and spindle elongation with temperatures of critical key locations by multivariate linear regression analysis [99].

The above models show their ability to predict the TVEs (data sequence) of the ma-chine tools. However, the volume and variety of data become bigger; the model has more clues to predict outputs better, but the calculation cost and internal parameters increase. In some cases are over the capability of manual analysis [77, 100]. Elaborate training of the model is expected to be able to guarantee reliable results in the long term [101]. Deep learning technique enables learning complex relations and is a new approach for these problems. LSTM and GRU networks have attracted attention in recent time and Liang et al. proved that they have a better quality than Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for processing sequence data and avoiding overfitting problem [83]. Ngoc et al. [100] applied LSTM to predict multi step ahead data of thermally induced geometric errors, which define the errors between and within individual machine axes, from activity cycles involving the two rotary axes B and C of a five-axis machine tool. Others [84-87] presented the efficiency of LSTM networks for modelling the thermal behaviors of a machine tool. [84, 85] modeled the thermal elongation of spindle as a function of rotational speed. [86] used thermal elongation of spindle in the z direction as output and temperature at key locations as inputs. [87] found the relationship between thermal errors of ball screws and temperature of the feed drive system. While in [100] thermally induced geometric errors (TGEs) were predicted using SLSTM, the current paper applies SLSTM and SGRU to directly predict the TVEs without the use of a rigid body kinematic model. This avoids having to develop the machine kinematic model and may be able to model error sources that a kinematic model may not consider. Calculating them from predicted TGEs does not consider errors that cannot be explained with the estimated geometric errors [64, 102]. TVEs directly relate to compensation values of the thermal errors of paths to move the tool to target positions in the working space of the machine tool. These predictions are more difficult than doing TGEs because the predictors need to realize not only the change of thermal behavior of machine, but also the positions we want to move the tool to. Therefore, the input data include not only the power consumptions of axes B and C, but also the machine linear and rotary axis positions. Increasing the number of inputs also increases the calculation cost. With a multivariate and multi-layer network, it may be trapped within multiple local minima; Sagheer and Kotb suggest having a suitable optimizer to facilitate the training of deep learning models [31]. Stochastic optimization is successfully applied in Deep learning [103] and Adam [43] is a popular optimizer. Adaptive moment with decoupled weight decay (AdamW) optimizer [104] is a modification of Adam with flexible learning rate, decoupling the weight decay from the gradient update, is expected to make the networks converge faster and improve accuracy. The proposed scheme is presented in six sections: Section 4.1 reviews previous works. Section 4.2 describes the experimental process. Section 4.3 presents the neural network configurations of the SLSTMs and SGRUs models. Section 4.4 shows the training and predicting processes. The results and discussion are in Section 4.5 and the conclusion follows in Section 4.6.

4.2 Experimental process

Motion sequences for machine axes were designed to warm up the machine tool without any machining [64, 105], while measuring the TVEs, or TCP displacements between a contactless R-test device (sensor nest) [106] and ball artefacts on a Mitsui-Seiki HU40T horizontal machining center with wCBXfZY(S)t topology as described in [64]. Figure 4.1 illustrates the experimental setup. Four separate artefact balls are fixed on stems with different lengths and a scale bar made of Invar with two balls separated by a calibrated distance on the work-piece side, while the sensor nest is mounted in the spindle on tool side.

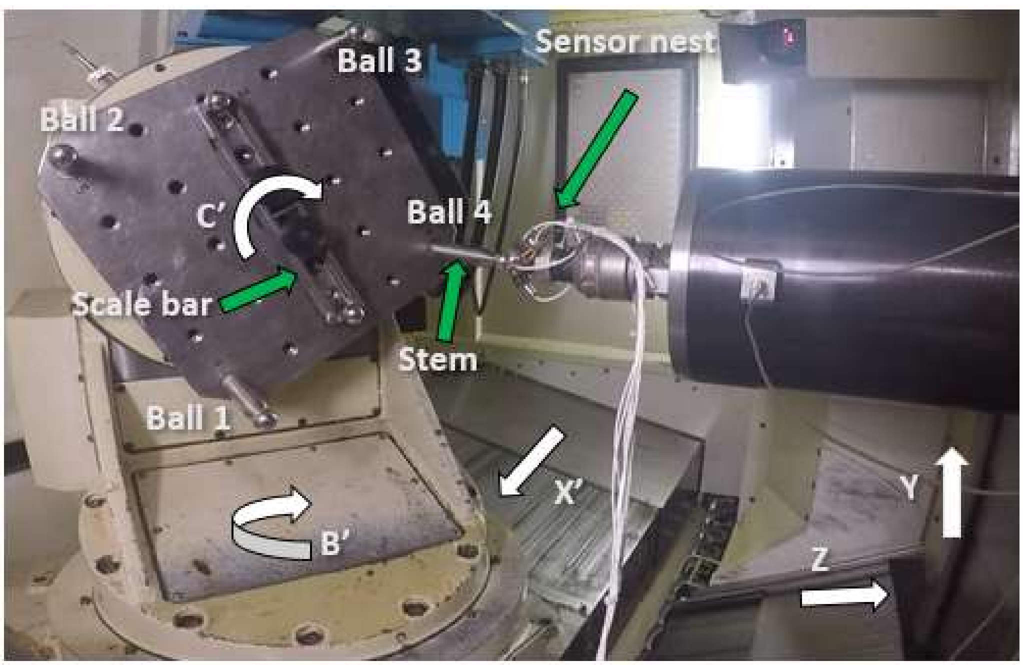

Figure 4.1. Experimental setup.

Heating (rotary axes' motions at different speeds) and cooling (machine stopped) cycles are shown in Figure 4.2 (on the left). Figure 4.2 (on the right) illustrates ambient temperature controlled between 22.1 and 23.5 Celsius degrees in 132 h (528 cycles) training process.

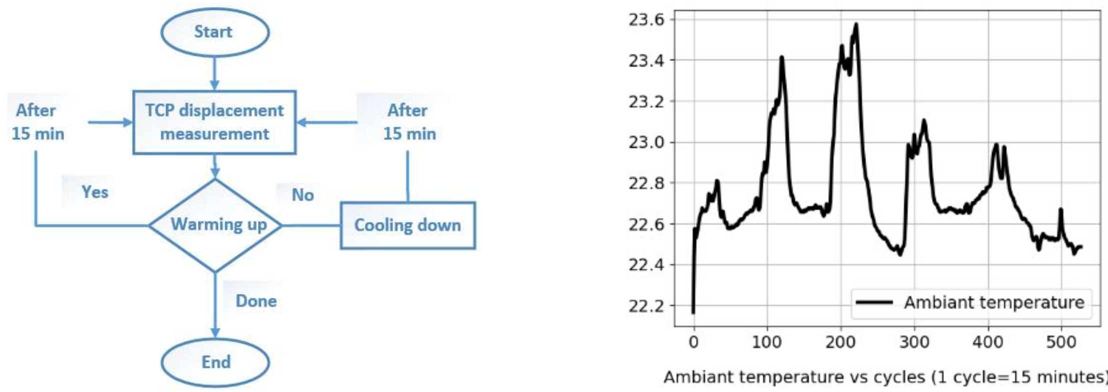

Figure 4.2. Heating and cooling processes and ambient temperature.

The activity sequences are implemented using a machine tool G-code. TCP positions are measured at 15 min intervals for both training and testing processes. TVEs**,** estimated by the Scale And Master Ball Artefact (SAMBA) method [107] in both processes in the x, y, and z directions, are the subtractions of the initial value from all subsequent values to avoid the effect of machine quasi-static geometric errors as given by Equation ().

$$\mathbf{TVE} = \mathbf{VE} - \mathbf{VE}_0 \qquad (4.1)$$

Figure 4.3 on the left shows the long process of activity with different measured powers of the B and C rotary axes lasting 132 h. In this process, each axis is exercised separately to capture their individual effect on the behavior of the machine tool. Figure 4.3 on the right shows the short process lasting 40 h with individual and simultaneous motions of rotary axes, which was run after some days and with similar starting condition to the long process. The short process is different from the long process and includes individual axis motions as well as combined axis motions as would be likely during a machining process.

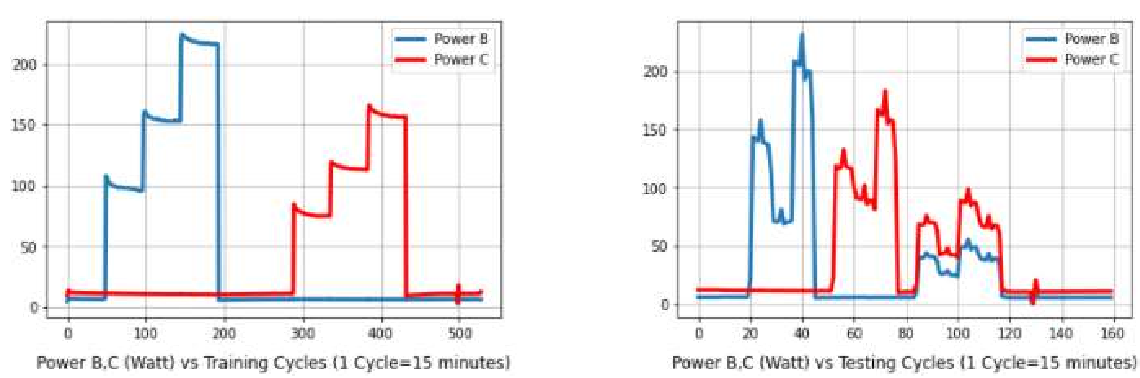

Figure 4.3. Long (left) and short (right) cycling processes [100].

4.3 Stacked LSTMs and GRUs

An LSTM unit has three gates: an input gate (i_t), an output gate (0_t) and a forget gate (f_t) as shown in Figure 4.4 (on the left).

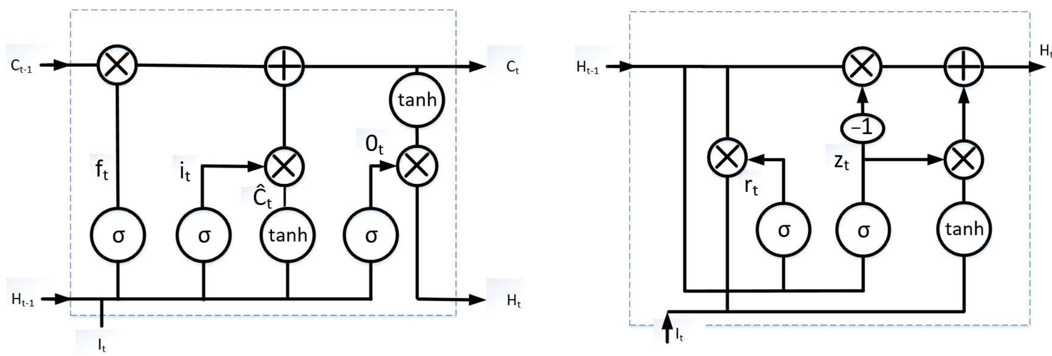

Figure 4.4. LSTM and GRU unit.

The function of the LSTM unit is to remove or add information to the cell state C_t that is carefully regulated by these three gates. $\widehat{C}_t$ represents new information that can be applied to cell state C_t. H_t is the hidden state; σ and tanh are sigmoid and tangent activation functions, respectively.

Gated recurrent unit (GRU) [37] is also working by a gating mechanism. GRU unit contains a reset gate r_t and an update gate z_t. GRU does not have an output gate O_t and long-term memory unit $\widehat{C}_t$ as the LSTM unit. Both of them have the ability to eliminate the vanishing gradient problem by keeping the relevant information and passing it to the next time steps of the network.

SLSTM/SGRUs in Figure 4.5 consists of stacked layers of LSTMs/GRUs and the output layer makes a prediction [31]. The SLSTMs/SGRUs run when the first LSTM/GRU layer takes the input sequence and every LSTM/GRU layer feeds hidden states to the next LSTM/GRU in the stack. SLSTMs/SGRUs calculates only one error signal at the final output layer, and then back propagates it through all previous layers [31]. This structure is programmed with Pytorch library in Python using torch.nn.LSTM/GRU.

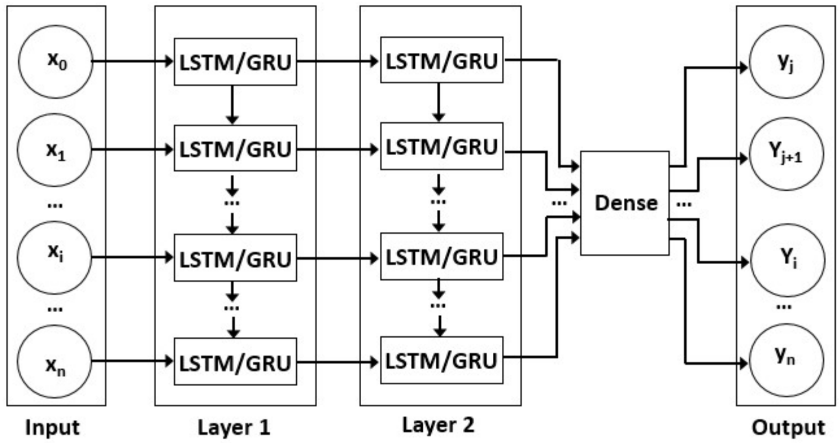

Figure 4.5. Stacked LSTMs and GRUs structure.

4.4 Training and testing processes

The machine schematic in Figure 4.6 shows TVEs measured at four artefact balls. The workpiece and tool branches are illustrated in green and black colors, respectively.

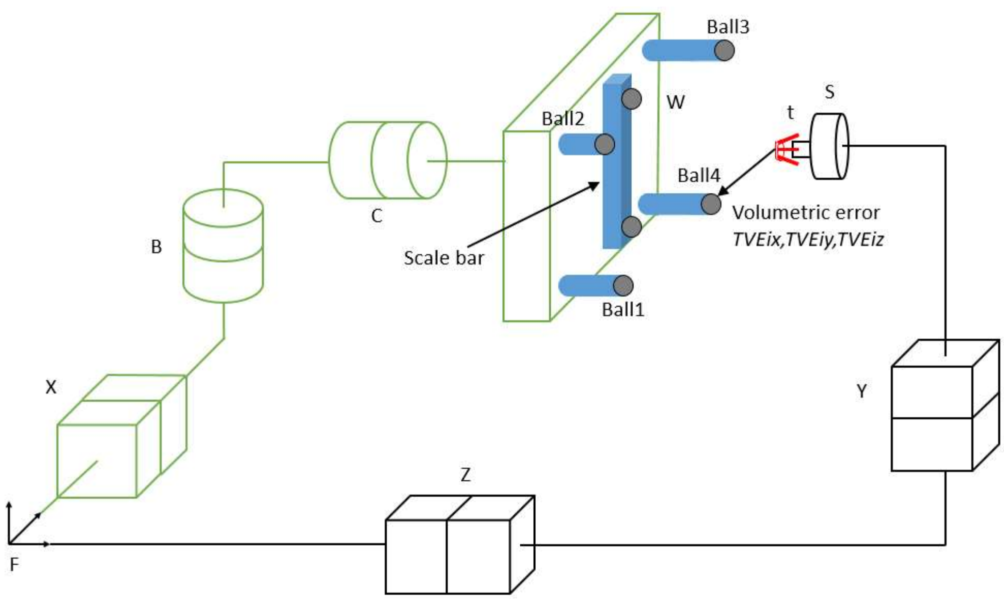

Figure 4.6. Machine schematic with thermally induced volumetric errors.

The TVEs have three positioning error components as described in Equation (4.2). These error components may change with activities of the B- and C-axis, positions, and indexations.

$$\mathbf{TVE_i} = [TVE_{ix}\ TVE_{iy}\ TVE_{iz}]^T \text{ with } i = 1{:}N \tag{4.2}$$

where N is the number of measured positions.

In the training phase, the SLSTMs/SGRUs models are built using long process data, as illustrated in Figure 4.7, and TVE*s* in 24 positions of the artefact balls 1, 2, and 3, shown in Table 4.1, are used as targets. The power consumptions of the axes B and C, the 24 linear axis positions used to move the sensor nest to the ball artefacts and the nine (B, C) indexations, or rotary axis positions, are used as inputs.

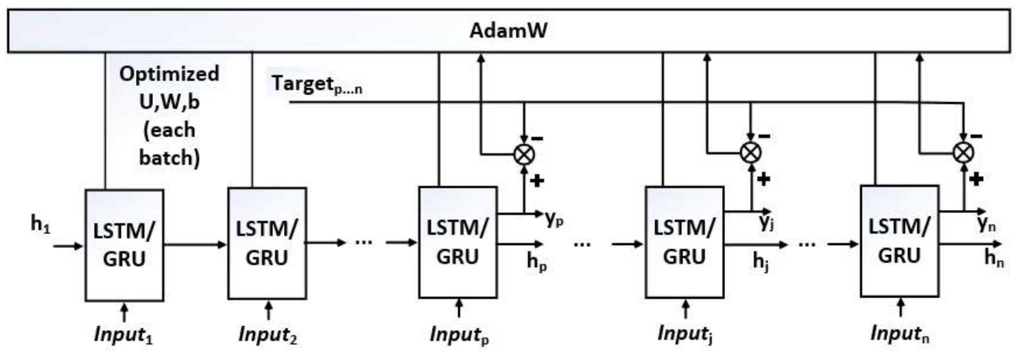

Figure 4.7. Flow chart of the training process.

The testing phase will consider two cases: long test data consisting of TVE*s* at seven positions of artefact ball 4 as well as the short test data of all 31 positions at all four artefact balls.

Table 4.1. 31 positions with nine different indexations of balls 1, 2, 3, and 4.

Positions of Artefact Balls				Indexations A,B,C (Degrees)		
$P_{1,1,1}$	$P_{1,2,2}$	$P_{1,3,3}$		0	−90	135
$P_{2,4,1}$	$P_{2,5,4}$	$P_{2,6,3}$		0	−90	−45
$P_{3,7,3}$	$P_{3,8,4}$	$P_{3,9,1}$	$P_{3,10,2}$	0	−30	−60
$P_{4,11,1}$	$P_{4,12,2}$	$P_{4,13,3}$	$P_{4,14,4}$	0	0	180
$P_{5,15,2}$	$P_{5,16,3}$	$P_{5,17,4}$	$P_{5,18,1}$	0	0	0
$P_{6,19,3}$	$P_{6,20,4}$	$P_{6,21,1}$	$P_{6,22,2}$	0	20	15
$P_{7,23,2}$	$P_{7,24,3}$	$P_{7,25,4}$		0	90	45
$P_{8,26,3}$	$P_{8,27,2}$	$P_{8,28,1}$		0	90	−45

$P_{9,29,4}$	$P_{9,30,1}$	$P_{9,31,2}$	0	90	−135

where: $P_{q,i,k}$ with q = indexation (1 to 9); i = linear axis position (1 to 31); k = artefact (1 to 4).

The 31 linear axis positions are shown in Figure 4.8. These positions are pre-determined by calculations based on the indexations of B and C axes and the artefact's positions in the machine frame. They are used for the G-code to move the sensor nest to the four artefact balls.

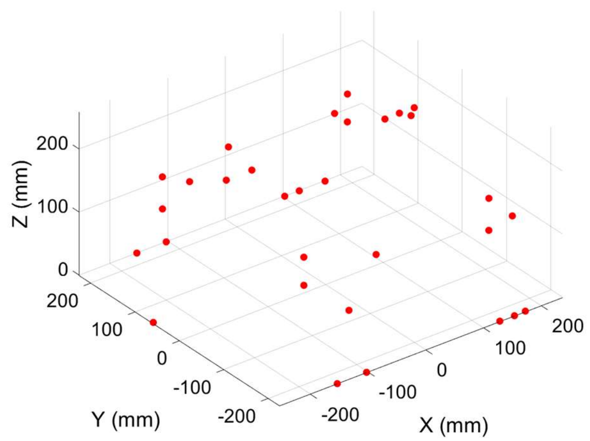

Figure 4.8. 31 linear axis positions at four artefact balls in the machine frame.

