# DeepMachining: Online Prediction of Machining Errors of Lathe Machines

Xiang-Li Lu, Hwai-Jung Hsu, Che-Wei Chou, H. T. Kung, Chen-Hsin Lee, and Sheng-Mao Cheng

Abstract—**We describe DeepMachining, a deep learning-based AI system for online prediction of machining errors of lathe machine operations. We have built and evaluated DeepMachining based on manufacturing data from factories. Specifically, we first pretrain a deep learning model for a given lathe machine's operations to learn the salient features of machining states. Then, we fine-tune the pretrained model to adapt to specific machining tasks. We demonstrate that DeepMachining achieves high prediction accuracy for multiple tasks that involve different workpieces and cutting tools. To the best of our knowledge, this work is one of the first factory experiments using pre-trained deep-learning models to predict machining errors of lathe machines.**

Index Terms—**Deep learning, convolutional neural network, pre-trained model, model fine-tuning, online prediction of machining errors, lathe machines, computer numerical control (CNC) machine.**
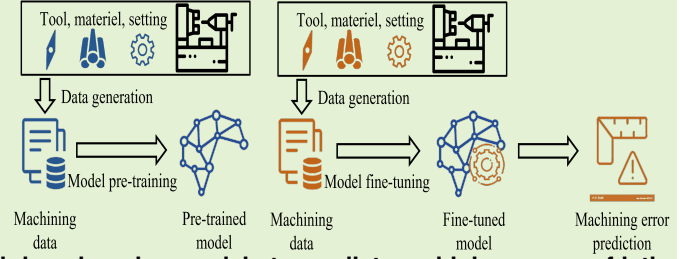
I. INTRODUCTION

THe structures of modern manufacturing devices are increasingly complex, while tolerance requirements for possible machining errors become more strict. In the manufacturing of high-precision parts, high-quality machining with low errors is essential.

For lathe machines, popular in the manufacturing of precision parts, various machining errors such as geometric tooling, thermal-induced, and load-induced errors [1, 2], etc., can lead to inaccuracies above the tolerance level of manufactured workpieces, resulting in monetary losses to the manufacturers. Early detection of manufacturing quality degradation and process anomalies [3, 4], and assessment of the wear of cutting tools in material removal processes [5] can help reduce such risks. In particular, implementing real-time monitoring and online machining quality prediction can enhance error detection's efficiency and efficacy.

In recent years, tool condition monitoring (TCM), enabled by sensor technology and artificial intelligence (AI), has been employed to address these needs [6]. For example, TCM has been widely used for fault detection and diagnosis (FDD) [7–10], predictive maintenance (PdM) [11–14], prognostics and health management (PHM) [4, 15, 16], etc. in the manufacturing industry.

Deep-learning-based AI driven by manufacturing data is a promising approach for error detection, given that these data-driven methods have been successful in fields like computer vision and natural language processing [4, 7, 10, 13, 16, 17]. However, applying deep learning techniques to manufacturing brings new challenges, such as model generalization

for factory environments. For example, real-world machining processes involve a variety of workpiece materials, cutting tools, process recipes, and equipment models. As a result, supervised deep-learning models trained on signals from sensors of specific CNC machines may not apply to other machines. In other words, AI-powered solutions may not generalize to diverse manufacturing environments [18].

We may apply the classical transfer learning approach [19–21] to address the model generality issue, where a pre-trained models trained on a large labeled dataset is fine-tuned to the target task. However, acquiring the abnormal data corresponding to machining states that lead to the manufacture of erroneous workpieces is extremely costly in the machinery industry. Thus, gathering sufficient amounts of high-quality data for training the pretrained model is challenging. In addition, the limited computational resources of the CNC machine must be addressed to ensure that the AI solution is deployable.

To address these challenges, this paper develops DeepMachining, a deep learning-based AI system, to predict machining errors utilizing the pre-trained model. As Fig. 1 shows, the pre-trained model was trained over the lifetime of the cutting tool until it was completely worn out. For model generalization, we perform model pretraining involving multiple spindle speeds. For fine-tuning, we propose a method similar to BitFit [22], which adjusts the model's biases. This allows the pre-trained model to adapt to the target tasks using few-shot learning (typically two-shot). In other words, fine-turning uses data collected from two instances of the target machining task. Merely 6.5% of the total parameters of the model are fine-tuned in less than 12.5% of epochs of the model pre-training. Thus, the proposed fine-tuning method not only suits existing machining processes but can also be completed with the
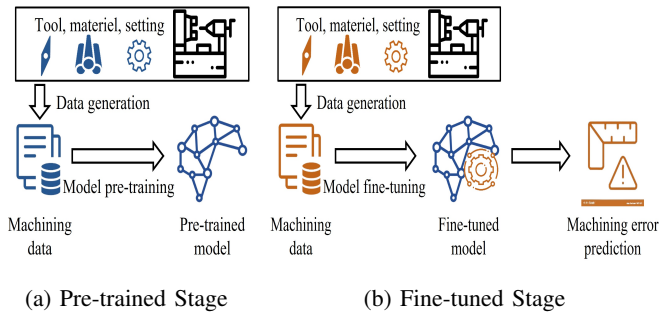
(a) Pre-trained Stage   (b) Fine-tuned Stage

Fig. 1: The Process of DeepMachining

limited computational power of the industrial computers in the CNC machines.

To evaluate the proposed approach in predicting machining errors under various manufacturing settings, we use four machining tasks for the validation.