SLSTMs/SGRUs [33] are capable of capturing longer patterns in sequential data. AdamW optimizer is used to generate the optimized weights and biases based on the outputs of the SLSTMs\SGRUs and the measured TVEs at 24 different positions. In Figure 4.7, $\text{Input}_j = \left[\text{PowerB}_j, \text{PowerC}_j, x_j, y_j, z_j, b_j, c_j\right]^T$ with j = 1: n; p is the number of inputs in one sample = 8; n is the number of training data = 528 × 24 = 12,672, corresponding to 24 position measurements or observations at each of the 528 measurement cycles;

Target_j is the desired y_j of SLSTMs/SGRUs equal to the measured TVEj; $\left[y_p,\ldots,y_n\right]$ = trained $\left[\text{TVE}_p,\ldots,\text{TVE}_n\right]$; U, W and b are weight and bias that need to be updated after each batch (number of training samples) in the training process as Equations (4.3) to (4.5) [45]

$$\mathbf{W}^{new} = (1-\lambda_w)\mathbf{W}^{old} - \frac{\eta_w}{\sqrt{\hat{v}_{wt}+\epsilon}}\widehat{m}_{wt} \tag{4.3}$$

$$\mathbf{U}^{new} = (1-\lambda_u)\mathbf{U}^{old} - \frac{\eta_u}{\sqrt{\hat{v}_{ut}+\epsilon}}\widehat{m}_{ut} \tag{4.4}$$

$$\mathbf{b}^{new} = (1\text{-}\lambda_b)\mathbf{b}^{old} - \frac{\eta_b}{\sqrt{\hat{v}_{bt}+\epsilon}}\widehat{m}_{bt} \quad (4.5)$$

where η is the learning rate, $\epsilon = 10^{-8}$, $\widehat{m}$ and $\hat{v}$ are bias-corrected first and second moment estimates and λ is the rate of the weight decay per step.

The diagram in Figure 4.9 shows the testing procedure applying multistep predictions with the trained SLSTMs/SGRUs and inputs without using measured TVEs as targets. In this phase, to match the initial conditions, eight first states with values of zeros are added to the data sequence, as Ngoc et al. conducted this in [100].

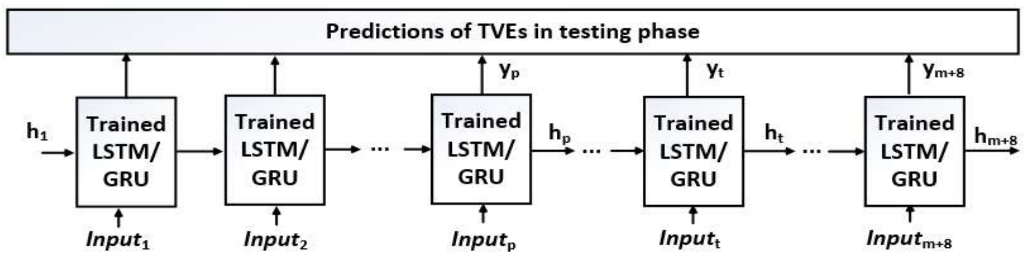

Figure 4.9. Prediction process of thermally induced volumetric errors.

In Figure 4.9: $\mathbf{Input}_t = [\text{PowerB}_t,\text{PowerC}_t,x_t,y_t,z_t,b_t,c_t]^T$, t = 1 to m+8, p = 8, m = 528 and 160 cycles in the long and short processes, respectively; and $[y_p \ldots y_{m+8}]$ = Predicted $[\text{TVE}_p \ldots \text{TVE}_{m+8}]$.

4.5 Results and discussions

In this section, the prediction ability of the trained SLSTMs for all TVEs are shown before some the best and worst cases are chosen to compare with the SGRUs'. In the training phase, three Stacked LSTMs are used to build the models for TVEx, TVEy and TVEz separately. Data of the thermal behaviors of the machine in 24 positions (31 positions of Table 4.1 less positions 5, 8, 14, 17, 20, 25, and 29 of ball 4) were put in series. Figure 4.10 shows the measured and predicted thermal volumetric errors (TVEs) for each of the 24 balls' positions where for each position, 528 measuring and prediction cycles are conducted. The results for each position are concatenated horizontally, so the timeline of the abscissa repeats for every position (Pi) for compactness of presentation.

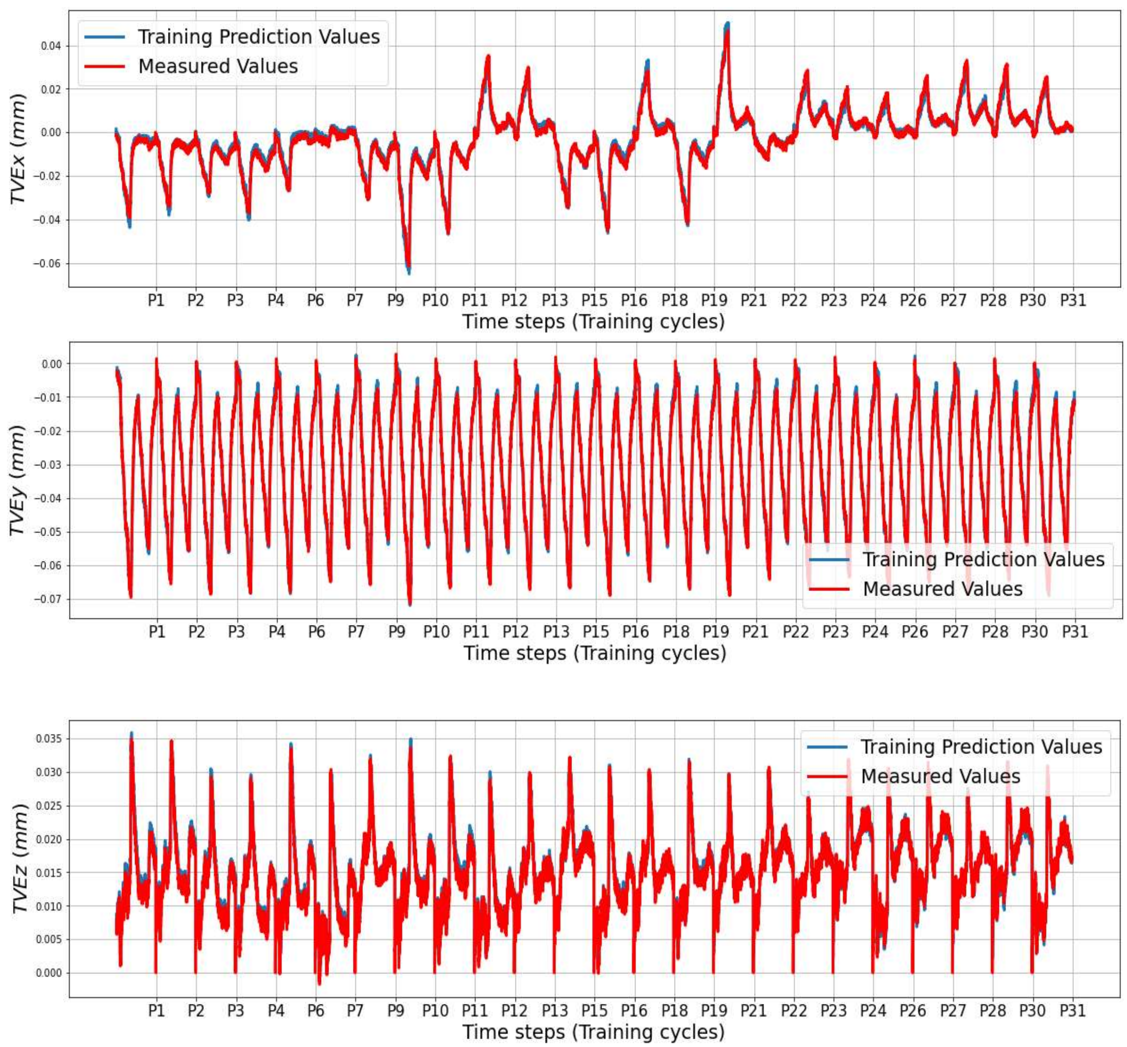

Figure 4.10. Training TVEx, TVEy, and TVEz from 24 positions in the long training process. For example, between P2 and P3 are the 528 cycle measured and predicted thermal volumetric errors at position P3 (1 cycle = 15 min).

The B axis has a greater impact on volumetric errors (TVEs) than the C axis. In fact, B axis activity dominates almost all of the TVEx. However, the influence of C axis activity relative to the B axis increases for TVEy and TVEz.

In the testing phase, three sets of volumetric errors were considered to test the ability of the trained model: data of seven positions at ball 4 in the long process (different positions in workspace but same process with training data), and data of 31 positions in the short process which is divided in two sets: the first set with 24 positions at balls 1, 2, 3 (same positions but different process) and the second set with seven positions of ball 4 (different positions and process). Due to the limitation of space, some of the best and the worst predictions of the TVEs in each set are presented. Multistep predictions (528 steps (cycles)) for positions P14 and P20 in the long process are computed and illustrated from Figure 4.11 to Figure 4.13. In the short process, 160 prediction steps are shown from Figure 4.11 to Figure 4.19 for positions P4, P5, P9, and P25.

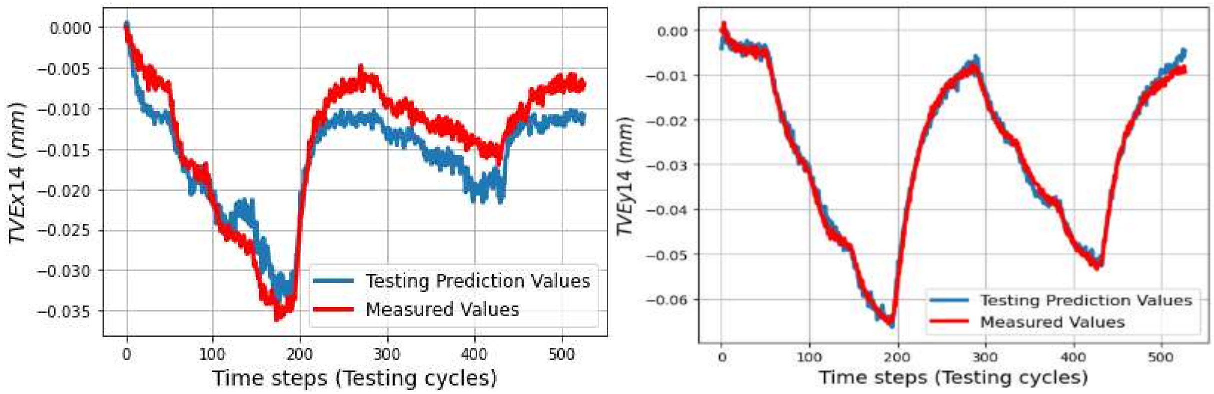

Figure 4.11. Prediction of TVEx14 and TVEy14 in the long process (LP) (1 cycle = 15 min).

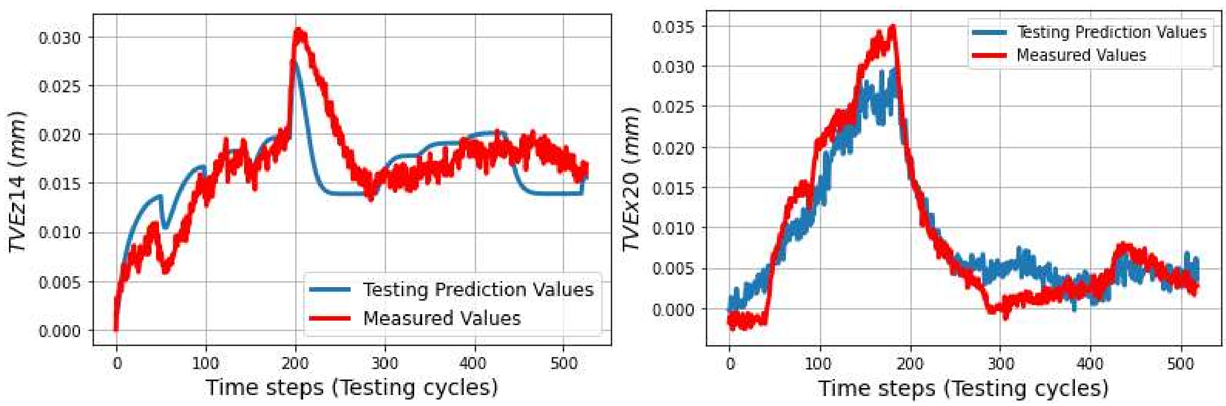

Figure 4.12. Prediction of TVEz14 and TVEx20 in the long process (LP) (1 cycle = 15 min).

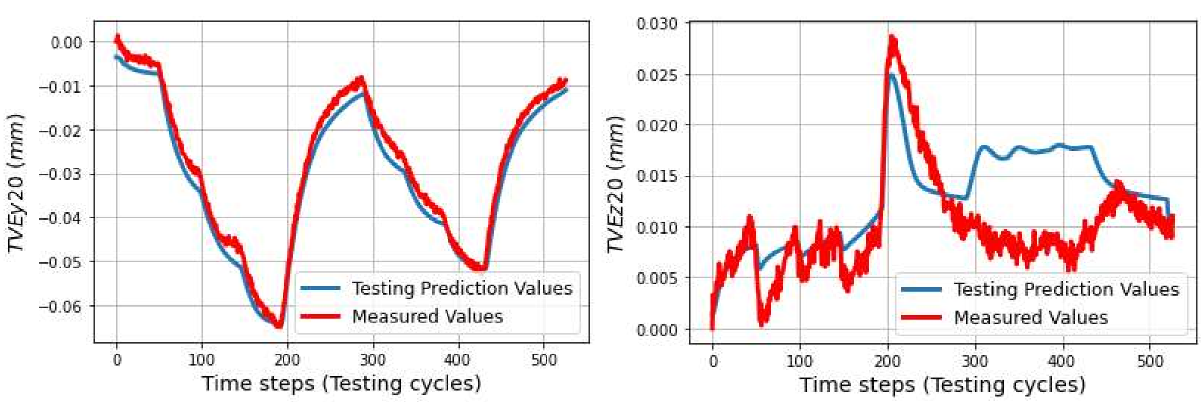

Figure 4.13. Prediction of TVEy20 and TVEz20 in the long process (LP) (1 cycle = 15 min).

The trained model appears to have difficulty predicting TVEz20 from cycles 300 to 450, as its parameters tend to rely heavily on the patterns observed in the training data and do not adjust well to changes in TVEz20 during the testing process. As a result, the model's predictions for TVEz20 during this period are likely to be inaccurate. Conversely, TVEz14 tends to be better predicted during this time. The model is able to adapt more easily to changes in TVEz14.

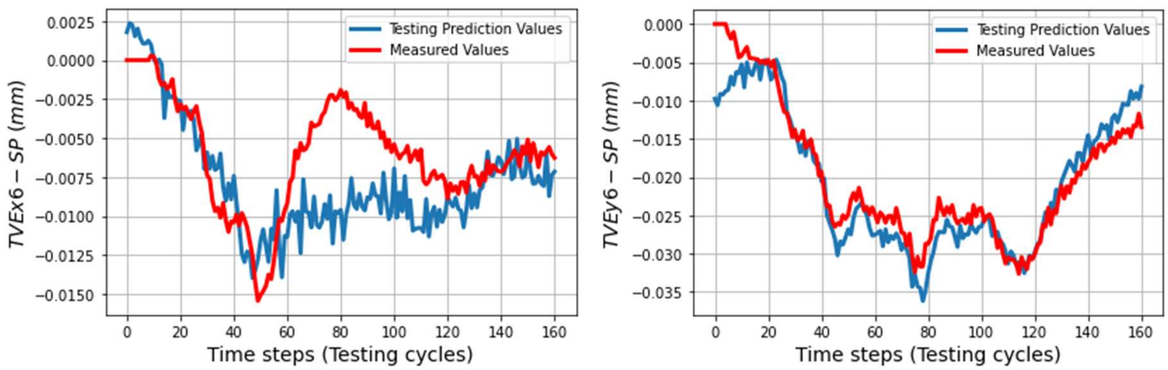

Figure 4.14. Prediction of TVEx6 and TVEy6 in the short process (SP) (1 cycle = 15 min).

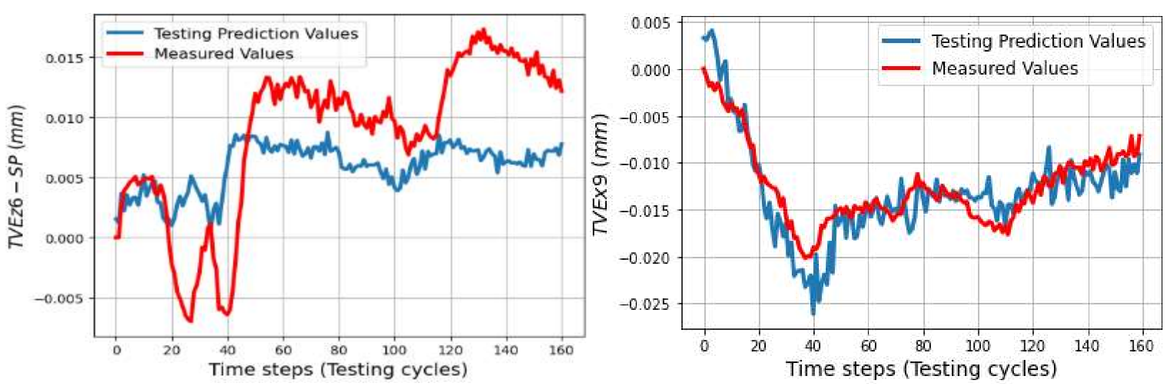

Figure 4.15. Prediction of TVEz6 and TVEx9 in the short process (SP) (1 cycle = 15 min).

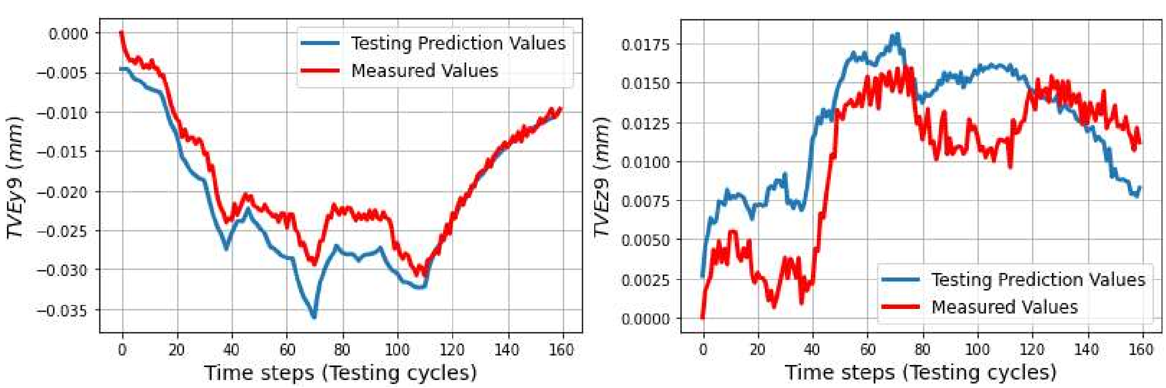

Figure 4.16. Prediction of TVEy9 and TVEz9 in the short process (SP) (1 cycle = 15 min).

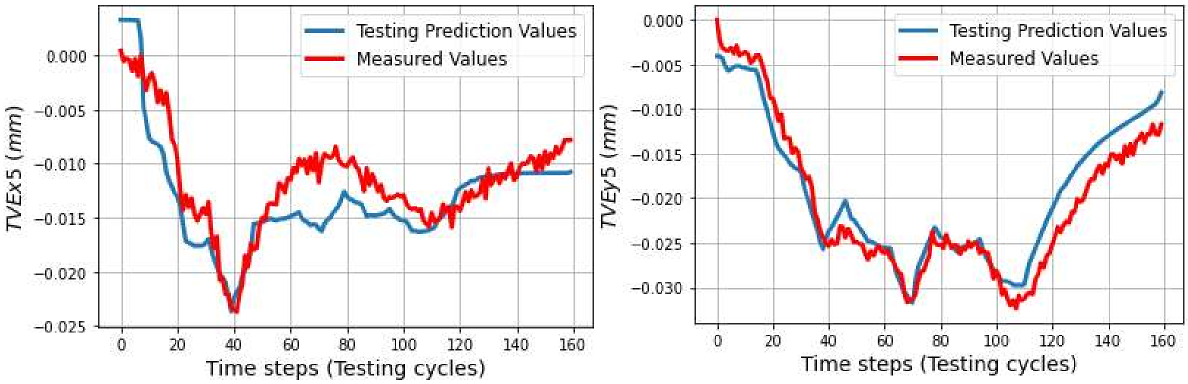

Figure 4.17. Prediction of TVEx5 and TVEy5 in the short process (SP) (1 cycle = 15 min).

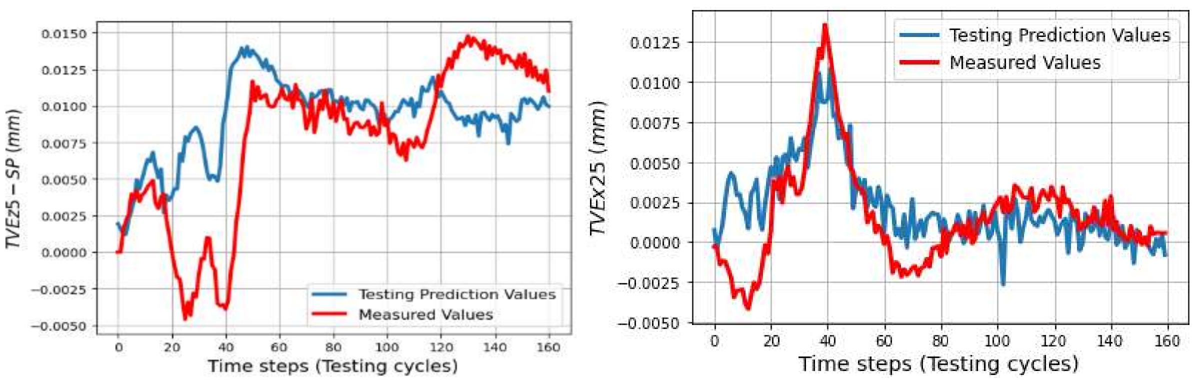

Figure 4.18. Prediction of TVEz5 and TVEx25 in the short process (SP) (1 cycle = 15 min).

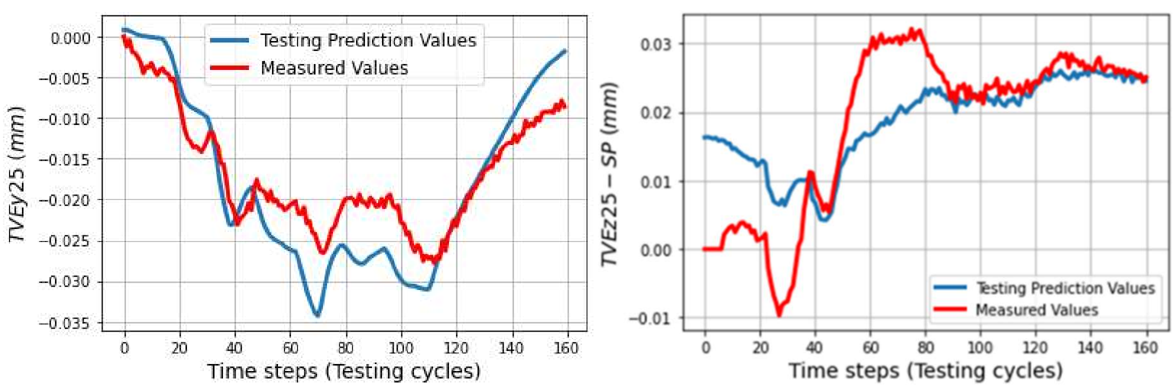

Figure 4.19. Prediction of TVEy25 and TVEz25 in the short process (SP) (1 cycle = 15 min).

The Root Mean Square Errors (RMSEs) and fittings of predictions compared with measured values are calculated by Equations (4.6) and (4.7) are shown from Figure 4.20 to Figure 4.24.

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{j=1}^{n}\left(\text{Prediction values}_j\text{-Measured values}_j\right)^2}{n}} \tag{4.6}$$

$$\text{fitting (\%)}=\left(1-\frac{\mathrm{RMSE}}{\left|\text{Measured values}_{jmax}\text{-Measured values}_{jmin}\right|}\right)*100 \tag{4.7}$$

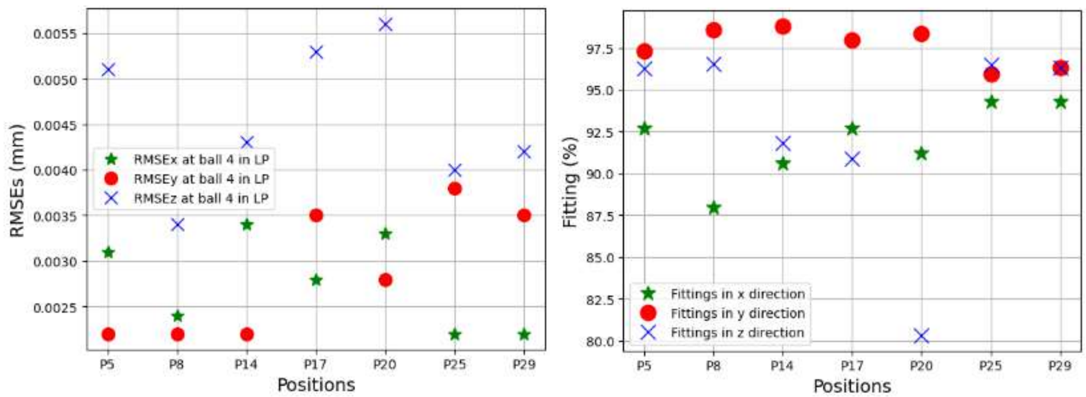

Figure 4.20. RMSEs and fittings of seven positions at ball 4 in the long process (LP).

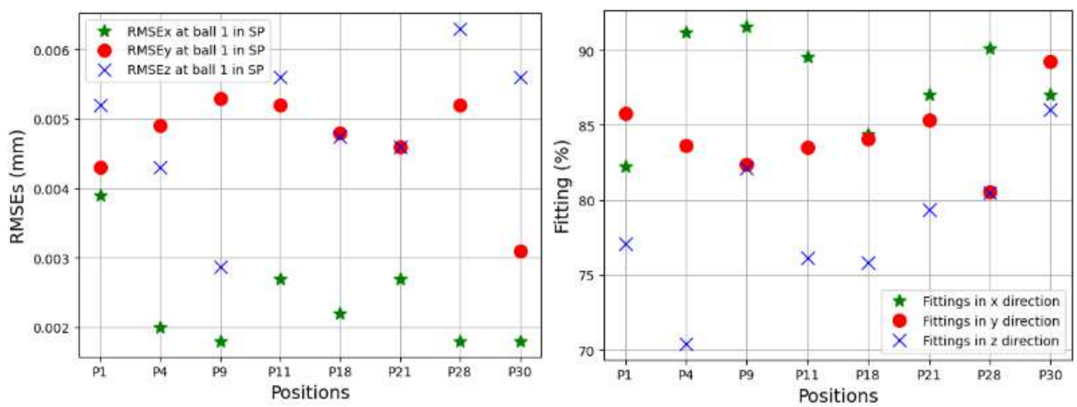

Figure 4.21. RMSEs and fittings of eight positions at ball 1 in the short process (SP).

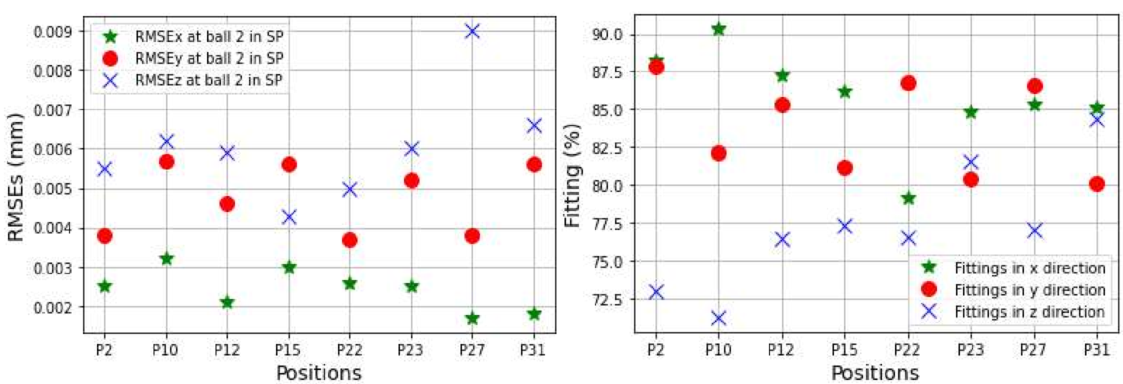

Figure 4.22. RMSEs and fittings of eight positions at ball 2 in the short process (SP).

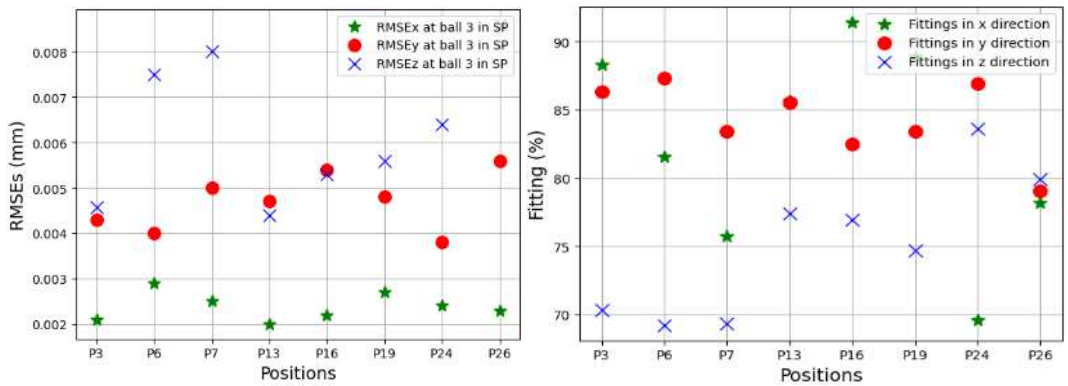

Figure 4.23. RMSEs and fittings of eight positions at ball 3 in the short process (SP).

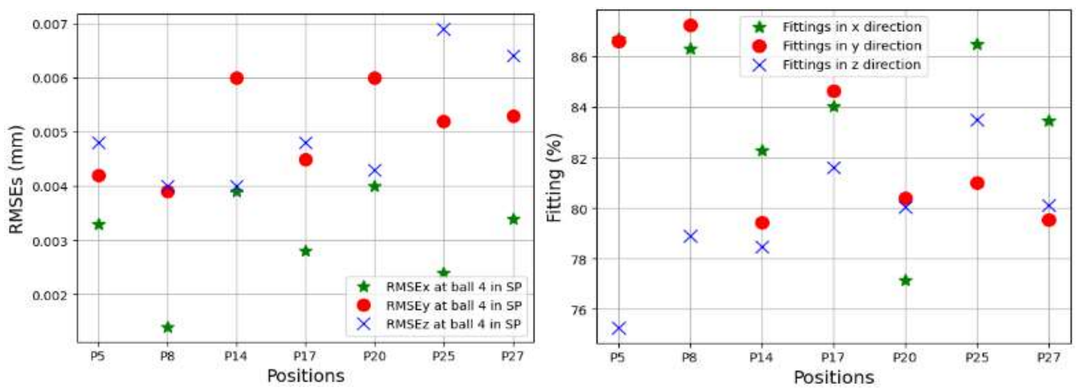

Figure 4.24. RMSEs and fittings of seven positions at ball 4 in the short process (SP).

The changes of TVEs depend on the inputs: powers of rotary axes, durations of activities, positions, and indexations. The trained models have capability to predict TVEs, respectively illustrated from Figure 4.11 to Figure 4.24 and Figure 4.25 presents a set of TVEs in 33 linear axis positions with directional components of TVEx, TVEy, and TVEz at one cycle in the testing process. Predictions of TVEs of seven positions at ball 4 in the long process can reach very high fitting with RMSE = 2.2 µm (fitting = 98.8%) for TVEy14, while the lowest proportion of fitting in this process is for TVEz20 with RMSE = 5.6 µm (fitting = 80.3%).

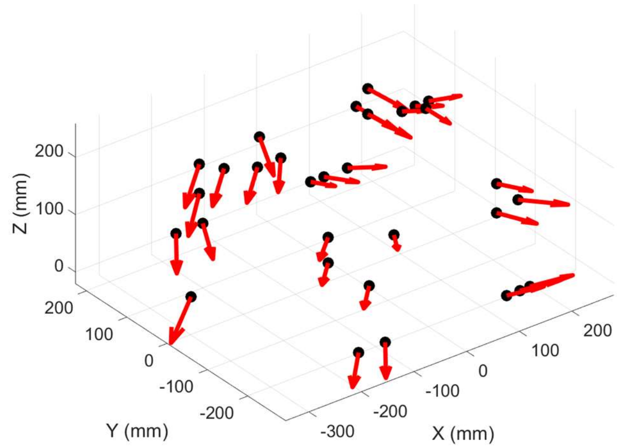

Figure 4.25. 31 linear axis positions and an example set of TVEs.

The trained models gave less quality when they predict TVEs at 7 positions at ball 4 in the short testing process as shown by RMSEs from 3.3 µm (86.7% fitting) for TVEx5 to 4.8 µm (75.27% fitting) for TVEz5. The predictions are good for positions at ball 1, 2, and 3 in the short testing process with the highest fitting 91.6% (1.8 µm) for TVEx9, and the worst case in this process with 69.2% (7.5 µm) for TVEz6. The results also indicate that the model gives the highest capability of prediction for TVEy and the worst for TVEz. These situations happened in [100], with geometric errors relating to y and z components. There are some over responses of the trained model in the beginning of the short process for all TVEs, but the predictions gradually improve to closely track the measured values for the rest of the process. Figure 4.20 to Figure 4.24 reveal an interesting finding: for the positions located at the artifact ball four, TVEy tends to have a higher fitting than TVEx and TVEz. On the other hand, for the positions located at the artifact balls one, two, and three, TVEx tends to dominate the others.

The SLSTMs model showed its ability to predict thermally induced volumetric errors of the machine tool based on the motor powers of the rotary axes. GRU unit was invented after LSTMs with a simpler structure for time series application. The question is: which one is better for data sequences? To answer this question, this work continues by taking some cases including the best and worst predictions of SLSTMs and uses SGRUs to train model and predict these thermally induced volumetric errors with similar structure, learning rate, and weight decay.

SLSTMs predict TVEy14, TVEx4 better than SGRUs as shown in Figure 4.26 to Figure 4.29 with RMSEs = 2.2 µm, 2 µm and 2.4 µm, 4.4 µm, respectively. For TVEz5, 6, SGRUs performs better with RMSEs = 3.2 µm and 6.6 µm, while SLSTMs have RMSEs turnoff 4.8 µm and 7.5 µm. Both the SLSTMs and SGRUs proved that they have potential to predict the sequence data. The results in this section partly prove the ability of SLSTMs and SGRUs to predict TVEs at the 40 h process with different activities and inputs.

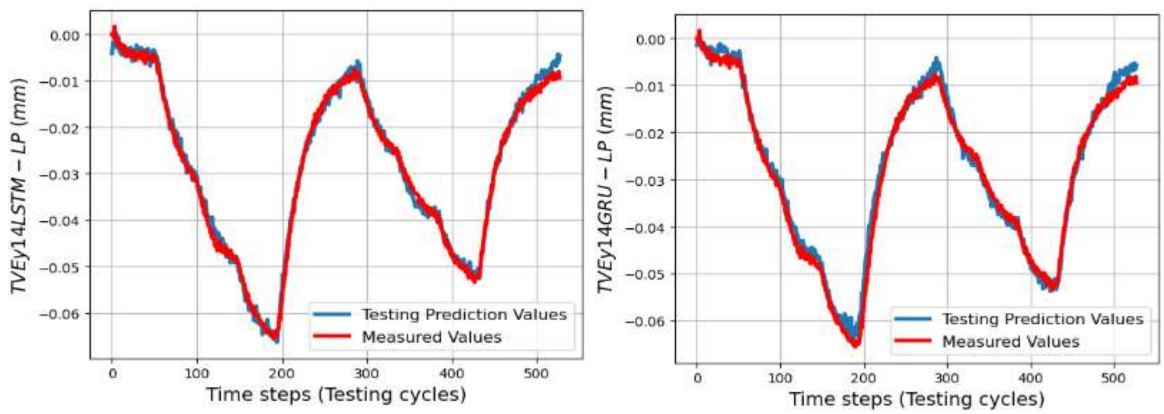

Figure 4.26. Prediction of TVEy14 by SLSTMs and SGRUs in the long process (LP) (1 cycle = 15 min).

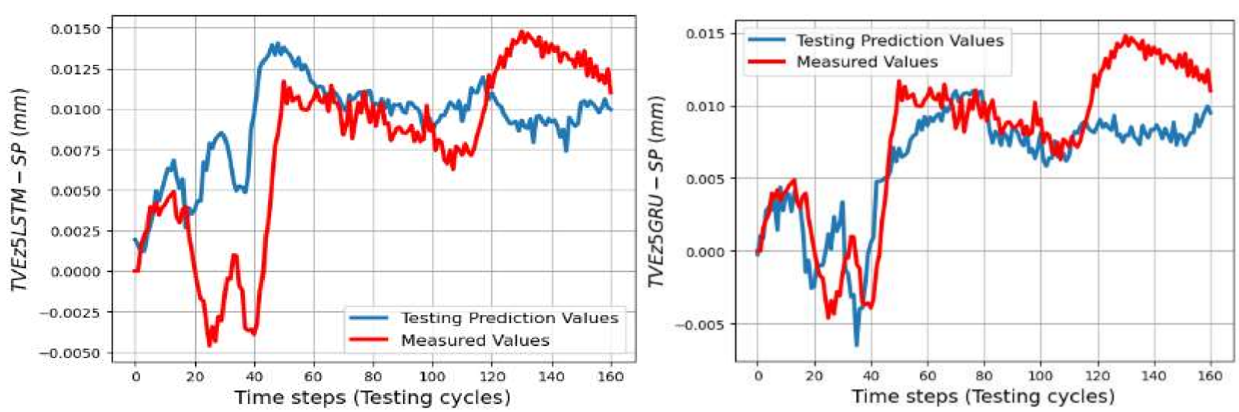

Figure 4.27. Prediction of TVEz5 by SLSTMs and SGRUs in the short process (SP) (1 cycle = 15 min).

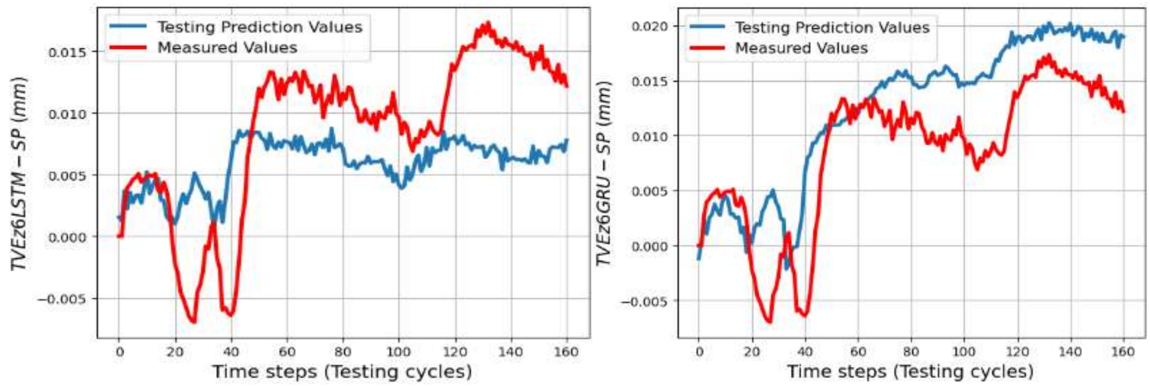

Figure 4.28. Prediction of TVEz6 by SLSTMs and SGRUs in the short process (SP) (1 cycle = 15 min).

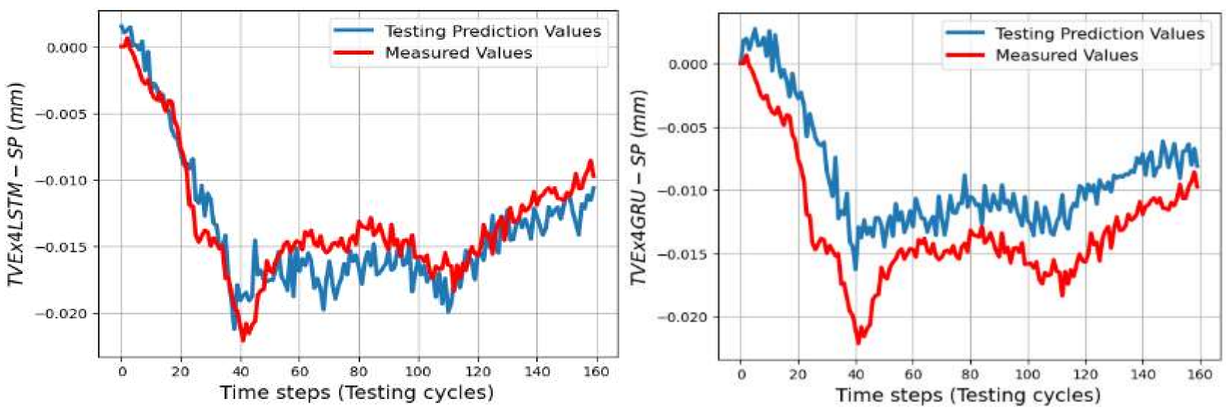

Figure 4.29. Prediction of TVEx4 by SLSTMs and SGRUs in the short process (SP) (1 cycle = 15 min).