The main contributions of this paper are:

- The proposed DeepMachining approach, and performed validation showing that, under the approach, we can pre-train a model that can be adapted to various downstream tasks.
- A few-shot model fine-tuning method (typically, two-shot) for adaptation to new manufacturing settings.
- The useful insight that the fine-tuning required in these manufacturing tasks is basically shifts of model's biases.
- An end-to-end factory demonstration of DeepMachining based in real-world factories.

The rest of this paper is organized as follows. Section II reviews the related literature on TCM and its applications. Section III addresses the DeepMachine framework for on-line prediction of machining errors. Section IV details the experiments and analysis using real world machining tasks in factories. Section V discusses the limitations and lessons learned in this study. Conclusions are drawn in Section VI.

II. RELATED WORK

Machining error, surface roughness, and tool wear are key quality control metrics in machining processes. Intelligent sensors, including accelerometers, data acquisition encoders, acoustic emission sensors, microphones, dynamometers, and image sensors, are utilized to monitor and diagnose machine health degradation and process anomalies [23]. Accelerometer sensors are sensitive and reliable in measuring workpiece dimensions in high precision [24, 25]. Therefore, we adapted accelerometers, DAQ encoder, and microphone to collect manufacturing data building DeepMachining for predicting machining error in this study.

Traditional machine learning (ML) approaches have been used to predict product quality during CNC machining tasks. Du et al. [26] proposed a power spectral density based feature extraction method from spindle vibration and cutting force signals, which accurately predicted product roughness, profile, and roundness using tree-based regressor approaches in hard turning processes. Denkena et al. [27] optimized workpiece quality and tool life in cylindrical turning processes by identifying the machined material based on machine

learning algorithms. Papananias et al. [28] proposed principal component analysis (PCA) based multilayer perceptron (MLP) networks to accurately predict the true position and circularity requirements of a workpiece in an experimental setting. ML approaches could predict well on collected numerous datasets in manufacturing scenarios. However, ML approaches cannot be used as the kernel for pre-trained models to adapt to various downstream tasks via fine-tuning.

The advent of deep learning has reformed predictive approaches, enabling end-to-end prediction and diagnosis procedures to enhance CNC machining precision and reliability within smart tool condition monitoring systems [9, 29]. Huang and Lee [30] proposed one-dimensional convolutional neural network (1D-CNN) and sensor fusion approach accurately estimated tool wear and surface roughness for the CNC machining. Hesser and Markert [31] demonstrated the feasibility of predicting CNC machine status and tool wear for maintenance plan using artificial neural networks. Proteau et al. [32] proposed a variational autoencoder (VAE) regression model to predict the geometrical and dimensional tolerances of workpieces using sensor data in industrial settings. Zhu [33] established a long short-term memory (LSTM) model for one-dimensional time series and CNN for two-dimensional images.

Transformer-based networks have been applied to capture association relationships and dependency from vibration signals through the self-attention mechanism for improving performances of the developing models in recent years, [34–38]. Wu et al. [34] and Li et al. [35] studied fault detection and classification in a rotary system with transformer-based models. Li et al. [36] and Liu et al. [38] applied for tool wear prediction in TCM topics. Compared to transformer-based approaches, in this study we utilize 1D-CNN networks with attention mechanism to address the time series data of vibration signals, considering latency and computing power for prompt inference in practice.

Transfer learning (TL), which learns two types of networks to extract representations, solves cross-domain diagnosis problems with small and imbalanced data [9, 39–41]. Wang and Gao [42] proposed a CNN-based transfer learning technique using vibration analysis for rolling bearing fault diagnosis. Specifically, adapting a pre-trained VGG19 network [43], using non-manufacturing images from ImageNet [44] (i.e., model transfer) and transferring the adapted network structure to different fault severity levels and bearing types (i.e., feature transfer). Guo et al. [45] proposed a deep convolutional transfer learning network to classify bearing health conditions with unlabeled data. Bahador et al. [46] investigated a transfer learning approach for classifying tool wear based on tool vibration in hard turning processes. Ross et al. [47] proposed a transfer learning model with Inception-V3 network [48] to detect tool flank wear under distinct cutting environments.

However, the research gap between practitioners and researchers remains in practice [18]. Different processing parameters result in different data distributions, which poses a significant challenge to ML and DL models. Collecting and labeling data with different combinations of materials, tools, process recipes, and machines in practice is difficult and expensive. Furthermore, even if labeled data is obtained
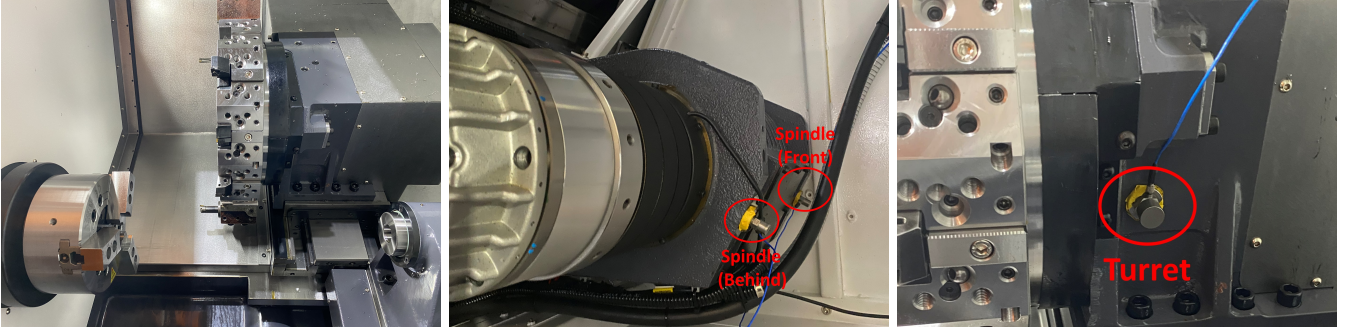
(a) Internal Structure of CNC Lathe Machine (b) Accelerometer Placement on the Spindle (c) Accelerometer Placement on the Turret