This study has several limitations that should be taken into account. Firstly, the heat sources are limited to the rotary axes, which may affect the quality of predictions of TVEz. To improve the accuracy of the predictions, it may be beneficial to include more information on the activities of the linear axes and temperatures as inputs. Additionally, the ambient temperature should not be overlooked as it can directly affect the change in temperature of the axes following a sequence of activities.

4.6 Conclusions

In this study, an approach using SLSTM/SGRU for directly predicting thermally induced volumetric errors without the use of a kinematic model with geometric error parameters was proposed. SLSTMs\SGRUs+AdamW are used for modelling and predicting the sequential data. The training and testing data sets used different positions and different B- and C-axis exercise cycles. The SLSTMs model's best prediction in the long process reached RMSE of 2.2 μm (98.8% fitting to the experimental measurement), while in a short testing process, it achieved 1.8 μm (91.6%). The worst case predicted by the model in pro-cesses is 7.5 μm (69.2% fitting). This work gives a potential solution in practice to predict the thermal volumetric errors at different positions in the workspace directly over an ex-tended period of 40 h without the need for remeasuring the thermal errors, which allows sufficient time for the machining of most workpieces.

A comparison between two popular deep learning models for sequential data, SLSTMs and SGRUs, is carried out for some outstanding TVEs. The purpose is to determine which one is better for this application. Their capabilities were similar.

Future work can focus on exploring the heat sources associated with the activities of five-axis machine tools, as well as examining the robustness of deep learning models when faced with variations in ambient temperature.

CHAPTER 5 ARTICLE 3: APPLYING DEEP LEARNING AND MONTE CARLO DROPOUT FOR MODEL'S CONFIDENCE OF THERMALLY INDUCED KINEMATIC ERRORS OF FIVE AXIS MACHINE TOOL

Huy Vu Ngoc*, J.R.R. Mayer, Elie Bitar-Nehme

Mechanical Engineering Department, École Polytechnique de Montréal, P.O. Box 6079, Station Downtown, Montréal, QC H3C 3A7, Canada

Submitted to CIRP Journal of Manufacturing Science and Technology, 03/06/2023.

Abstract

Applying Deep learning LSTM (Long Short Term Memory) for thermally induced error problems in machine tool has achieved positive outcomes. However, many studies fail to mention the model uncertainty, which refers to output distribution or confidence bounds caused by variations in model parameters. In manufacturing, the model uncertainty is relevant because it provides operators with greater confidence to decide if predicted thermal errors are reliable without needing to re-measure the outputs over a long period. This study involves building a deep learning model capable of predicting thermally induced kinematic errors (TKEs) of a five-axis machine tool in multistep time series while offering confident upper and lower bounds. Dropout technique is employed to prevent network overfitting as it causes output variation. Hence, Monte Carlo dropout (MCD) is used to establish the prediction distribution during the testing process. Separate training and testing experiments are conducted to investigate the thermal effects of a five axis machine tool activities over 264 hours and 84 hours, respectively. The confidence of predictions ranges from 50% to 99%.

Keywords:

Bayesian SLSTMs, Monte Carlo dropout (MCD), model uncertainty, confident bounds, machine tool, thermal errors.

5.1 Introduction

The distribution of a model's outputs caused by input variance are investigated in [108, 109], but less attention has been paid to another potential effect from model's structure. Although deep learning is becoming increasingly popular for modelling thermal behaviors of machine tool, the stochasticity of the model's structure and parameters can lead to unstable outputs. One question that arises is: how robust is a deep learning model? Ngoc et al. modelled thermal behavior of machine tool, but did not account for the distribution of predictions [100, 110]. [111] demonstrated that the model's output distribution can be determined by calculating a probability distribution over the model parameters instead of having the deterministic weights and single point optimization. Bayesian method can be applied to do this task.

[112] used Bayesian network to predict thermal errors in a machine tool under different conditions using Support Vector Machine (SVM) model. The SVM builds relationships between temperature rise of the X-axis ball screw nut and thermally induced X-axis positioning error of a three axis vertical machining center, accounting for variations in axis rates, spindle speeds and loads on the X-axis table. In [113] Papananias et al. proposed Bayesian regularized artificial neural networks for modelling and predicting the distribution of points' actual coordinates with inputs: nominal coordinates, ambient temperature and temperature of the work-piece. [114] modelled the correlation between the temperature field (the set of temperature values at all points in a given space at a given instant) rise and position error of the feed drive system across various operating conditions by Bayesian Neural Network (BNN). Similarly, [20] applied BNN to model the thermal errors of the XY linear motor feed system. [115] generated a probability of stability map with Bayesian machine learning in the spindle speed-axial depth of cut domain. These works used the basic structure of neural network.

LSTM is known for processing time series data. Weights and biases are optimized based on global minimum of loss function or maximizing likelihood function. Dropout technique is popular to prevent the overfitting of the LSTM model [52], but this leads to random outputs at each stage or the output distribution of the model (posterior distribution). Like neural networks, LSTM networks are deterministic functions that train the models by pointwise optimization for sequential data. It does not have the ability to qualify the posterior distribution, which may not be reliable in decision making [50, 54, 116, 117].

In [50] it is shown that it is difficult to estimate Bayesian variational inference in deep learning like convolutional neural networks and SLSTM (Stacked LSTM) due to the high dimensionality of the parameter space. [50, 61] opened new way to use Monte Carlo dropout (MCD) to approximate variational inference. Firstly they modified the distribution of the weights by applying a prior distribution over them, which is done by using Bernoulli distribution through the dropout technique. Repeating the model's training process in number of epochs could get multiple possible outputs, which were then averaged using MC. This is similar to a Bayesian approximation. Fujishima et al. [118] applied Monte Carlo dropout to approximate Bayesian inference for Convolutional neural network. This work investigated the prediction capability in severe situation with unexpected temperature change and sensor failure by generating the prediction range based on the mean and variances of prediction values. [100, 110] proved SLSTM is a potential solution to predict thermal errors in machine tool in long term without re-measurement of the outputs. This work quantifies the posterior distribution of deep learning with SLSTM model with data from activities of the two rotary and three linear axes of a five-axis machine tool. This study is presented in six sections. Section 5.1 reviews previous works. Section 5.2 describes the experimental process. Section 5.3 shows the thermal kinematic errors. Section 5.4 presents MCD for SLSTMs. The results and discussion are in Section 5.5 and the conclusion follows in Section 5.6.

5.2 Experimental process

Figure 5.1 shows the experimental setup on a five axis machine tool model HU40T made by Mitsui Seiki. Two datasets, one for the training and the other for the testing process of the thermal effects are generated from activities of all five axes. The spindle is not active as it is not considered in this study.

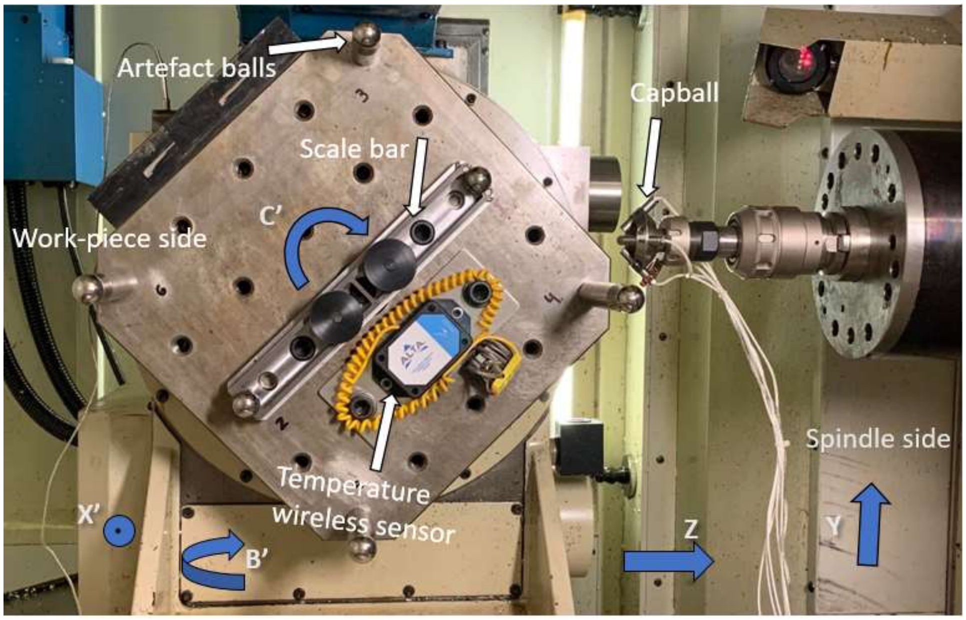

Figure 5.1. Experimental setup with four artefact balls, the scale bar, a wireless temperature sensor on the work-piece side and the Capball on the spindle side

Four stem mounted artefact balls are fixed to the work table on the work-piece side. The Capball measuring head with five capacitive sensors is installed in a tool holder and mounted on the spindle [105] to provide measurements of the ball positions that are analyzed using the SAMBA method [107], which yields estimates of the kinematic errors of the five-axis CNC machine tool. Two experiments were conducted at different axes' speeds and combination of activities of the five axes. The first experiment lasted 264 hours including 24 hours of machine warming, 24 hours of heating through exercising for each axis, and 24 hours of cooling through resting between the heating of each axis. The second experiment lasted 84 hours, with 24 hours of warming, 36 hours of the simultaneous movements of all five axes at two speed levels, and 24 hours of cooling. Every 15 minutes, in the first and second experiments, a measuring cycle lasting almost 4 minutes is conducted using the Capball at 45 and 46 linear axis positions, respectively. These measured positions are at four artefact balls, which are calculated based on the indexations of the B and C axes as well as the positions of the artefact balls on the work table, as shown in Figure 5.2. B- and C-axis indexations are illustrated in Figure 5.3.

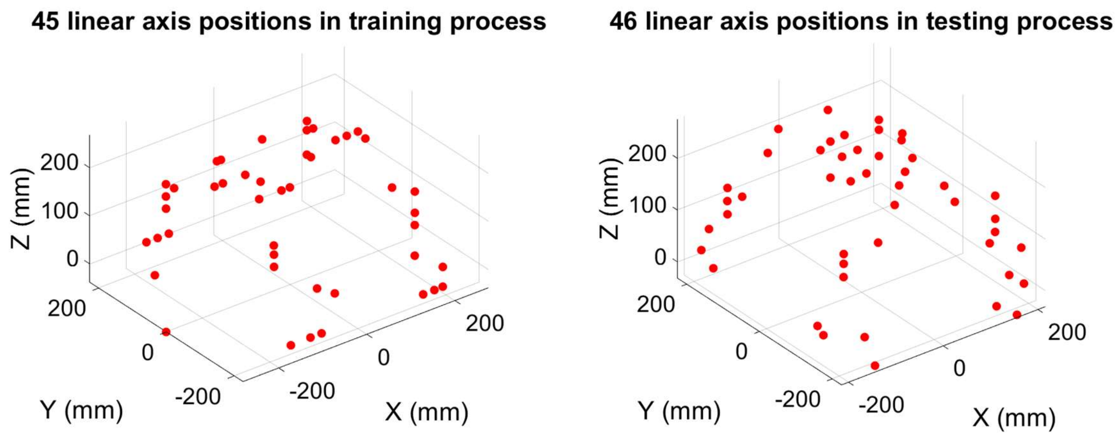

Figure 5.2. Linear axis positions in the training and testing processes

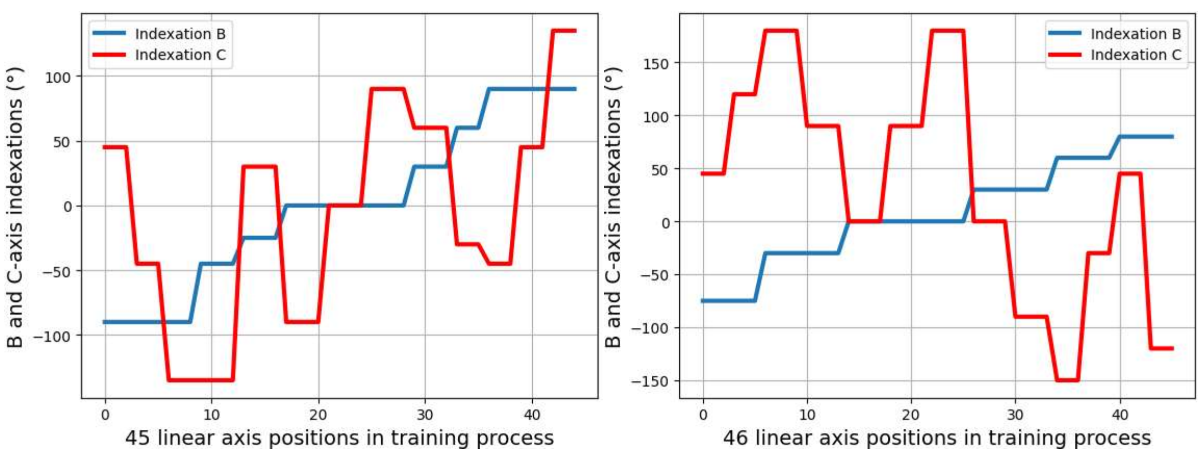

Figure 5.3. B- and C-axis indexations in the training (left) and testing (right) processes

The positions of the four artifact balls and the indexations of the B and C axes in the first experiment are different from those in the second experiment. As a result, the measured positions during the testing process (the second experiment) are dissimilar to those in the training process. Figure 5.4 show the activity sequences with the supply powers of axes X, Y, Z, B, and C.

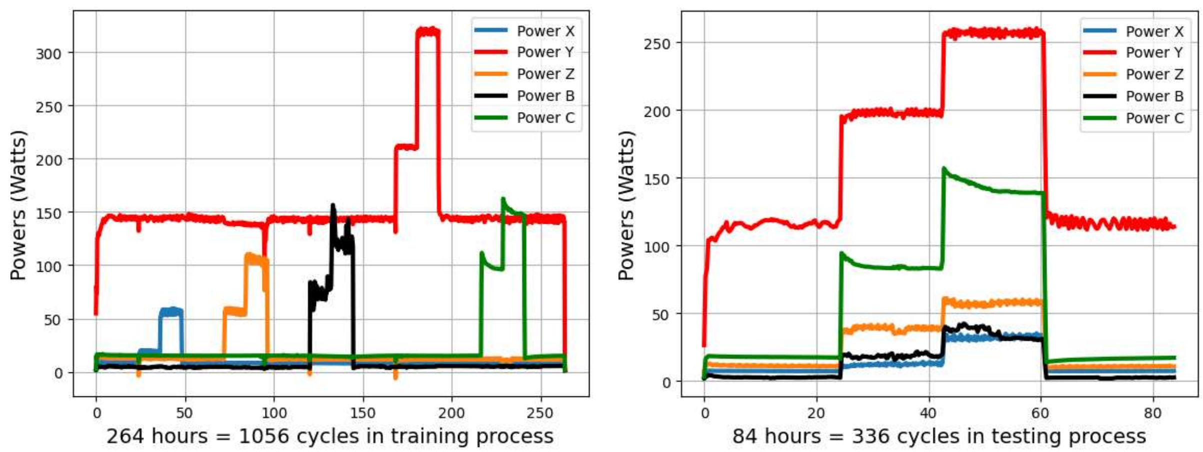

Figure 5.4. Training (left) and testing (right) processes

Only seven temperature sensors are used as these are fairly intrusive. In addition, supply powers are also measured and is the preferred data source in this work. The temperature sensors are K-type wired and wireless ones, which are mounted on the bearings of axes X, Y and B, Z axis's ball nut and the table (work-piece), near the C-axis. One thermocouple measures the ambient temperature inside the machine tool. These temperatures and the five axes' supply powers are the inputs of the model for the training and testing processes.

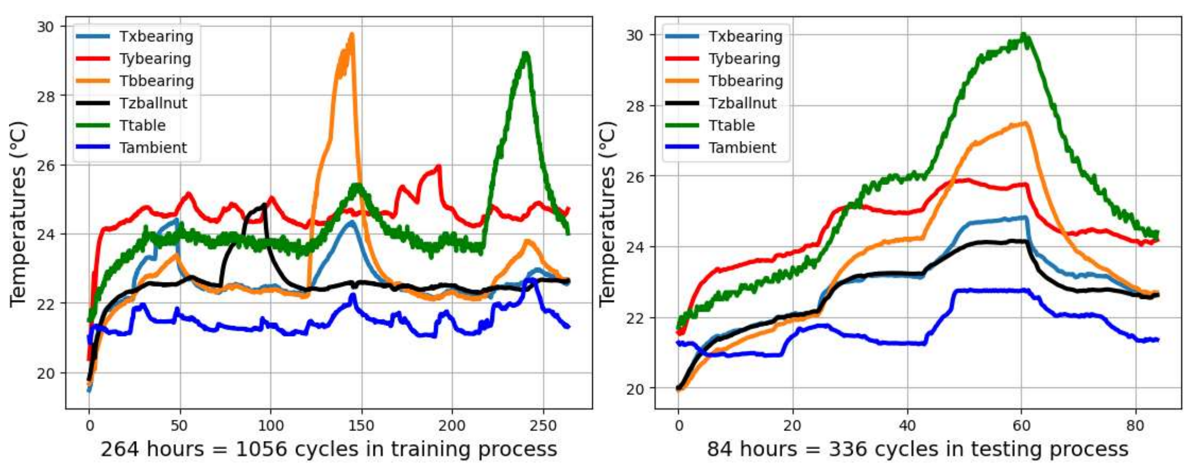

Figure 5.5: Temperatures in the training (left) and testing (right) processes

Figure 5.5 shows the temperatures at different times for different temperature sensors. The initial conditions to start both training and testing processes are close with ambient temperatures of 21.1 °C and 21.3 °C, respectively.

5.3 Thermal kinematic errors

To get the machine tool's kinematic errors data for the deep learning model, this work uses SAMBA method [107] to estimate kinematic error parameters. The thermally induced kinematic errors, TKEs, are the subtractions of their initial values, KE_0, from all subsequent values, KE.

$$TKE = KE - KE_0 \tag{5.1}$$

A total of 13 kinematic errors are estimated as shown in Figure 5.6 and their descriptions are presented in Table 5.1

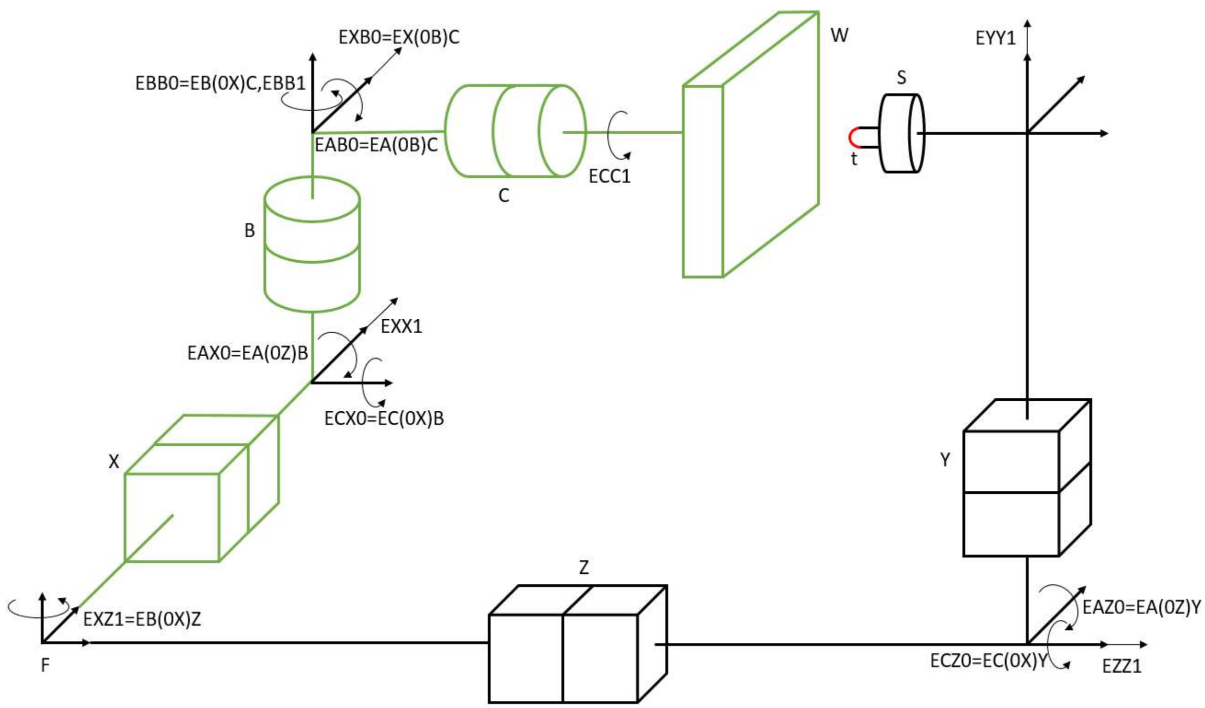

Figure 5.6: Machine schematic with thermally induced kinematic errors

Table 5.1. 13 thermally induced kinematic error parameters

Kinematic error parameters	Description
EXX1	1^{st} degree term of linear positioning error motion in X-axis (mm/mm)
EYY1	1^{st} degree term of linear positioning error motion in Y-axis (mm/mm)
EZZ1	1^{st} degree term of linear positioning error motion in Z-axis (mm/mm)

EBB0=EB(0X)C	Out of squareness angle of C-axis relative to X-axis (radian)
EBB1	1st degree term of angular positioning error motion of B-axis (radian/radian)
ECC1	1st degree term of angular positioning error motion of C-axis (radian/radian)
EXB0=EX(0B)C	Distance error between C and B-axes (mm)
EAX0=EA(0Z)B	Out of squareness angle of B-axis relative to Z-axis (radian)
ECX0=EC(0X)B	Out of squareness angle of B-axis relative to X-axis (radian)
EAB0=EA(0B)C	Out of squareness angle of C-axis relative to B-axis (radian)
ECZ0=EC(0X)Y	Out of squareness angle of Y-axis relative to X-axis (radian)
EXZ1=EB(0X)Z	1st degree term of straightness error motion of Z-axis in X-axis = Out of squareness angle of Z-axis relative to X-axis (mm/mm or radian)
EAZ0=EA(0Z)Y	Out of squareness angle of Y-axis relative to Z-axis (radian)

5.4 MCD for SLSTMs

LSTMs are popular for processing time series data. The differences between LSTMs and Recurrent neural networks (RNNs) are the gates (forget gate (f_t), input gate (i_t) and output gate (o_t)), the activation functions sigma σ and tanh, and cell state, which can help prevent the vanishing gradient problem commonly encountered in the RNNs.