Fig. 2: The experimental environment of the CNC lathe machine

from some manufacturing conditions, the resulting predictive models may fail to classify unlabeled data due to intricate manufacturing settings and data distribution discrepancies [39, 47]. Therefore, designing pre-trained models that are easily adaptable to empirical field applications with strong performance is an important topic [49–51]. Furthermore, fine-tuning on task-specific supervised data enables seamless adaptation to various specific tasks in practical settings [22, 52–54].

III. METHODOLOGY

A. Problem Definition

The machining error is the difference between the dimension measured after the machining of a workpiece and the target dimension described in the specification. The proposed DeepMachining estimates machining errors under various processing conditions, e.g., different combinations of machining tools and configurations, on CNC lathes without actual measurement. Several factors can impact the machining error of a workpiece. These include the wear condition of cutting tools, the hardness and processing difficulty of the material, the environmental temperature (thermal expansion), and the wear of machine components on the equipment (i.e., the lathe). In order to perceive the factors, accelerometers are installed to collect the vibration signals that occur during the machining process; the machine status, such as the spindle speed and motor current, etc., during the machining process is also recorded. Besides, it's important to note that any specific section of a workpiece can be machined multiple times. In other words, multiple cutting processes may be performed at the same place on a workpiece to achieve the target size. The signals and data generated from multiple machining sessions should be gathered and processed.

B. Machine Settings

This study conducted experiments on a horizontal CNC lathe machine, which features an internal spindle and three-axis linear guides, as shown in Fig. 2. Piezoelectric accelerometers are deployed at three distinct positions, (1) behind the spindle, as depicted in Fig. 2b, (2) in front of the spindle, also shown in Fig. 2b, and (3) at the base of the tool turret, illustrated in Fig. 2c, to collect relevant vibration signals. The machine controller records and outputs the spindle speed and

current of the drive motors for the spindle and the turret during the machining process, which serves as the machine status.

C. Input Formulation

In order to predict the machining error $y \in \mathbb{R}^1$, two inputs $\mathcal{X} = \{X_1, X_2, \ldots, X_n\} \in \mathbb{R}^{N \times SR \times C_1}$, including the vibration signals and machine status during the machining process, and $\tilde{\mathcal{X}} = \{\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_n\} \in \mathbb{R}^{N \times (\frac{SR}{2}+1) \times C_2}$, the transformation of vibration signals in $\mathcal{X}$ from time domain to frequency domain using Fourier Transform [55], are used. The duration of each input $X_i \in \mathcal{X}$ is one second around the location of the workpiece where the machining error y is measured. N is the number of cuts, C_1 and C_2 are the number of input channels, and SR indicates the sampling rate of the sensors used in input collection.

D. The Core of DeepMachining

The core of DeepMchining, as illustrated in Figure 3, is a two-stage model handling multiple cuttings across machining processes to estimate the machining error of a workpiece. In the first stage, Dual Signal Encoder, one for $X_n \in \mathcal{X}$ and the other for $\tilde{X}_n \in \tilde{\mathcal{X}}$, plays the primary role of feature extraction. Each Signal Encoder contains three blocks, Stem, Downsampling, and Dilated-Inception (D-Inception). Stem retains input representation while reducing its sampling rate. D-Inception and Downsampling are then stacked to extract features. The Signal Encoder simultaneously utilizes Global Average Pooling (GAP) and Global Max Pooling (GMP) to summarize the extracted features and concatenates them to form the output.

In the second stage, hidden states $\mathcal{H} = \{H_1, H_2, \ldots, H_n\} \in \mathbb{R}^{N \times 4d}$ extracted from the features produced by the Dual Signal Encoder for each cut are concatenated and processes using an additional D-Inception. d is the dimension of features in the Signal Encoder. Since the Signal Encoder simultaneously concatenates the outputs from the GAP and GMP as its own output. Lastly, the output hidden state from the second stage is used to estimate machining error with a Projection Head, a single-layer feed-forward network.