The output of each LSTM unit is calculated as follows:

$$H_t=\sigma(f_t*c_{t-1}+i_t*\sigma(W_c x_t+U_c H_{t-1}+b_c)) \tag{5.2}$$

$$Y_t=W_h H_t \tag{5.3}$$

where H_t and Y_t are hidden state and output at state t; W_c, U_c and b_c are weights and bias of the model.

Dropout [52] is a technique that is used in deep neural networks with a large number of parameters to deal with overfitting problem. It regularizes the networks by randomly deactivating neurons. In this work, dropout is applied for SLSTM as shown in Figure 5.7. The dropout rate = 0.5 is chosen based on [119].

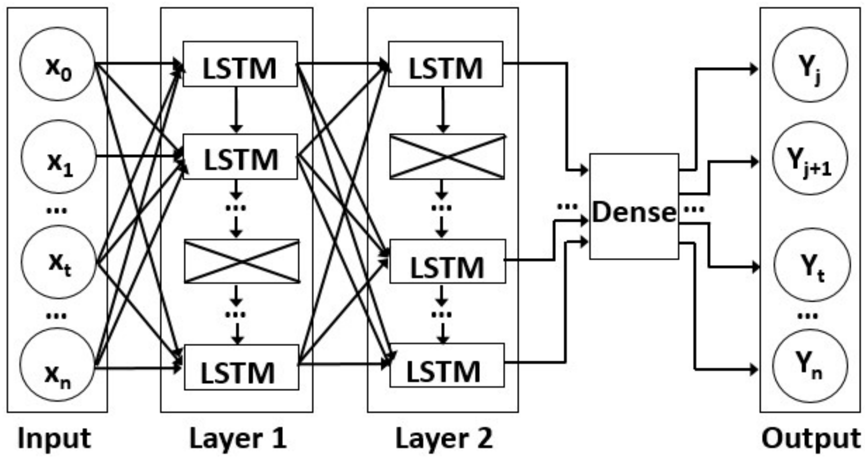

Figure 5.7. SLSTM with dropout

The SLSTMs have two hidden layers, the widths of these two layers are 48 and 8, respectively. Dropout is illustrated in Figure 5.7 by rectangles with crosses. The input variables x_t are the ambient temperature, supply powers and temperatures at predetermined positions, which are selected depending on their impacts on the thermal kinematic errors. The outputs Y_t are the predicted thermally induced kinematic errors. The first value is Y_j with j=size of input sample for deep learning model. This work uses j = 8.

Bayes' theorem offers a mathematically grounded technique for quantifying model uncertainty [48, 49]. The model uncertainty is estimated by finding posterior distribution $p(\theta|X,Y)$:

$$p(\theta|X,Y)=\frac{p(Y|X,\theta)p(\theta)}{p(Y|X)} \tag{5.4}$$

where $p(Y|X,\theta)$ is the likelihood, $p(Y|X)$ is the evidence, and $p(\theta)$ is the prior distribution. X, Y are the inputs and outputs of the model in the training process and θ = [W,U,b] are the model parameters.

Gal [50] showed that Bayes' theorem can be incorporated into available neural network without modifying the models or the optimization to determine the distribution of output. The posterior distribution $p(\theta|X,Y)$ is the main part in the predictive output distribution $p\left(Y^*|X,Y,X^*\right)$ with new inputs X^*. However, estimating the posterior distribution is still a challenge for multilayer networks. Gal proposed a mathematically practical solution that uses MCD to address this problem in deep learning networks [50, 61].

MCD is simply equivalent to performing T stochastic forward passes through the network with different dropout configurations, which create a distribution of predictive values. These values are recorded and averaged at each stage to find the mean and divergence of prediction that is equal to finding model uncertainty [50], [61] as shown in Figure 5.8.

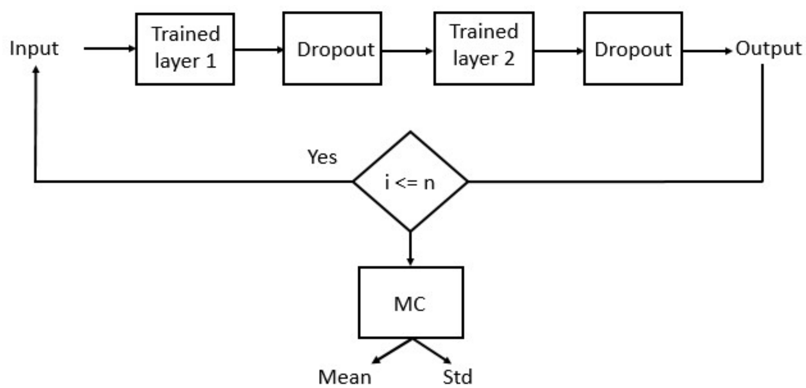

Figure 5.8. MCD diagram

MC is applied to calculate means and deviations of the predictions:

$$\bar{\hat{y}}^* \approx \frac{1}{T}\sum_{t=1}^{T} \hat{y}^*(x^*, W_1^t, W_2^t, U_1^t, U_2^t, b_1, b_2) \tag{5.5}$$

Standard deviation (std) presents normal distribution around a central region of the mean of predictions, which illustrate the confidence range of the potential network outputs. Model's predictive std is

$$\mathrm{std}_{q(y^*|x^*)}(y^*) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} (\hat{y}^* - \bar{\hat{y}}^*)^2} \tag{5.6}$$

5.5 Results and discussion

Five thermal kinematic errors EBB0, EBB1, ECC1, EC(0X)B, EC(0X)Y are excluded in this section because they exhibit a weak response to the activities of axes as shown in Figure 5.9, Figure 5.10 and Figure 5.11.

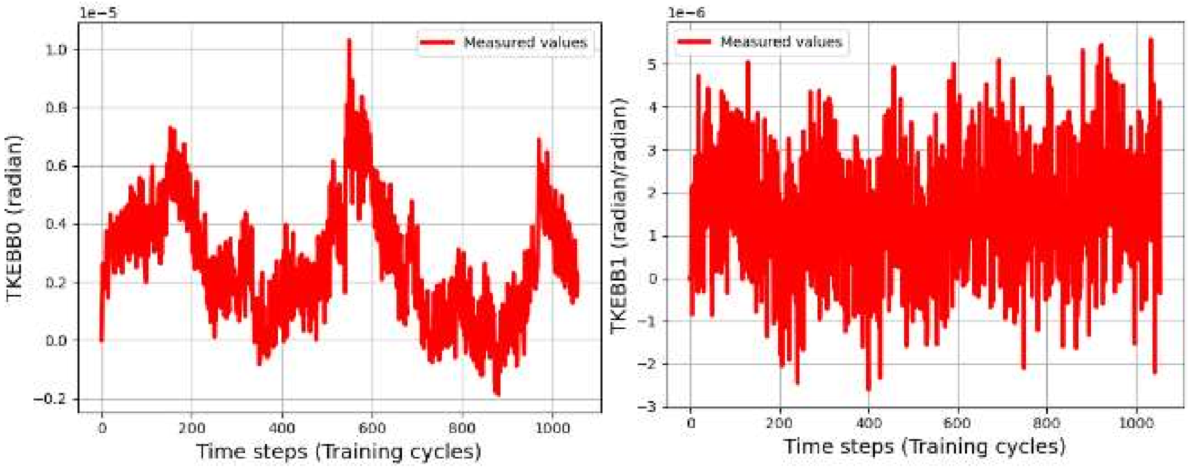

Figure 5.9. Measured values of TKEBB0 and TKEBB1 in the training process

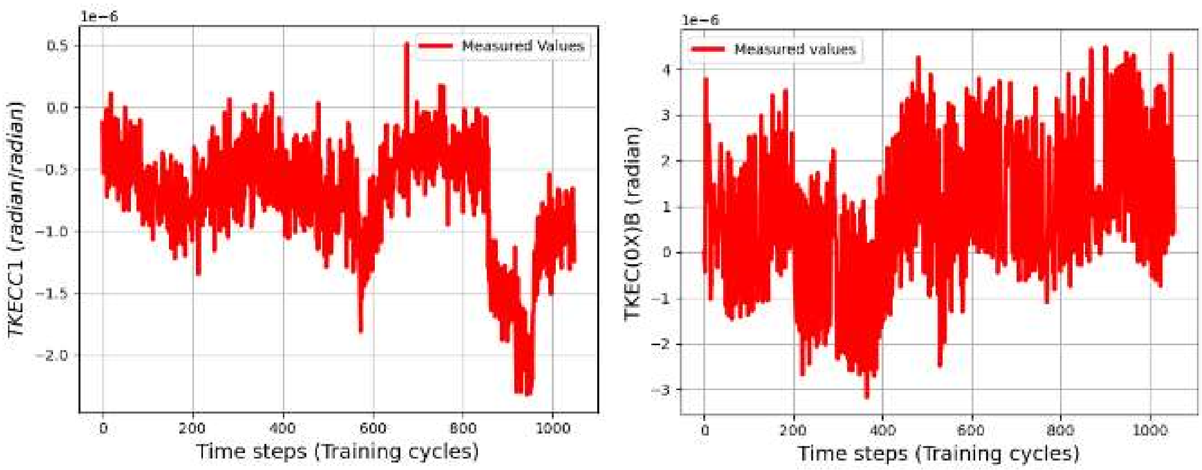

Figure 5.10. Measured values of TKECC1 and TKEC(0X)B in the training process

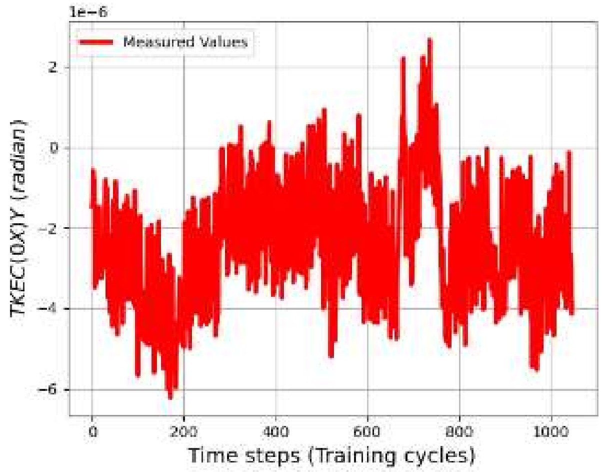

Figure 5.11. Measured values of TKEC(0X)Y

Their thermal variations are smaller than 12×10^{-6} (radian or radian/radian) in the training process, which corresponds to around 3 microns of volumetric error. The geometric symmetry of the machine tool with respect to the YZ plane, coupled to the symmetrical bidirectional heating activity of the B- and C-axis, might have limited the thermal variations of the three errors TKEBB0, TKEC(0X)B and TKEC(0X)Y. The effect of C-axis activity on TKECC1 can be seen on Figure 5.10 (left), but it is negligible. The other eight thermal kinematic errors will be investigated to illustrate the objectives of this work.

The training process is carried out in pre-set epochs = 150 (the number of passes of total training dataset through the network) using Adam optimizer with the learning rate = 0.001. Typically, datasets are divided into batches and after each batch, the model parameters are updated. Before presenting the distribution of predictions, several factors are taken into account, such as selection of inputs for model and the impact of ambient temperature to accuracy of trained model's predictions. In order to support the objectives outlined in the subsequent parts, some thermal kinematic errors are presented by plots, while the remaining ones are summarized in the table under their corresponding RMSEs.

Inputs for training model

The objective of this part is to select the most suitable inputs, which is important in building the model [120].

Figure 5.12 and Figure 5.13 show the predictions of TKEYY1 and TKEA(0B)C with different inputs from one side (work-piece (W) or spindle side (S)) or both sides (WS).

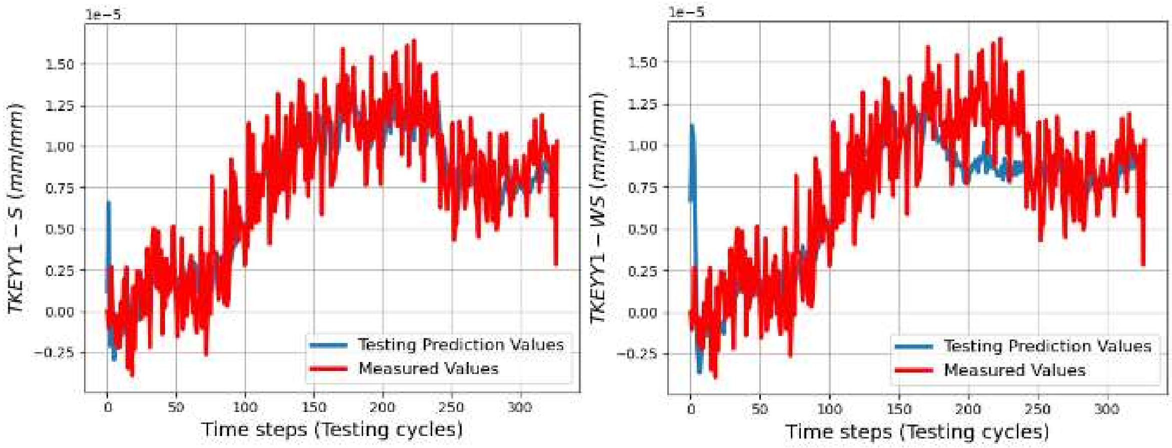

Figure 5.12. Predictions of TKEYY1-S with inputs from spindle (S) side (left) and TKEYY1-WS with inputs from both spindle and work-piece sides (right)

TKEYY1-S presents the predictions of TKEYY1 using the supply powers and temperatures on the spindle side as inputs. The predictions of TKEYY1-WS uses the supply powers and temperatures on both sides (the spindle and work-piece sides) as the inputs. The RMSEs are 2.25×10^{-6} and 2.72×10^{-6}, respectively.

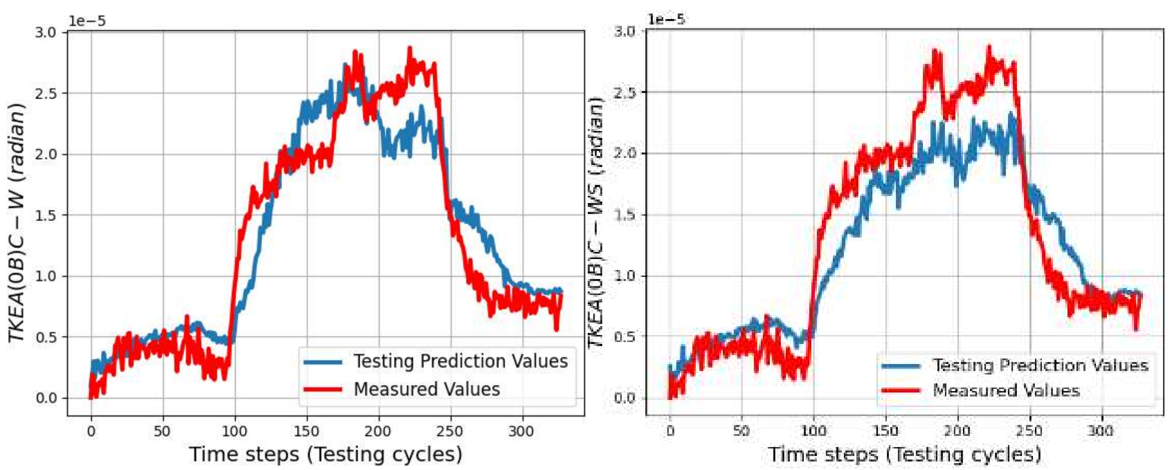

Figure 5.13. Predictions of TKEA(0B)C-W with inputs from work-piece (W) side (left) and TKEA(0B)C-WS with inputs from spindle and work-piece sides (right)

The RMSEs for TKEA(0B)C-W and TKEA(0B)C-WS are 3.02×10^{-6} and 3.47×10^{-6}, respectively, with an advantage for TKEA(0B)C-W. It seems advantageous both for TKEYY1 and TKEA(0B)C to use inputs only from the corresponding machine side with the ambient temperature. We continue considering the inputs of the model from axes whose activities affect the errors in the training process.

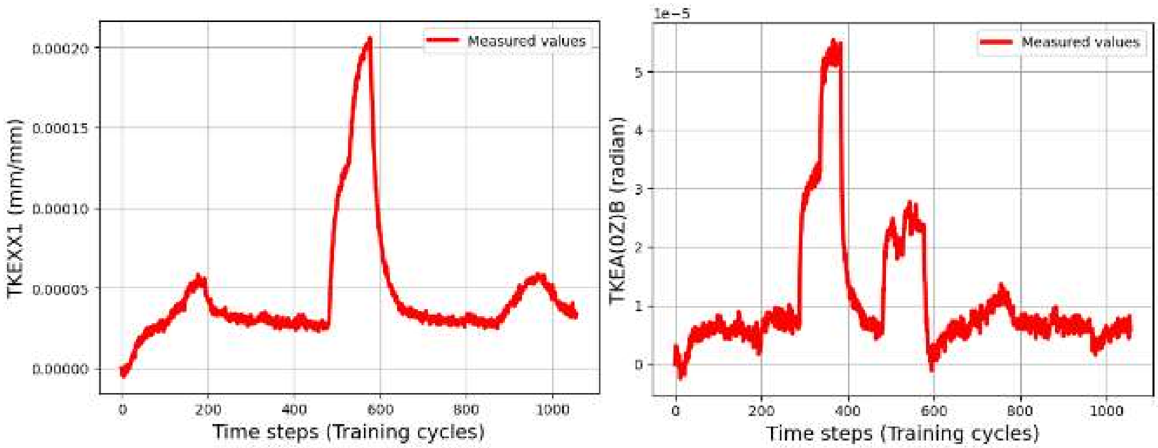

Figure 5.14. Measured values of TKEXX1 and TKEA(0Z)B in the training process

Figure 5.14 (left) shows that the TKEXX1 is affected by the activities of axes X, B and C. TKEA(0Z)B on the right of Figure 5.14 varied with the activities of the B, Y and Z-axis. The inputs used for training model and predicting TKEA(0Z)B are the temperatures, the supply powers of B, Y, and Z-axis and the ambient temperature.