The details regarding Stem, D-Inception, and Downsampling are explained in the following sections.
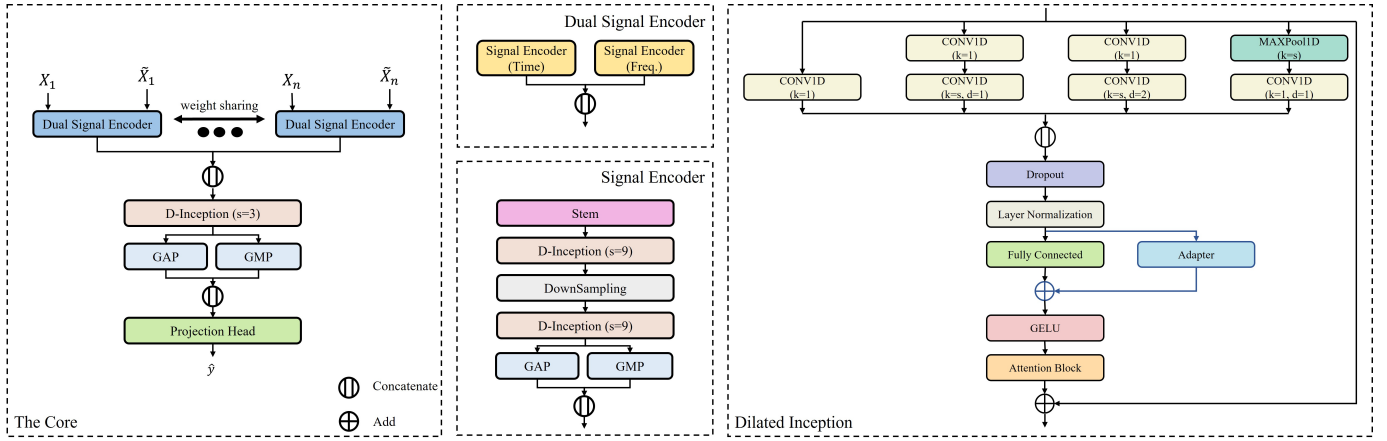
Fig. 3: The Structure of The Core of DeepMachining

1) Stem:

Stem reduces input length by lowering the sampling rate using 1D convolution with strides, and thus is learnable. Stem is used at the beginning of the model to reduce the computation costs, with minimal loss of information. The equation of Stem is described as following:

$$Stem(X_n) = \delta(f^{11}(LN(Dropout(W(X_n))))) \quad (1)$$

LN stands for Layer Normalization [56], f^{11} is 1D Convolution whose kernel size equals 11 and strides equals 5, δ is GELU [57], $W \in \mathbb{R}^{C \times d}$ is a learnable weight used to project the input into high dimension space, $C \in \{C_1, C_2\}$ and d are the input and the output channel dimension.

2) Dilated Inception:

Dilated Inception (D-Inception) is established for extracting features. The D-Inception incorporates the idea behind Inception, using multiple branches with different convolutions to get features with multiple receptive fields, into transformer-like design, e.g. replacing Batch Normalization [58] with Layer Normalization [56], replacing activation function ReLU [59] with GELU [57], and removing activation functions from the convolution layers in the bottleneck [60].

Similar to GoogleLenet [48], D-Inception utilizes a four-branched Inception, and applies dilated convolution [61] which has the same receptive fields with fewer parameters, instead of normal convolution in some blocks. Applying dilated convolution [61] allows models to be light-weighted with few performance loss. D-Inception is composed of four different one-dimensional (1D) convolution block, normal 1D-convolution with kernel size of 1 (CONV_1) and s (CONV_s), and dilated convolution with dilated size of 2 and size of 1 (DCONV_1) and s (DCONV_s). The four branches are composed of (1) one CONV_1, (2) one CONV_1 and one CONV_s, (3) one CONV_1 and one DCONV_s, and (4) one max-pooling block of size s and one CONV_1. s is a hyper-parameter indicating the kernel size of some CONV_s and DCONV_s in D-Inception. The value of s varies along with the length of the input of a D-Inception block. The outputs of the four branches are concatenated as the features extracted by the convolutional layers. The features are then processed with Dropout [62] and Layer Normalization [56], and passed to a fully connected layer with activation function of GELU [57] to form $F \in \mathbb{R}^{L \times d}$ as output. L is the length of the output.

Moreover, similar to CBAM [63], an Attention Block processing the F added to improve model performance with limited increase of number of model parameters. The Attention Block sequentially calculates a channel attention map $M_c \in \mathbb{R}^{1 \times d}$ and a temporal attention map $M_t \in \mathbb{R}^{L \times 1}$. Our Attention Block is defined as follows:

$$\begin{aligned} F' &= M_c(F) \otimes F, \\ F'' &= M_t(F') \otimes F' \end{aligned} \quad (2)$$

The channel attention aggregates the temporal information from feature F using average-pooling and max-pooling into F_{avg}^c and F_{max}^c. F_{avg}^c and F_{max}^c are then concatenated to feed into a two-layer feed-forward network for calculating attention between features along with channels, and F' is produced as the output of channel attention. The feed-forward network has different activation functions among the layers that GELU (δ) [57] for the first layer and sigmoid (σ) for the second. GELU [57] gives non-linear transformation to the features, and sigmoid scale the output of the attention block between 0 and 1. The calculation of M_c is defined as follows:

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)||MaxPool(F))) \\ &= \sigma(W_1(\delta(W_0(F_{avg}^c||F_{max}^c)))) \end{aligned} \quad (3)$$

$W_0 \in \mathbb{R}^{\frac{d}{r} \times d}$ and $W_1 \in \mathbb{R}^{d \times \frac{d}{r}}$ are learnable parameters, and r is the reduction ratio used to reduce parameter overhead.

As with channel attention, temporal attention consumes F' as input, and uses average-pooling and max-pooling on temporal dimension of F' as $F_{avg}^{'t}$ and $F_{max}^{'t}$ separately. $F_{avg}^{'t}$ and $F_{max}^{'t}$ are the concatenated, and processed with 1D convolution with kernel size of 1. The calculation of M_t is defined as follows:

$$\begin{aligned} M_t(F') &= \sigma(f^1(AvgPool(F')||MaxPool(F'))) \\ &= \sigma(f^1(F_{avg}^t||F_{max}^t)) \end{aligned} \quad (4)$$

f^1 is 1D Convolution whose kernel size equals 1. $\otimes$ denotes element-wise multiplication.

Attention values are broadcast as follows: channel attention values are broadcast along the temporal dimension, and

temporal attention values are broadcast along the channel dimension. Channel and temporal attention weights determine the importance of features and enhance or degrade them accordingly. The output of the attention block is added with the raw input of D-Inception brought by the Residual Connection [64].

3) Downsampling:

Similar to Stem, Downsampling reduces input length. However, since Downsampling uses max-pooling to achieve the reduction, it is not learnable. Stacking Downsampling with D-Inception increases the receptive field of D-Inception blocks. The equation of Downsampling is described as follows:

$$Downsampling(F) = W_1(LN(W_0(MaxPool(F)))) \quad (5)$$

LN stands for Layer Normalization [56], and $MaxPool$ represents max-pooling. $W_0 \in \mathbb{R}^{4d \times d}$ and $W_1 \in \mathbb{R}^{d \times 4d}$ are learnable weights that shuffle features based on a bottleneck structure.

E. Fine-tuning Method

In order to fine-tune the model for diverse machining tasks, we adopt a methodology inspired by TinyTL [65]. TinyTL inserts an additional lightweight residual module into the pre-trained model, and trains only the residual module, bias, and regressor during the fine-tuning process.

Instead of the lightweight residual module, we insert an Adapter composed of two feed-forward layers for feature enhancing and shuffling in conjunction with the pre-trained model. Adapters are inserted into D-Inception and Downsampling blocks. During fine-tuning, only the Adapter, Bias, and Projection Head undergo training while all other parameters remain frozen. The Adapter is formally defined as follows:

$$Adapter(F) = F + W_1(W_0(F)) \quad (6)$$

$W_0 \in \mathbb{R}^{d \times \frac{d}{r}}$ and $W_1 \in \mathbb{R}^{\frac{d}{r} \times d}$ are learnable weights. r indicates the reduction ratio, and $F \in \mathbb{R}^{L \times d}$ is the feature in D-Inception or Downsampling.

In practice, workpiece dimensions are measured during CNC machine restarts, process resets, or when operators deem reconfiguration necessary. Then, the operators adjust the cutting tools and machine parameters on the basis of the measurements to ensure the accuracy of subsequent machining. Therefore, the proposed DeepMachining incorporates few-shot learning (typically two-shot) to fine-tune the pre-trained model at these instances to adapt the variations in the machining contexts, parameters, and diverse changes in workpieces and cutting tools. In summary, the core of DeepMachining is a relatively small model with around **260,000** parameters. Only **6.5%** of the parameters needs to be fine-tuned with **12.5%** of epochs of the pre-training process for adaptation to various machining tasks and configuration.

IV. EXPERIMENTS

A. Settings

Datasets: The datasets were collected from three distinct machining tasks, and were named on the basis of the material

TABLE I: Machining Configuration of Datasets

Dataset	Spindle RPM	Feed Rate (mm/rev)	# of Configuration Changes
WC_AO-MS	1100 to 2700	[0.25,0.1]	14
WC_TAN-MS	1600 to 2200	[0.25,0.12]	2
WC_TC-AS	1000 to 2100	[0.12,0.25]	3

and coating of the cutting tool, as well as the material of the workpieces under machining. All of the cutting tools used in the experiments were made of Tungsten Carbide (WC). The coatings of the cutting tools include Aluminium Oxide (AO), Titanium Aluminium Nitride (TAN), and Titanium Carbonitride (TC). The materials of the workpieces included Medium-Carbon Steel (MS) and Alloy Steel (AS). On the other hand, except for the vibration and the machine controller signals, adjustments to the machining configurations (e.g. spindle speed, initial tool position) and context changes (e.g. machining dates) along with the machining processes were also recorded. TABLE I shows the summary of each dataset, and the details are described as follows:

- WC_AO-MS: 347 MS workpieces were machined using a tool made of WC and coated with AO. The workpieces were machined on seven different dates, with varying spindle speeds for each date. Besides, according to the judgment of on-site personnel, the cutting tool underwent eight position adjustments to offset its machining precision. Besides the first machining, the acts of machining on the other dates and the too position adjustments are considered as a configuration change. This dataset was used for model pre-training. To evaluate the performance of the pre-trained model, testing dataset was split from the dataset for assessment. The testing dataset was generated in two different ways. First, 80% of the data was randomly selected for training, and the remaining 20% for testing. Second, the first 80% of the dataset (sequenced by machining time) was used for training, and the remaining for testing. In the following sections, the first dataset is named as WC_AO-MS (Random) and the second one as WC_AO-MS (Sequential).
- WC_TAN-MS: 87 MS workpieces were machined using a tool made of WC and coated with TAN. The workpieces were machined on two different dates, with varying spindle speeds for each date. Since we plan to fine-tune the pre-trained model to adapt the tool differences in WC_TAN-MS, each machining date in WC_TAN-MS is considered as a configuration change. To assess whether the pre-trained model can adapt to changes in cutting tools through fine-tuning, few learning is applied for each machining date, i.e. configuration change, as the section III-E describes. In other words, for each date, the first two workpieces are used for model fine-tuning, and the remaining ones are used for testing.
- WC_TC-AS: 34 AS workpieces were machined using a tool made of WC and coated with TC. All the workpieces were machined on the same date. However, there were three instances of configuration change: (1) when the