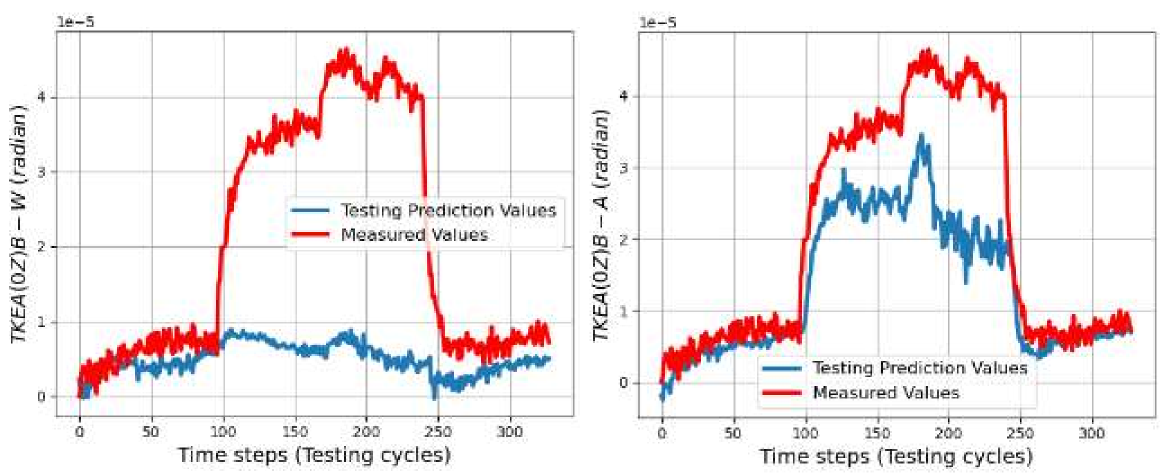

Figure 5.15. Predictions of TKEA(0Z)B-W with inputs from work-piece side (left) and TKEA(0Z)B-A affected (A) by inputs from B, Y and Z-axis (right)

Figure 5.15 (left) shows the predictions of TKEA(0Z)B-W with the inputs (supply powers and temperatures) from the work-piece side and TKEA(0Z)B-A in Figure 5.15 (right) is predicted from the inputs of the axes B, Y and Z. The ambient temperature is also used as an input in above cases. The predictions are significantly different with RMSEs = 20.83×10^{-6} and 10.43×10^{-6}, respectively. It can be explained by considering Figure 5.4, Figure 5.5 and Figure 5.14 (right). The trained model gives more importance to the inputs from the B, Y and Z-axis than from the X and C-axis. For more detail, the activity of the B, Z and Y-axis (from hour 72 (cycle 288) to hour 192 (cycle 768)) in the training process has a bigger effect on TKEA(0Z)B than the activity of X (from cycle (96 to 192) and C axis (from cycle 864 to 960) as shown in Figure 5.14 (right). The others will be listed in Table 5.2 and Table 5.3 by considering the same reason.

Table 5.2. TKEs and axes' activities affecting these TKEs

TKEs	Axes' activities affecting TKEs in training process
TKEXX1 (mm/mm)	X, B, C
TKEYY1 (mm/mm)	Y, Z
TKEZZ1 (mm/mm)	Y, Z
TKEA(0B)C (radian)	B, C
TKEX(0B)C (mm)	X, B, C
TKEA(0Z)B (radian)	B, Z, Y
TKEB(0X)Z (radian)	B, X, Z
TKEA(0Z)Y (radian)	Y, Z

Table 5.3. RMSEs of the TKEs in cases of using different supply powers and temperatures as inputs

Thermal kinematic errors	Inputs from one side (work-piece or spindle side)	Inputs from both sides (work-piece and spindle sides)	Inputs from axes affecting thermal kinematic errors
TKEXX1 (mm/mm)	7.09×10^{-6}	9.20×10^{-6}	7.09×10^{-6}
TKEYY1 (mm/mm)	2.25×10^{-6}	2.72×10^{-6}	2.25×10^{-6}
TKEZZ1 (mm/mm)	3.90×10^{-6}	8.78×10^{-6}	3.90×10^{-6}
TKEA(0B)C (radian)	3.02×10^{-6}	3.47×10^{-6}	2.79×10^{-6}
TKEX(0B)C (mm)	0.79×10^{-3}	0.85×10^{-3}	0.79×10^{-3}
TKEA(0Z)B (radian)	20.83×10^{-6}	13.10×10^{-6}	10.43×10^{-6}
TKEB(0X)Z (radian)	6.68×10^{-6}	6.55×10^{-6}	6.03×10^{-6}
TKEA(0Z)Y (radian)	11.24×10^{-6}	17.16×10^{-6}	9.69×10^{-6}

The trained model's predictions of thermal kinematic errors are more accurate when inputs are from one side rather than both sides such as TKEA(0B)C, TKEX(0B)C, TKEA(0Z)Y, TKEXX1, TKEYY1, TKEZZ1. However, for the others like TKEA(0Z)B and TKEB(0X)Z, better predictions are obtained when the inputs are provided from both sides. Interestingly, the best predictions are achieved when the trained model uses the inputs from the axes that affect the thermal kinematic errors as shown in Table 5.2. To proceed, we need to determine whether we should use only the temperatures, powers or both.

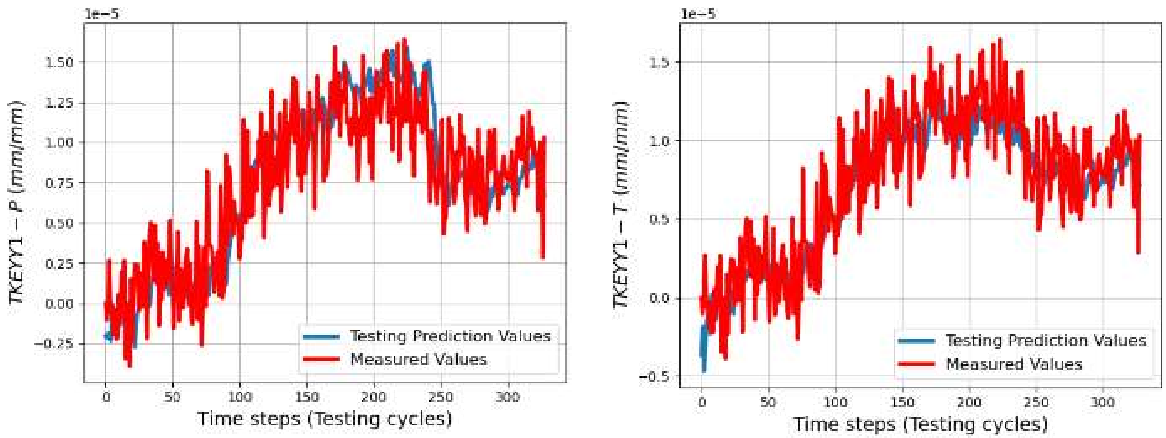

Figure 5.16. Predictions of TKEYY1-P with supply powers (P) of Y and Z-axis and ambient temperature as inputs (left) and TKEYY1-T with temperatures (T) at Y-bearing and Z-ball nut and ambient temperature as inputs (right)

Figure 5.16 shows the predictions of TKEYY1 in two cases: TKEYY1-P uses only supply powers of Y and Z-axis as inputs and TKEYY1-T uses temperatures at Y-bearing and Z-ball nut as inputs. The ambient temperature is also an input in both cases. The predictions can track the measured values and the quality of predictions using the temperatures is only slightly better than using the supply powers for this thermal error with RMSEs = 2.61×10^{-6} and 2.31×10^{-6}, respectively.

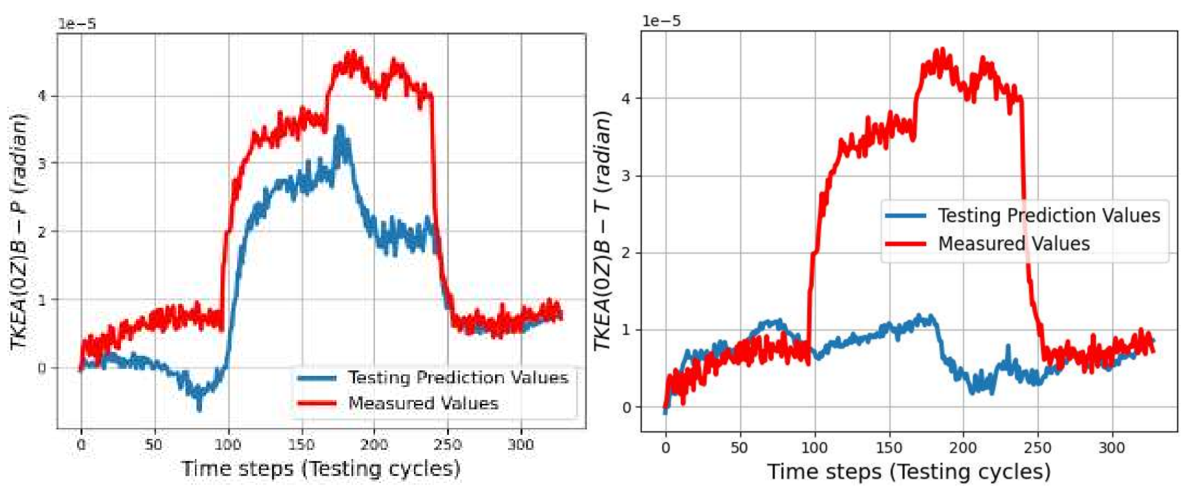

Figure 5.17. Predictions of TKEA(0Z)B-P with supply powers of B,Y and Z-axis and ambient temperature as inputs (left) and again TKEA(0Z)B-T but with temperatures (T) at B, Y-bearing and Z-ball nut and ambient temperature as inputs (right)

Conversely, it can be observed that TKEA(0Z)B-P is more closely related to the supply powers of B, Y and X-axis used as the inputs than the TKEA(0Z)B-T with the temperatures at B, Y-bearing and Z-ball nut as shown in Figure 5.17, whose RMSEs are 11.91×10^{-6} and 20.45×10^{-6}, respectively. The other cases are presented in Table 5.4 under RMSEs.

Table 5.4. RMSEs of the TKEs in cases of using the temperatures or the supply powers

Thermal kinematic errors	RMSE	
	Using supply powers as inputs	Using temperatures as inputs
TKEXX1 (mm/mm)	15.74×10^{-6}	15.18×10^{-6}
TKEZZ1 (mm/mm)	13.20×10^{-6}	12.05×10^{-6}
TKEA(0B)C (radian)	4.94×10^{-6}	6.99×10^{-6}
TKEX(0B)C (mm)	0.89×10^{-3}	0.81×10^{-3}
TKEB(0X)Z (radian)	6.78×10^{-6}	9.92×10^{-6}
TKEA(0Z)Y (radian)	13.46×10^{-6}	21.49×10^{-6}

The predictions of the trained model for some errors using the temperatures as the inputs have slightly better quality than the ones with the supply powers such as TKEXX1, TKEYY1, TKEZZ1, TKEX(0B)C, while the others in Figure 5.17 and Table 5.4 prefer the supply powers to the temperatures as the inputs.

Based on the RMSEs in Table 5.4 and Table 5.3, we propose to use the ambient temperature, the supply powers and the temperatures at axes as selected in Table 5.2 as the inputs to train and predict TKEs, which lead to better predictions of the TKEs compared to using the supply powers or the temperatures separately.

Ambient temperature

The ambient temperature always contributes to the thermal errors and the change of machine temperatures [121]. The ambient temperatures in the training and testing processes are shown in

Figure 5.5, which are between 21 and 22.5 degrees. In this part, the effect of the ambient temperatures on the accuracy of the trained model's predictions will be studied. The other inputs from the selected axes are also used as inputs.

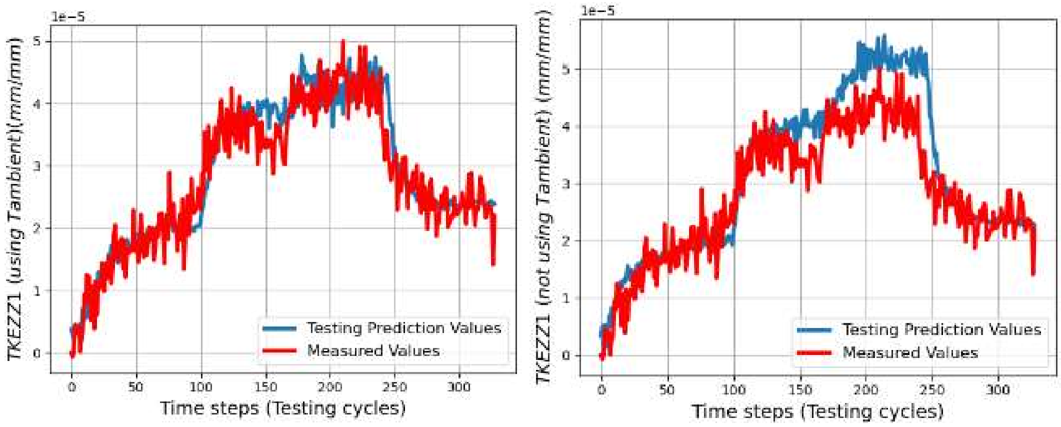

Figure 5.18. Predictions of TKEZZ1: using supply powers of Y and Z-axis and temperatures at Y-bearing and Z-ball nut with Tambient (left) and without Tambient (right) in the testing process

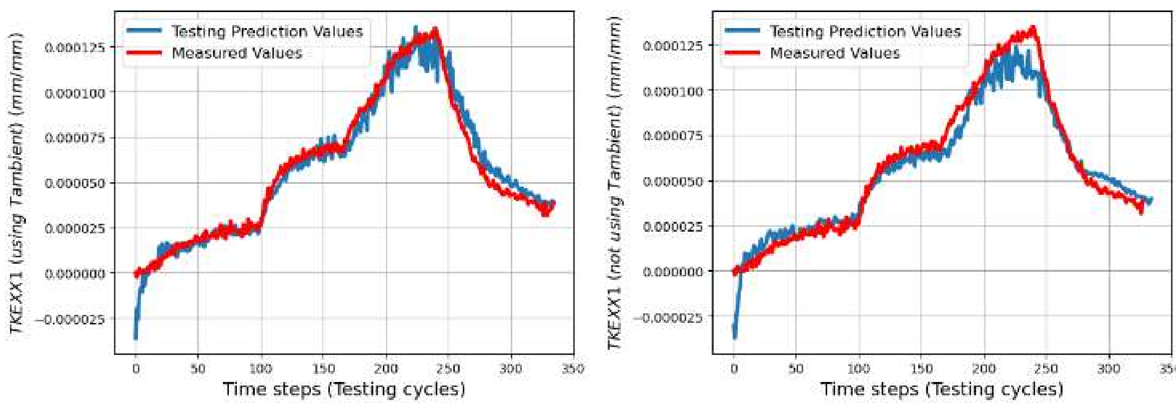

Figure 5.19. Predictions of TKEXX1: using supply powers of X, B and C-axis and temperatures at X, B, and C bearings with Tambient (left) and without Tambient (right) in the testing process

Both training and testing processes are sensitive to the ambient temperature. Figure 5.18 and Figure 5.19 show that the predictive accuracy of TKEZZ1 and TKEXX1 using the ambient temperature as the input (left) is higher than the ones depicted on the right without using ambient temperature with RMSEs of 3.9×10^{-6} and 6×10^{-6} for TKEZZ1, and 7.09×10^{-6} and 8.52×10^{-6} for TKEXX1, respectively. Table 5.5 shows the comparisons for other six thermal errors.

Table 5.5. RMSEs of the TKEs in cases of using and not using the ambient temperature as input

Thermal kinematic errors	Using Tambient as input	Not using Tambient as input
TKEYY1 (mm/mm)	2.25×10^{-6}	2.66×10^{-6}
TKEA(0B)C (radian)	2.79×10^{-6}	3.26×10^{-6}
TKEX(0B)C (mm)	0.79×10^{-3}	0.84×10^{-3}
TKEA(0Z)B (radian)	10.43×10^{-6}	11.46×10^{-6}
TKEB(0X)Z (radian)	6.03×10^{-6}	6.40×10^{-6}
TKEA(0Z)Y (radian)	9.69×10^{-6}	10.22×10^{-6}

Including ambient temperature as an input variable increases the accuracy of the predictions.

Predicted distribution

After n epochs of training process with the different dropout structures, a set of trained parameters are generated. The output distribution is obtained by applying dropout in the testing process and by recording the outputs of many forward passes through the network with the performance of the MCD on the testing data. The number of forward iterations performed to get the mean and distribution of the predictions is 300. Figure 5.20 and Figure 5.21 show the predicted boundaries of predictions, means of predictions and the measured values of fours thermal kinematic errors TKEXX1, TKEYY1, TKEZZ1, and TKEA(0B)C. The predicted boundaries are illustrated with +/- two stds between the lower bound blue line and the upper bound black line delimiting a yellow region. The std is directly proportional to the confident bounds or the range of expected predictions of the probabilistic model.

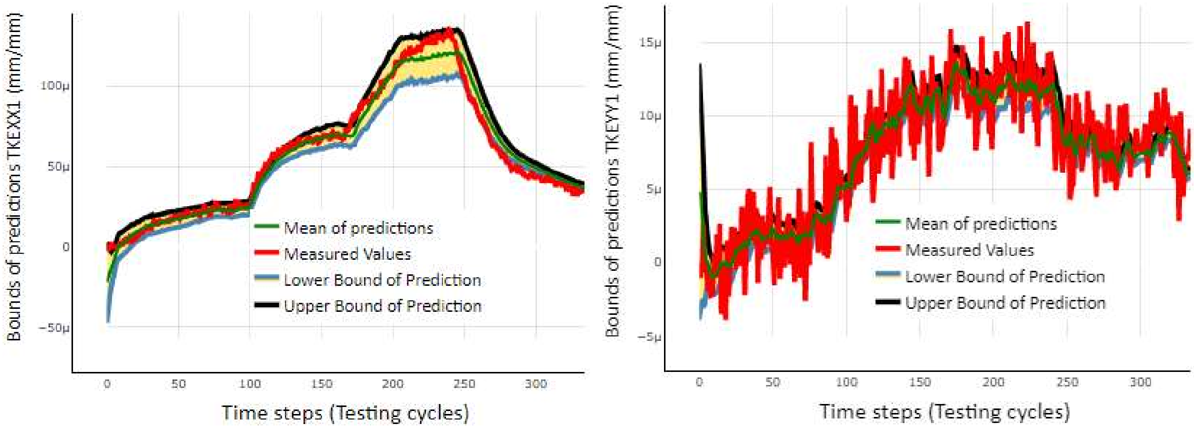

Figure 5.20. The predicted boundaries with +/- two stds for TKEXX1 (left) and TKEYY1 (right)

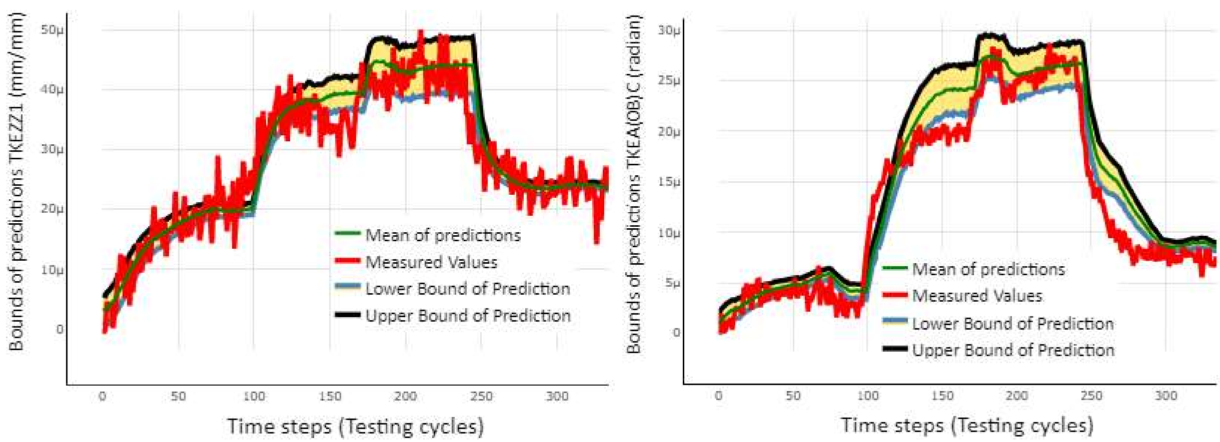

Figure 5.21. The predicted boundaries with +/- two stds for TKEZZ1 (left) and TKEA(0B)C (right)

The predicted values of TKEXX1, TKEYY1, TKEZZ1 and TKEA(0B)C at 336 cycles are tracking the measured values. The +/- two std boundaries match and cover the measured values of TKEXX1. The region is particularly large during the period when all five axes are moving at higher speed from cycle 168 to 240. The other TKEs' measured values in Figure 5.20 and Figure 5.21 regularly exceed +/- two std boundaries. Figure 5.22 and Figure 5.23 showed the predicted boundaries of the last four TKEs listed in Table 5.3.