TABLE II: Train/Test Split of Pre-trained Datasets

Dataset	Train (#Workpieces)	Test (#Workpieces)	Total (#Workpieces)
WC_AO-MS (Random)	277[277]	70[70]	347
WC_AO-MS (Sequential)	281[277,2,2]	66[11,7,48]	347

TABLE III: Train/Test Split of Adapted Datasets

Dataset	Train (#Workpieces)	Test (#Workpieces)	Total (#Workpieces)
WC_TAN-MS	4[2,2]	83[37,46]	87
WC_TC-AS	6[2,2,2]	28[5,2,19]	34

TABLE IV: Performance Comparison of Pre-trained Dataset

Dataset	Method	MAE	RMSE	CORR
WC_AO-MS (Random)	SVR	0.0049	0.0062	0.5052
	1D-CNN	0.0039	0.0053	0.5864
	2D-CNN	0.0036	0.0049	0.7353
	Our Approach	**0.0026**	**0.0040**	**0.8020**
WC_AO-MS (Sequential)	SVR	0.0050	0.0061	-0.0463
	1D-CNN	0.0045	0.0057	0.4722
	2D-CNN	0.0043	0.0052	0.4029
	Our Approach	**0.0028**	**0.0036**	**0.7754**

machine started in the morning, (2) when the machine resumed after the lunch break, and (3) when the the cutting tool is adjusted for precision offset. The two workpieces processed after each configuration change were used for model fine-tuning. Subsequent workpieces, processed until the next configuration change or end of the machining, were used for testing. This allowed us to assess whether our fine-tuning approach could adapt to changes in both cutting tools and workpiece materials.

Evaluation Metrics: The performance of our method is evaluated by Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Pearson Correlation (CORR). MAE and RMSE are both used to assess whether a model accurately estimates actual machining errors. RMSE is more sensitive to outliers compared to MAE, and MAE is considered more intuitive to the domain experts.

CORR is used to observe whether the model's estimation of machining error is correlated with the actual ones. Since in certain machining process demanding high precision, the variations of machining errors are small. A model can give machining error estimation in a small value interval to get small MAE and RMSE. However, in such circumstances, if the model is not trully capable of predicting the machining error, the CORR would be low. In other words, CORR assists us in distinguishing whether a model really learns the relationships between the signals during machining process and the machining errors. On the other hand, low CORR with high MAE or RMSE indicates that the estimation made by the model is biased.

Baselines: Three methods were chosen as the baseline methods for comparison.

- SVR: Support vector regression (SVR) is a kernel-based machine learning model for regression tasks [66]. SVR utilizes kernel functions to identify key data points influencing the regression hyperplane and achieve efficient high-dimensional space mapping. In CNC machine applications, SVR is applied for engineering optimization problems such as optimizing surface roughness and cutting forces in milling [67], and controlling the motor current of machine tool drives [68], etc. In this study, referring to the [69], the statistical features of vibration, spindle speed, and motor current signals were processed as input of SVR for model training and machining error inferences.
- 1D-CNN: The one-dimensional convolutional neural network (1D-CNN) is commonly employed for the analysis of time series data. In this study, we adopted the 1D-CNN method proposed by Huang and Lee [30] as a representative for the baseline comparison. In Huang and Lee's

approach [30], 1D-CNN was adopted in conjunction with a sensor fusion technique to accurately estimate tool wear and surface roughness in CNC machining. In this study, the vibration signals were utilized as inputs to the model for the estimation of machining errors.

- 2D-CNN: Once a series of vibration or sound signals is transformed into spectrograms, a visual representation of the spectrum of frequencies of a signal as it varies with time, they can be analyzed using a two-dimensional convolutional neural network (2D-CNN). In this study, the approach delivered by Liao et al. [70] was introduced as a representative of 2D-CNN approaches for the baseline comparison. Liao et al. processed sound signals using Short-Time Fourier Transform (STFT) [55], and transformed the spectrum variations over time into spectrograms to predict specific machining configurations [70]. Liao et al. fine-tuned a VGG16-based model, which was pre-trained on ImageNet [44] to accept spectrograms as input. In this study, we adopted the model framework proposed by Liao et al. [70]. However, instead of sound signals, we transformed the vibration signals into spectrograms using STFT as the model inputs. We then fine-tuned the model accordingly for the estimation of machining errors.