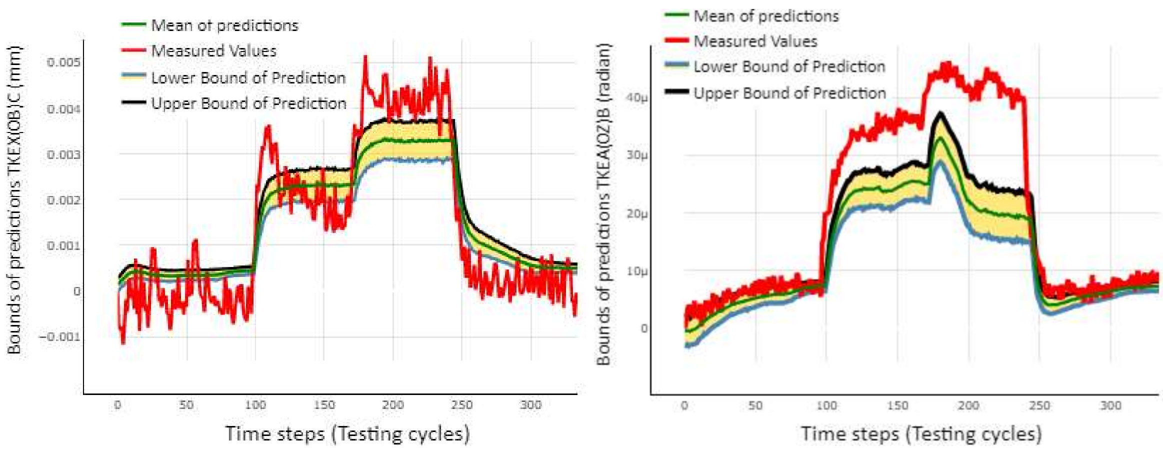

Figure 5.22. The predicted boundaries with +/- two stds for TKEX(0B)C (left) and TKEA(0Z)B (right)

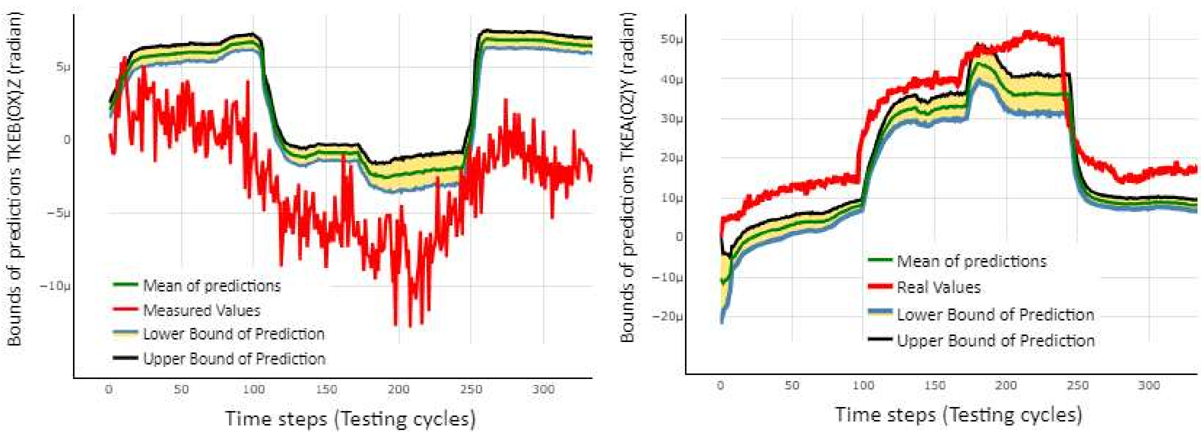

Figure 5.23. The predicted boundaries with +/- two stds for TKEB(0X)Z (left) and TKEA(0Z)Y (right)

Although the trained model's predictions tracked the measures values of TKEX(0B)C, TKEA(0Z)B, TKEB(0X)Z, and TKEA(0Z)Y, the yellow regions with +/- two stds in Figure 5.22 and Figure 5.23 do not encompass the measured values most of the time. The worst case falls on TKEB(0X)Z in Figure 5.23 (left). To evaluate the fitting between the predictions and the measured values, the RMSEs are given in Table 5.3 for the case "Inputs from axes affecting the thermal kinematic errors". The reason to select this case (not the mean of predictions (green lines) in Figure 5.20 to Figure 5.23 is that it is the best prediction of trained model for each thermal kinematic error with smallest RMSE in the training process among multiple models with different dropout structures. Table 5.6 presents the fittings, which are calculated by equation (5.7).

$$\text{fitting(\%)}=\left(1-\frac{RMSE}{Measured\ values_{max}-Measured\ values_{min}}\right)\times 100 \qquad (5.7)$$

Table 5.6. Fitting of eight thermal kinematic errors' predictions

TKEs	Fittings between the predictions and the measured values (%)
TKEXX1 (mm/mm)	95.47
TKEYY1 (mm/mm)	88.75
TKEZZ1 (mm/mm)	91.54
TKEA(0B)C (radian)	90.28
TKEX(0B)C (mm)	87.55
TKEA(0Z)B (radian)	77.52
TKEB(0X)Z (radian)	67.37
TKEA(0Z)Y (mm/mm)	81.44

The results presented in Table 5.6 shows that the trained model give the best prediction for the biggest error TKEXX1 comparing to the other seven kinematic errors with fitting = 95.47% (RMSE=$7.09\text{x}10^{-6}$). The TKEB(0X)Z has the smallest fitting with 67.37% (RMSE=$6.03\text{x}10^{-6}$). The linear positioning errors varied more and got better predictions than out of squareness angles. To enhance the reliability of the findings, z-scores and p-values are investigated across 336 cycles during the testing process for the cases in Table 5.6 as outlined below:

$$\text{zscore}=\frac{\text{x-mean}}{\text{std}} \qquad (5.8)$$

where x: predicted values of thermal kinematic errors, mean and std: presented in equations (5.5) and (5.6).

p-values is calculated by a right-sided tail test:

$$\text{p-value}=1-\phi\times\text{z-score} \qquad (5.9)$$

where ϕ is the cumulative distribution function of the standard normal distribution.

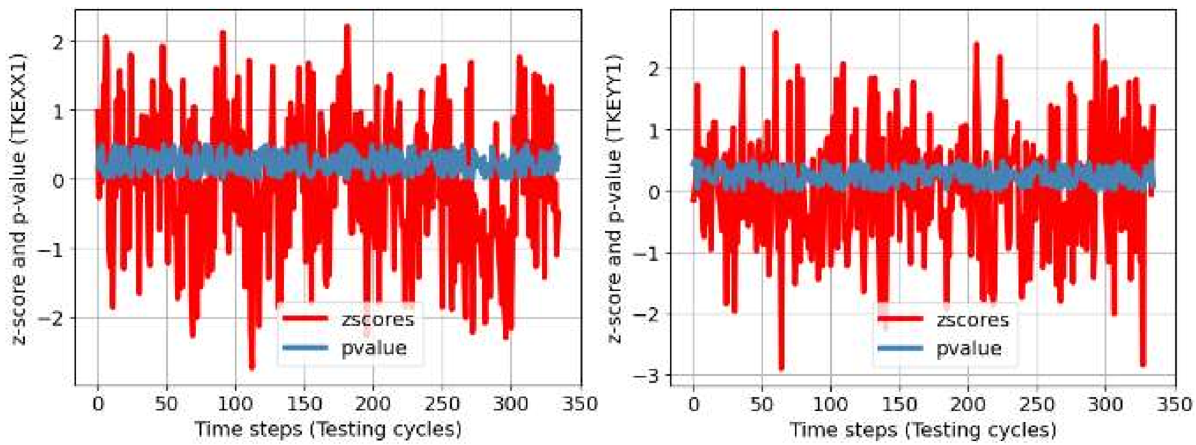

Figure 5.24. z-scores and p-values of predictions TKEXX1 (left) and TKEYY1 (right)

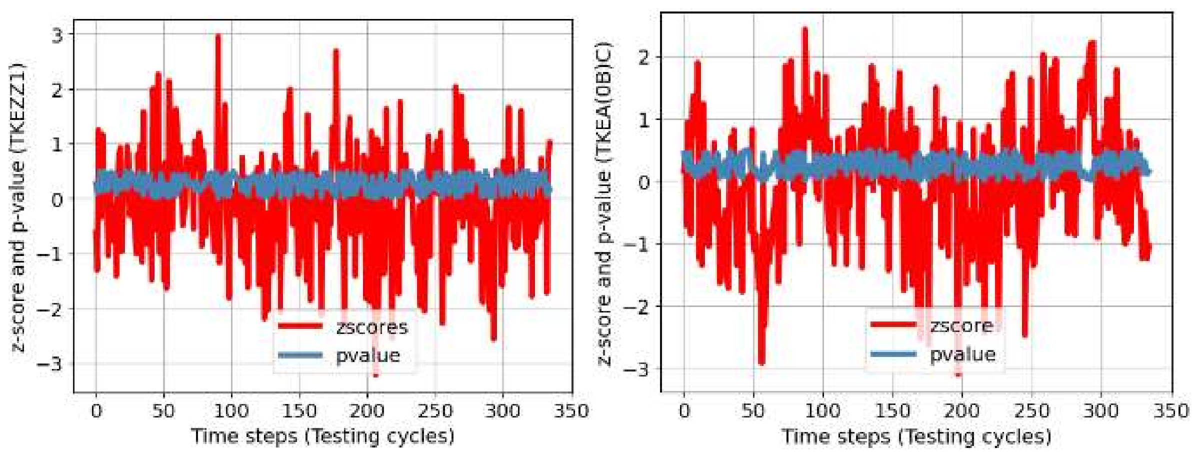

Figure 5.25. z-scores and p-values of predictions TKEZZ1 (left) and TKEA(0B)C (right)

Figure 5.24 and Figure 5.25 demonstrate the distance (number of stds) between the predictions for the cases in Table 5.6 and the mean of predictions (green lines) in Figure 5.20 and Figure 5.21 under the z-scores and probabilities of observing the predictions under p-values for the four thermal kinematic errors presented in Figure 5.20 and Figure 5.21, whose predicted boundaries with +/- two stds encompass most of the measured values. z-scores are positive if the predicted values are greater than the means of the predictions and negative if the predicted values are less than the means. The z-scores are around 2 most of the time, but there are some points with z-scores over 2 and these values are listed in Table 5.7.

Table 5.7. z-scores for four thermal kinematic errors

TKEs	z-scoremax	z-scoremin
TKEXX1 (mm/mm)	2.22	-2.73
TKEYY1 (mm/mm)	2.68	-2.89
TKEZZ1 (mm/mm)	2.95	-3.22
TKEA(0B)C (radian)	2.42	-3.1

p-values present the probability of observing the results or the confidence to accept the prediction results.

Table 5.8. p-values and the probabilities of observing the predictions (confidence)

p-value	TKEXX1	TKEYY1	TKEZZ1	TKEA(0B)C
p-value max	0.499 (50.1%)	0.497 (50.3%)	0.498 (50.2%)	0.499 (50.1%)
p-value min	0.0031 (99.69%)	0.0018 (99.88%)	0.0006 (99.94%)	0.0009 (99.91%)

The p-values and confidences of predictions of the four thermal kinematic errors are listed in Table 5.8. The probability of observing the predictions (confidence) can reach as high as 99% and the lowest confidence is 50%.

5.6 Conclusion

The thermal behavior of a five axis machine tool associated with activities of all five axes (the spindle is locked) is investigated. The data collected in the two experimental processes are temperatures at predetermined positions, supply powers of the five axes' motors, ambient temperature and 13 thermal kinematic errors to understand the thermal behavior of the machine tool. The thermal kinematic errors on the work-piece side are larger than the ones on the spindle side.

Deep learning SLSTM is employed to model and predict eight dominant TKEs in another experimental process. The prediction is best for TKEXX1 with a fitting value of 95.47% (RMSE=7.09x10^{-6}), and the predicted boundaries with +/- two stds encompassing the measured values. In contrast, the other predictions exhibit lower quality compared to TKEXX1, since the predicted boundaries with +/- two stds cannot encompass all the measured values. The worst case is for TKEB(0X)Z, with a fitting of 67.37% (RMSE=6.03x10^{-6}). The trained model predicts the linear positioning errors better than out of squareness angles.

To enhance the robustness of the model, certain measures were taken such as: selecting the relevant inputs for training and predicting the thermal kinematic errors, and considering the impact of the ambient temperature on the accuracy of the deep learning model. Monte Carlo Dropout (MCD) gives a practical method to infer the prediction distribution of the probabilistic SLSTM model. During the testing phase, the model's prediction distribution is captured by retaining the dropout mechanism and running multiple predictions. The prediction's confidences were calculated and the results indicate that the highest confidence for the predictions of the three linear positioning errors TKEXX1, TKEYY1 and TKEZZ1 and out of squareness angle TKEA(0B)C can reach as high as 99%. The confidence range or the probabilities of observing predictions is from 50% to 99%.

CHAPTER 6 GENERAL DISCUSSION

A methodology of applying deep learning for modeling the thermal variation of geometric errors and volumetric errors of the five-axis machine tool with some available inputs such as supply powers and temperatures has been proposed and presented in this thesis.

- The thermal behaviors of the machine tool are investigated and understood by the data. The data are collected from a stable and fast measurement device CAPBALL, power sensors, and thermocouples. The thermal errors are estimated by the SAMBA method.

- The effective modeling of thermal errors has been proven in another process with different activity sequences of axes. The first and second works investigated the thermal effects from the motion of rotary axes. The third work is tested with the simultaneous motions of all five axes. This would greatly facilitate the implementation in practice for real-time thermal compensation on existing machining centers. For the first work, the deep learning LSTM+Adam can predict the thermal error up to 40 hours with fitting 93% (RMSE=$8.2.10^{-6}$ for EXX1 (mm/mm)) between the predicted and measured values of thermally induced geometric errors. The second work reaches to a more complicated problem for thermally induced volumetric errors, which depend on not only the power consumptions, but also the positions in the workspace, with the highest fitting up to 91.6% (RMSE=1.8µm). The trained model in the third work can do predictions up to 84 hours without re-measuring the outputs and can have the error between predicted and measured values is RMSE = $7.09.10^{-6}$ (95.47% fitting) for EXX1 (mm/mm). The longer prediction time may be from the reason of having more information used as inputs.- The impact of the ambient temperature is unavoidable and investigated in this thesis. The work using ambient temperature as input in training and testing the model can give lower error RMSE = $7.09.10^{-6}$ comparing with the case not using ambient temperature as an input RMSE = $8.52.10^{-6}$ for EXX1. For other errors, the RMSEs keep the lower for the cases using ambient temperature as an input.

- The effective deep learning models are strengthened with optimizers and the dropout technique to keep predictions in all testing processes without re-measuring the outputs. Monte Carlo is also applied to determine the prediction distribution with +/- 2 stds (95% confidence). Finally, the hypothesis test is also done to see how reliable of the deep learning model by determining the probability of predictions falling in 95% confidence (2 stds) through the p-values.

- The limitations of this thesis are divided to two technical and theoretical problems

Technical problems: The spindle is the main heat source which is not investigated in this project. The reason is from the way to setup experiments to indirectly measure the errors. To get as more as the error parameters in indirect method, we need to set up at least three artefact balls on the work-piece side, and the CAPBALL with 5 wired capacitive sensors on the spindle side, where the spindle rotation will break the CAPBALL. If we put the CAPBALL on the work-piece side we can have one artefact ball on the spindle side. Also, the number of thermocouples are limited due to the available positions on the machine tool when the machine's axes are moving. One more limitation is that the cases investigated are only from the activities of axes without load or cooling systems. Besides, the outputs of capacitive sensors (CAPBALL) may be affected by unrecognized electromagnetic sources.

Theoretical problems: The deep learning LSTM can understand the thermal behavior of the machine tool, but there are some things to be more investigated such as: the best features (randomly selected from the total data), initial parameters of the network…to increase the prediction quality.

For all these achievements, the methodology proposed by this research work constitutes a possible solution to compensate thermal errors in advance to increase the precision of manufacturing processes.

CHAPTER 7 CONCLUSION AND RECOMMENDATIONS

The research works presented in this thesis have focused on modelling the variation of thermal errors of the five-axis machine tool with the internal heat sources, which relate to the activities of five axes. Deep learning models applied in this thesis are SLSTM and SGRU. To evaluate the robustness of the models, MCD is applied to infer the distributions of outputs and null hypothesis is used to evaluate the reliability of the predictions. The behaviors of the machine tool are understood through the data captured by data acquisition system using temperature sensors, power sensors, CAPBALL. SAMBA is used to estimate the thermal errors. The effects of ambient temperature are also investigated. Optimizers Adam and AdamW are used to improve the convergence of the training models.

The deep learning models predict the thermal errors in another process with different activities of the axes without re-measuring the outputs in a long period. The range of the models' prediction is investigated in +/- two stds (95% confidence).

The first task showed that the combination of SLSTM and Adam could model the relationship between TGEs (outputs) and motor powers of rotary axes (inputs) and predict TGEs in another shorter experiment based on new inputs.

The second task involved the use of two deep learning models, namely SLSTM and SGRU, combined with AdamW, to model and predict TVEs. TVEs are directly relating to the errors of the pathways to move machine tool to the predetermined positions in working space. This facilitate the compensation in advance.

The third task investigated the thermal behavior of five axis machine tool associated with activities of five axis machine tool (the spindle is locked). We employed deep learning SLSTM and MCD to model, predict and assess the robustness of predictions in another experimental process.

For all achievements in this thesis, the methodology proposed in this research work contributes a possible solution to predict thermal errors in advance for the compensation.

Although there is the null hypothesis test to evaluate the reliability of the prediction models, another experiment test in real part with the compensation in advance is necessary to know how really good of the deep learning model is. To do this, a synchronous program must cover many tasks from getting inputs from sensors to supply to the trained deep learning model, connecting to the controller of machine tool to provide the compensation values in x, y, z directions.

Besides the algorithms, the costly experiments should be repeated at least three times to get better-measured outputs, which may get the less variations of thermally measured errors. This improves the quality of predictions.

One more suggestion is to do prediction of the thermal errors in different starting conditions like different ambient temperatures to see how the model respond, which can give us more information to adjust the compensation values.

REFERENCES

[1] Y. Li, M. Yu, Y. Bai, Z. Hou, W. Wu. A review of thermal error modeling methods for machine tools, Applied Sciences, 11 (2021) 5216.

[2] J. Mayr, J. Jedrzejewski, E. Uhlmann, M.A. Donmez, W. Knapp, F. Härtig, K. Wendt, T. Moriwaki, P. Shore, R. Schmitt. Thermal issues in machine tools, CIRP annals, 61 (2012) 771-791.

[3] Z. Li, B. Zhu, Y. Dai, W. Zhu, Q. Wang, B. Wang. Thermal error modeling of motorized spindle based on Elman neural network optimized by sparrow search algorithm, The International Journal of Advanced Manufacturing Technology, (2022) 1-18.

[4] H. Gui, J. Liu, C. Ma, M. Li, S. Wang. Mist-edge-fog-cloud computing system for geometric and thermal error prediction and compensation of worm gear machine tools based on ONT-GCN spatial–temporal model, Mechanical Systems and Signal Processing, 184 (2023) 109682.

[5] P. Blaser, F. Pavliček, K. Mori, J. Mayr, S. Weikert, K. Wegener. Adaptive learning control for thermal error compensation of 5-axis machine tools, Journal of Manufacturing Systems, 44 (2017) 302-309.

[6] J. Mayr, P. Blaser, A. Ryser, P. Hernandez-Becerro. An adaptive self-learning compensation approach for thermal errors on 5-axis machine tools handling an arbitrary set of sample rates, CIRP Annals, 67 (2018) 551-554.

[7] N. Zimmermann, T. Büchi, J. Mayr, K. Wegener. Self-optimizing thermal error compensation models with adaptive inputs using Group-LASSO for ARX-models, Journal of Manufacturing Systems, (2022).

[8] P. Blaser, J. Mayr, K. Wegener. Long-term thermal compensation of 5-axis machine tools due to thermal adaptive learning control, MM Science Journal, 4 (2019) 3164-3171.

[9] O. Horejš, M. Mareš, L. Havlík. Compensation of Thermally Induced Errors in Five-Axis Computer Numerical Control Machining Centers Equipped With Different Spindles, Journal of Manufacturing Science and Engineering, 144 (2022) 101009.

[10] Z.-j. Li, Z. Tan, Y. Chen, Z.-c. Lu, Y.-c. Fan. Thermal error prediction of ball screw feed system based on inverse heat transfer analysis, (2022).