Devices: We pre-trained the core of DeepMachining on a workstation equipped with an AMD Ryzen Threadripper processor 3990X (256M Cache, 2.9 GHz), 256 GB RAM, and an NVIDIA Quadro RTX 8000 (48 GB DDR6 RAM) using TensorFlow. AdamW was employed as the optimizer, with a learning rate of 0.001, a batch size of 512, and 512 epochs. For 2-shot tuning, the core was executed on a host featuring an Intel Xeon Silver 4210 processor (13.75M Cache, 2.2 GHz), 256 GB RAM, and an NVIDIA RTX 2080 Ti (11 GB DDR6 RAM) using TensorFlow. AdamW was again employed as the optimizer, with a learning rate of 0.00001, a batch size of 32, and 64 epochs. The 2-shot tuning time on GPU was 2.5 minutes, with an inference time of 0.026 seconds. The 2-shot tuning time on CPU was 35 minutes, and the inference time was 0.036 seconds.
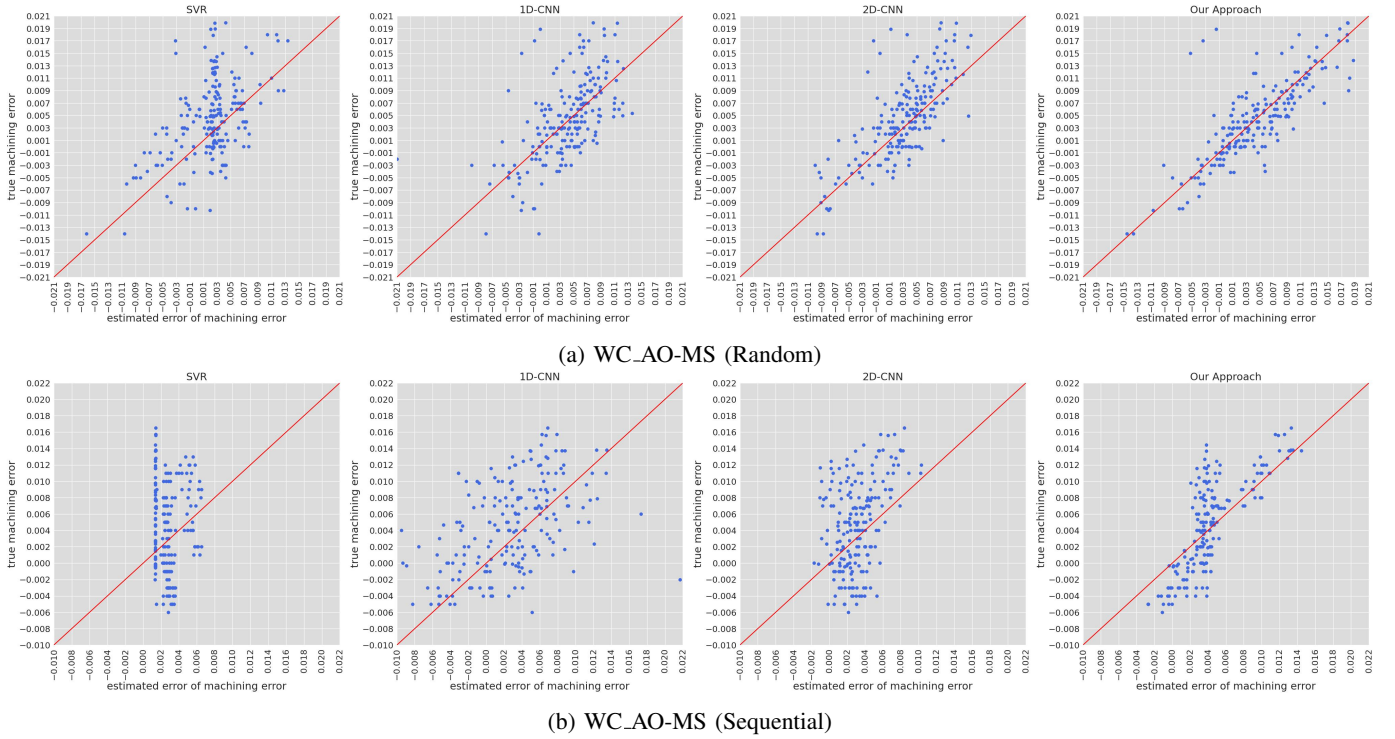
(a) WC_AO-MS (Random)

(b) WC_AO-MS (Sequential)

Fig. 4: Scatter plots for the actual machining errors vs. the ones estimated by different methods on our Pre-trained Dataset.

B. Evaluation and Comparison

Initially, we validated the model's performance on the pre-trained dataset WC_AO-MS as shown in Table II. As demonstrated in Table IV, whether the testing set is generated randomly WC_AO-MS (Random), or sequentially WC_AO-MS (Sequential), our approach surpasses all the baseline methods across various metrics. SVR presents the weakest performance among all the methods, as highlighted by the highest MAE and RMSE, coupled with the lowest CORR Notably, The CORR of SVR is close to 0.5 in WC_AO-MS (Random) but declines significantly to nearly 0 in WC_AO-MS (Sequential). This indicates its limited robustness as machining progresses. 2D-CNN outperforms 1D-CNN's in most metrics, and exhibits inferior in CORR for WC_AO-MS (Sequential). Besides, in comparison to WC_AO-MS (Random), both 1D-CNN and 2D-CNN demonstrate a substantial increase in estimation errors and decrease in CORR on WC_AO-MS (Sequential).

In reality, only first few workpieces processed can be used for model fine-tuning. An approach that cannot perform well with sequential workpieces in production is not practical. In comparison of WC_AO-MS (Random), our approach shows only a slight uptick in estimation errors and a limited reduction in CORR in WC_AO-MS (Sequential). Our results demonstrate that our approach can sustain its predictive performance during continuous machining.

Fig. 4 shows the relationships between the actual machining errors of the testing set (the y-axis) and the estimation of machining errors made by each method (the x-axis). Fig. 4a is for WC_AO-MS (Random), and Fig. 4b is for WC_AO-MS (Sequential). The scales for the plots on the same dataset are equivalent, and the red line in the middle is the identical line

TABLE V: Performance Comparison of Adapted Dataset

Dataset	Method	MAE	RMSE	CORR
WC_TAN-MS	SVR	0.0056	0.0064	0.0319
	1D-CNN	0.0027	0.0034	0.4058
	2D-CNN	0.0016	0.0022	0.7240
	Our Approach	**0.0013**	**0.0016**	**0.8838**
WC_TC-AN	SVR	0.2709	0.3247	0.0339
	1D-CNN	0.0041	0.0052	0.2518
	2D-CNN	0.0029	0.0037	0.6010
	Our Approach	**0.0024**	**0.0032**	**0.7599**

representing a perfect match between the actual values and the corresponding estimations.