[11] J. Guo, Q. Xiong, J. Chen, E. Miao, C. Wu, Q. Zhu, Z. Yang, J. Chen. Study of static thermal deformation modeling based on a hybrid CNN-LSTM model with spatiotemporal correlation, The International Journal of Advanced Manufacturing Technology, 119 (2022) 2601-2613.

[12] W. Chengyang, X. Sitong, X. Wansheng. Thermal error modeling of rotary axis based on convolutional neural network, Journal of Manufacturing Science and Engineering, 143 (2021).

[13] H. Ye, X. Wei, X. Zhuang, E. Miao. An Improved Robust Thermal Error Prediction Approach for CNC Machine Tools, Machines, 10 (2022) 624.

[14] Y. Li, H. Tian, D. Liu, Q. Lu. Thermal Error Analysis of Five-Axis Machine Tools Based on Five-Point Test Method, Lubricants, 10 (2022) 122.

[15] X. Wei, H. Ye, X. Feng. Year-Round Thermal Error Modeling and Compensation for the Spindle of Machine Tools Based on Ambient Temperature Intervals, Sensors, 22 (2022) 5085.

[16] J. Peng, M. Yin, L. Cao, Q. Liao, L. Wang, G. Yin. Study on the spindle axial thermal error of a five-axis machining center considering the thermal bending effect, Precision Engineering, 75 (2022) 210-226.

[17] M. Marei, W. Li. CNC Machining Centre Thermal Error Prediction from Temperature Sensors Using Autoencoder-Augmented Long Short Term Memory Networks, in: Journal of Physics: Conference Series: IOP Publishing, 2022, pp. 012023.

[18] J. Fan, P. Wang, H. Tao, R. Pan. A thermal deformation prediction method for grinding machine'spindle, The International Journal of Advanced Manufacturing Technology, 118 (2022) 1125-1139.

[19] H. Gui, J. Liu, C. Ma, M. Li, S. Wang. New mist-edge-fog-cloud system architecture for thermal error prediction and control enabled by deep-learning, Engineering Applications of Artificial Intelligence, 109 (2022) 104626.

[20] S. Liu, Z. Yang, Q. Wei, Y. Chen, L. Liu. Thermal Error Model of Linear Motor Feed System Based on Bayesian Neural Network, IEEE Access, 9 (2021) 112561-112572.

[21] K. Ahmadi. Bayesian updating of modal parameters for modeling chatter in turning, CIRP Journal of Manufacturing Science and Technology, 38 (2022) 724-736.

[22] X.-h. Yao, J.-z. Fu, Z.-c. Chen. Bayesian networks modeling for thermal error of numerical control machine tools, Journal of Zhejiang University-SCIENCE A, 9 (2008) 1524-1530.

[23] J. Wan, Y. Che, Z. Wang, C. Cheng. Uncertainty Quantification and Optimal Robust Design for Machining Operations, Journal of Computing and Information Science in Engineering, 23 (2022) 011005.

[24] S. Park, Y. Qin. Robust regenerative chatter stability in machine tools, The International Journal of Advanced Manufacturing Technology, 33 (2007) 389-402.

[25] X. Huang, M. Hu, Y. Zhang, C. Lv. Probabilistic analysis of chatter stability in turning, The International Journal of Advanced Manufacturing Technology, 87 (2016) 3225-3232.

[26] S.C. Worland, S. Steinschneider, W. Asquith, R. Knight, M. Wieczorek. Prediction and inference of flow duration curves using multioutput neural networks, Water Resources Research, 55 (2019) 6850-6868.

[27] K. Fang, D. Kifer, K. Lawson, C. Shen. Evaluating the potential and challenges of an uncertainty quantification method for long short-term memory models for soil moisture predictions, Water Resources Research, 56 (2020) e2020WR028095.

[28] P.J. Werbos. Backpropagation through time: what it does and how to do it, Proceedings of the IEEE, 78 (1990) 1550-1560.

[29] I.C. Education. Recurrent neural networks, source: https://www ibm com/cloud/learn/recurrent-neural-networks, (2020).

[30] A. Amidi, S. Amidi. Recurrent neural networks cheatsheet, Standford University Available online: https://stanford edu/~{} shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks (accessed on 15 July 2020), (2019).

[31] A. Sagheer, M. Kotb. Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems, Scientific reports, 9 (2019) 1-16.

[32] A. Tealab. Time series forecasting using artificial neural networks methodologies: A systematic review, Future Computing and Informatics Journal, 3 (2018) 334-340.

[33] S. Hochreiter, J. Schmidhuber. Long short-term memory, Neural computation, 9 (1997) 1735-1780.

[34] Y. Sun, J. He, H. Ma, X. Yang, Z. Xiong, X. Zhu, Y. Wang. Online chatter detection considering beat effect based on Inception and LSTM neural networks, Mechanical Systems and Signal Processing, 184 (2023) 109723.

[35] C. Olah. Understanding LSTM Networks–colah's blog, Colah github io, (2015).

[36] A. Sagheer, M. Kotb. Time series forecasting of petroleum production using deep LSTM recurrent networks, Neurocomputing, 323 (2019) 203-213.

[37] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv:14061078, (2014).

[38] K.S. Narendra, A.M. Annaswamy. Stable adaptive systems: Courier Corporation, 2012.

[39] J.E. Gaudio, T.E. Gibson, A.M. Annaswamy, M.A. Bolender, E. Lavretsky. Connections between adaptive control and optimization in machine learning, in: 2019 IEEE 58th Conference on Decision and Control (CDC): IEEE, 2019, pp. 4563-4568.

[40] S. Ruder. An overview of gradient descent optimization algorithms, arXiv preprint arXiv:160904747, (2016).

[41] G. Hinton, N. Srivastava, K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent, Cited on, 14 (2012) 2.

[42] M. Riedmiller, H. Braun. Rprop-a fast adaptive learning algorithm, in: Proc of ISCIS VII), Universitat: Citeseer, 1992.

[43] D.P. Kingma, J. Ba. Adam: A method for stochastic optimization, arXiv preprint arXiv:14126980, (2014).

[44] I. Loshchilov, F. Hutter. Fixing weight decay regularization in adam, (2018).

[45] I. Loshchilov, F. Hutter. Decoupled weight decay regularization, arXiv preprint arXiv:171105101, (2017).

[46] S. Hanson, L. Pratt. Comparing biases for minimal network construction with back-propagation, Advances in neural information processing systems, 1 (1988).

[47] B.A. Olshausen. Bayesian probability theory, The Redwood Center for Theoretical Neuroscience, Helen Wills Neuroscience Institute at the University of California at Berkeley, Berkeley, CA, (2004).

[48] O. Pourret, P. Na, B. Marcot. Bayesian networks: a practical guide to applications: John Wiley & Sons, 2008.

[49] A.B. Mrad, V. Delcroix, S. Piechowiak, P. Leicester, M. Abid. An explication of uncertain evidence in Bayesian networks: likelihood evidence and probabilistic evidence, Applied Intelligence, 43 (2015) 802-824.

[50] Y. Gal. Uncertainty in deep learning, University of Cambridge, 1 (2016).

[51] S. Kullback, R.A. Leibler. On information and sufficiency, The annals of mathematical statistics, 22 (1951) 79-86.

[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research, 15 (2014) 1929-1958.

[53] E.W. Weisstein. Bernoulli distribution, https://mathworld wolfram com/, (2002).

[54] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U.R. Acharya. A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Information Fusion, (2021).

[55] K.-C. Wang, P. Vicol, J. Lucas, L. Gu, R. Grosse, R. Zemel. Adversarial distillation of bayesian neural network posteriors, in: International conference on machine learning: PMLR, 2018, pp. 5190-5199.

[56] L.V. Jospin, H. Laga, F. Boussaid, W. Buntine, M. Bennamoun. Hands-on Bayesian neural networks—A tutorial for deep learning users, IEEE Computational Intelligence Magazine, 17 (2022) 29-48.

[57] J. Maronas, R. Paredes, D. Ramos. Calibration of deep probabilistic models with decoupled bayesian neural networks, Neurocomputing, 407 (2020) 194-205.

[58] A.D. Cobb, B. Jalaian. Scaling Hamiltonian Monte Carlo inference for Bayesian neural networks with symmetric splitting, in: Uncertainty in Artificial Intelligence: PMLR, 2021, pp. 675-685.

[59] T. Karaletsos, T.D. Bui. Hierarchical gaussian process priors for bayesian neural network weights, Advances in Neural Information Processing Systems, 33 (2020) 17141-17152.

[60] A. Foong, D. Burt, Y. Li, R. Turner. On the expressiveness of approximate inference in bayesian neural networks, Advances in Neural Information Processing Systems, 33 (2020) 15897-15908.

[61] Y. Gal, Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning: PMLR, 2016, pp. 1050-1059.

[62] J. Mayr, M. Egeter, S. Weikert, K. Wegener. Thermal error compensation of rotary axes and main spindles using cooling power as input parameter, Journal of Manufacturing Systems, 37 (2015) 542-549.

[63] T.-C. Chen, C.-J. Chang, J.-P. Hung, R.-M. Lee, C.-C. Wang. Real-Time Compensation for Thermal Errors of the Milling Machine, Applied Sciences, 6 (2016) 101.

[64] E. Bitar-Nehme, J.R.R. Mayer. Modelling and compensation of dominant thermally induced geometric errors using rotary axes' power consumption, CIRP Annals, (2018).

[65] E. Uhlmann, J. Hu. Thermal modelling of a high speed motor spindle, Procedia Cirp, 1 (2012) 313-318.

[66] C.-H. Wu, Y.-T. Kung. Thermal analysis for the feed drive system of a CNC machine center, International journal of machine tools and Manufacture, 43 (2003) 1521-1528.

[67] M.O. dos Santos, G.F. Batalha, E.C. Bordinassi, G.F. Miori. Numerical and experimental modeling of thermal errors in a five-axis CNC machining center, The International Journal of Advanced Manufacturing Technology, 96 (2018) 2619-2642.

[68] C. Zhang, F. Gao, L. Yan. Thermal error characteristic analysis and modeling for machine tools due to time-varying environmental temperature, Precision Engineering, 47 (2017) 231-238.

[69] C.M. Kang, C.Y. Zhao, K. Liu, T.J. Li, B. Yang. Comprehensive compensation method for thermal error of vertical drilling center, Transactions of the Canadian Society for Mechanical Engineering, 43 (2018) 92-101.

[70] K. Liu, H. Liu, T. Li, Y. Liu, Y. Wang. Intelligentization of machine tools: comprehensive thermal error compensation of machine-workpiece system, The International Journal of Advanced Manufacturing Technology, 102 (2019) 3865-3877.

[71] S.-C. Huang. Analysis of a model to forecast thermal deformation of ball screw feed drive systems, International Journal of Machine tools and manufacture, 35 (1995) 1099-1104.

[72] B.-F. Yu, K. Liu, K.-Y. Li. Application of Multiple Regressions to Thermal Error Compensation Technology – Experiment on Workpiece Spindle of Lathe, International Journal of Automation and Smart Technology, 6 (2016).

[73] C. Brecher, P. Hirsch, M. Weck. Compensation of thermo-elastic machine tool deformation based on control internal data, CIRP Annals, 53 (2004) 299-304.

[74] A. El Ouafi, M. Guillot, N. Barka. An integrated modeling approach for ANN-based real-time thermal error compensation on a CNC turning center, in: Advanced Materials Research: Trans Tech Publ, 2013, pp. 907-915.

[75] A.M. Abdulshahed, A.P. Longstaff, S. Fletcher. The application of ANFIS prediction models for thermal error compensation on CNC machine tools, Applied Soft Computing, 27 (2015) 158-168.

[76] M. Fujishima, K. Narimatsu, N. Irino, Y. Ido. Thermal displacement reduction and compensation of a turning center, CIRP Journal of Manufacturing Science and Technology, 22 (2018) 111-115.

[77] F. Provost, T. Fawcett. Data science and its relationship to big data and data-driven decision making, Big data, 1 (2013) 51-59.

[78] S. Fletcher, A.P. Longstaff, A. Myers. Flexible modelling and compensation of machine tool thermal errors, in: 20th annual meeting of American society for precision engineering: Citeseer, 2005.

[79] M. Mares, O. Horejs, J. Hornych, J. Smolik. Robustness and portability of machine tool thermal error compensation model based on control of participating thermal sources, Journal of machine engineering, 13 (2013).

[80] B. Zhao, H. Lu, S. Chen, J. Liu, D. Wu. Convolutional neural networks for time series classification, Journal of Systems Engineering and Electronics, 28 (2017) 162-169.

[81] H. Sak, A. Senior, F. Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, arXiv preprint arXiv:14021128, (2014).

[82] Y. Yu, X. Si, C. Hu, J. Zhang. A review of recurrent neural networks: LSTM cells and network architectures, Neural computation, 31 (2019) 1235-1270.

[83] Y. .Liang, W. Li, P. Lou, J. Hu. Thermal error prediction for heavy-duty CNC machines enabled by long short-term memory networks and fog-cloud architecture, Journal of Manufacturing Systems, (2020).

[84] J. Liu, C. Ma, H. Gui, S. Wang. Thermally-induced error compensation of spindle system based on long short term memory neural networks, Applied Soft Computing, 102 (2021) 107094.

[85] P.-L. Liu, Z.-C. Du, H.-M. Li, M. Deng, X.-B. Feng, J.-G. Yang. Thermal error modeling based on BiLSTM deep learning for CNC machine tool, Advances in Manufacturing, 9 (2021) 235-249.

[86] L. Yu-Chi, L. Kun-Ying, T. Yao-Cheng. Spindle Thermal Error Prediction Based on LSTM Deep Learning for a CNC Machine Tool, Applied Sciences, 11 (2021) 5444.

[87] X. Gao, Y. Guo, D.A. Hanson, Z. Liu, M. Wang, T. Zan. Thermal Error Prediction of Ball Screws Based on PSO-LSTM, (2021).

[88] I.M. Kamal, H. Bae, S. Sunghyun, H. Yun. DERN: Deep Ensemble Learning Model for Short- and Long-Term Prediction of Baltic Dry Index, Applied Sciences, 10 (2020) 1504.

[89] C. Gao, J. Yan, S. Zhou, B. Chen, H. Liu. Long short-term memory-based recurrent neural networks for nonlinear target tracking, Signal Processing, 164 (2019) 67-73.

[90] H. Rizk, M. Youssef. Monodcell: A ubiquitous and low-overhead deep learning-based indoor localization with limited cellular information, in: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019, pp. 109-118.

[91] O. Ogunmolu, X. Gu, S. Jiang, N. Gans. Nonlinear systems identification using deep dynamic neural networks, arXiv preprint arXiv:161001439, (2016).

[92] K. Lv, S. Jiang, J. Li. Learning gradient descent: Better generalization and longer horizons, arXiv preprint arXiv:170303633, (2017).

[93] S. Fernández, A. Graves, J. Schmidhuber. Sequence labelling in structured domains with hierarchical recurrent neural networks, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007, 2007.

[94] R. Chandra, S. Goyal, R. Gupta. Evaluation of deep learning models for multi-step ahead time series prediction, arXiv preprint arXiv:210314250, (2021).

[95] S. Ibaraki, W. Knapp. Indirect measurement of volumetric accuracy for three-axis and five-axis machine tools: a review, International Journal of Automation Technology, 6 (2012) 110-124.

[96] O. Horejs, M. Mares, P. Kohut, P. Barta, J. Hornych. Compensation of machine tool thermal errors based on transfer functions, MM Science Journal, 3 (2010) 162-165.

[97] Y. Liu, Y. Lu, D. Gao, Z. Hao. Thermally induced volumetric error modeling based on thermal drift and its compensation in Z-axis, The International Journal of Advanced Manufacturing Technology, 69 (2013) 2735-2745.

[98] C. Baum, C. Brecher, M. Klatte, T.H. Lee, F. Tzanetos. Thermally induced volumetric error compensation by means of integral deformation sensors, Procedia CIRP, 72 (2018) 1148-1153.

[99] J. Liu, C. Ma, S. Wang. Data-driven thermally-induced error compensation method of high-speed and precision five-axis machine tools, Mechanical Systems and Signal Processing, 138 (2020) 106538.

[100] H.V. Ngoc, J.R.R. Mayer, E. Bitar-Nehme. Deep learning LSTM for predicting thermally induced geometric errors using rotary axes' powers as input parameters, CIRP Journal of Manufacturing Science and Technology, 37 (2022) 70-80.

[101] P. Dahlem, D. Emonts, M.P. Sanders, R.H. Schmitt. A review on enabling technologies for resilient and traceable on-machine measurements, Journal of Machine Engineering, 20 (2020) 5--17.

[102] N.A. Mchichi, J. Mayer. Axis location errors and error motions calibration for a five-axis machine tool using the SAMBA method, Procedia CIRP, 14 (2014) 305-310.

[103] F.E. Curtis, K. Scheinberg. Adaptive Stochastic Optimization: A Framework for Analyzing Stochastic Optimization Algorithms, IEEE Signal Processing Magazine, 37 (2020) 32-42.

[104] S. Gugger, J. Howard. AdamW and Super-convergence is now the fastest way to train neural nets, last accessed, 19 (2018).

[105] E. Bitar-Nehme, J.R.R. Mayer. Thermal volumetric effects under axes cycling using an invar R-test device and reference length, International Journal of Machine Tools and Manufacture, 105 (2016) 14-22.

[106] S. Weikert. R-test, a new device for accuracy measurements on five axis machine tools, CIRP Annals-Manufacturing Technology, 53 (2004) 429-432.

[107] J.R.R. Mayer. Five-axis machine tool calibration by probing a scale enriched reconfigurable uncalibrated master balls artefact, CIRP annals, 61 (2012) 515-518.

[108] S. Sepahi-Boroujeni, J.R.R. Mayer, F. Khameneifar. Efficient uncertainty estimation of indirectly measured geometric errors of five-axis machine tools via Monte-Carlo validated GUM framework, Precision Engineering, 67 (2021) 160-171.

[109] S. Sepahi-Boroujeni, J.R.R. Mayer, F. Khameneifar, A. Woźniak. A full-covariance uncertainty assessment in on-machine probing, International Journal of Machine Tools and Manufacture, 167 (2021) 103768.

[110] H.V. Ngoc, J. Mayer, E. Bitar-Nehme. Deep Learning to Directly Predict Compensation Values of Thermally Induced Volumetric Errors, Machines, 11 (2023) 496.

[111] A. Der Kiureghian, O. Ditlevsen. Aleatory or epistemic? Does it matter?, Structural safety, 31 (2009) 105-112.

[112] R. Ramesh, M. Mannan, A. Poo, S. Keerthi. Thermal error measurement and modelling in machine tools. Part II. Hybrid Bayesian Network—support vector machine model, International Journal of Machine Tools and Manufacture, 43 (2003) 405-419.

[113] M. Papananias, S. Fletcher, A.P. Longstaff, A. Mengot. A novel method based on Bayesian regularized artificial neural networks for measurement uncertainty evaluation, in: european society for precision engineering and nanotechnology, 2016.

[114] H. Shi, C. Jiang, Z. Yan, T. Tao, X. Mei. Bayesian neural network–based thermal error modeling of feed drive system of CNC machine tool, The International Journal of Advanced Manufacturing Technology, 108 (2020) 3031-3044.

[115] J. Karandikar, K. Saleeby, T. Feldhausen, T. Kurfess, T. Schmitz, S. Smith. Evaluation of automated stability testing in machining through closed-loop control and Bayesian machine learning, Mechanical Systems and Signal Processing, 181 (2022) 109531.

[116] J. Heo, H.B. Lee, S. Kim, J. Lee, K.J. Kim, E. Yang, S.J. Hwang. Uncertainty-aware attention for reliable interpretation and prediction, in: Advances in neural information processing systems, 2018, pp. 909-918.

[117] J. van der Westhuizen, J. Lasenby. Bayesian LSTMs in medicine, arXiv preprint arXiv:170601242, (2017).

[118] M. Fujishima, K. Narimatsu, N. Irino, M. Mori, S. Ibaraki. Adaptive thermal displacement compensation method based on deep learning, CIRP journal of manufacturing science and technology, 25 (2019) 22-25.

[119] P. Baldi, P.J. Sadowski. Understanding dropout, Advances in neural information processing systems, 26 (2013).

[120] A. Abdulshahed, A.P. Longstaff, S. Fletcher. A particle swarm optimisation-based Grey prediction model for thermal error compensation on CNC machine tools, in: Euspen, 2015.

[121] B. Tan, X. Mao, H. Liu, B. Li, S. He, F. Peng, L. Yin. A thermal error model for large machine tools that considers environmental thermal hysteresis effects, International Journal of Machine Tools and Manufacture, 82 (2014) 11-20.