Fig. 4b indicates that SVR only estimates machining errors within a limited range of 0.003 to 0.007 and shows no correlation between the estimated values and actual ones on WC_AO-MS (Sequential). Both 1D-CNN and 2D-CNN provide estimates with medium correlation to the actual values in both datasets. From Fig. 4a, we can see that the estimates made by 1D-CNN have higher variance, as evidenced by the dots being more widely distributed around the identical line. However, in Fig. 4a, while 1D-CNN maintains the same tendency in estimation, the estimates made by 2D-CNN seem to be limited by an invisible lower bound around -0.001 mm (the lowest value of actual machining errors in WC_AO-MS (Sequential) is -0.006 mm). As illustrated in Fig. 4, compared to 1D-CNN and 2D-CNN, our approach exhibits smaller estimation errors and better correlation for both the WC_AO-MS (Random) and WC_AO-MS (Sequential) datasets. Additionally, the dots in the plots follow the identical line in a more compact manner.

As Table III shows, WC_TAN-MS is used to show the adapt-

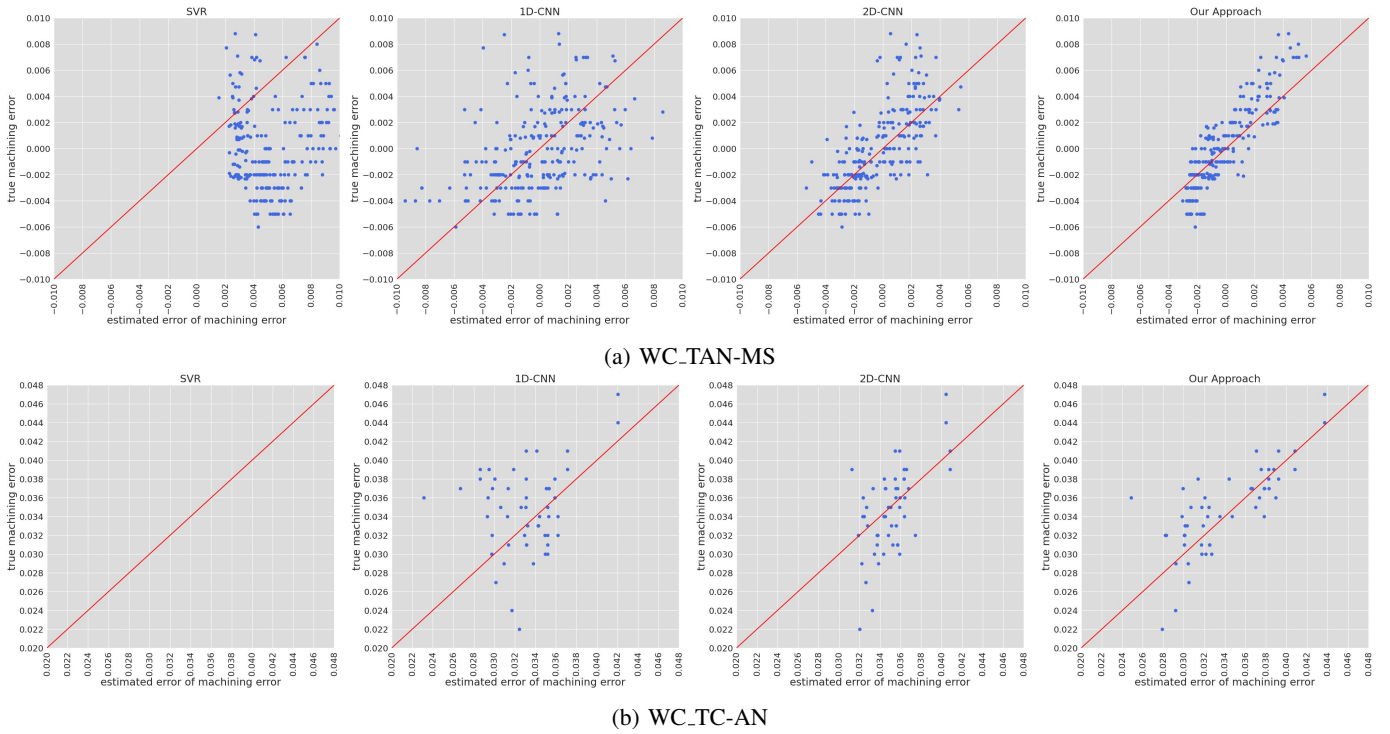(a) WC_TAN-MS

(b) WC_TC-AN

Fig. 5: Scatter plots for the actual machining errors vs. the ones estimated by different methods on our adapted Dataset.

ability for the models in machining with identical material in workpieces but different in tools, and WC_TC-AN is used to demonstrate the adaptability of the models involving different materials in both workpieces and tools. As shown in Table V, The CORR of SVR is close to zero, which reveals that SVR fails to adapt the differences in machining condition (no matter in workpieces or tools) for machining error prediction. On the other hand, 2D-CNN outperforms 1D-CNN in both the datasets. However, it's important to note that our approach surpasses all the other methods, emerging as the best across all the metrics.

From Fig. 5, SVR's estimation shows no correlation to the actual data, and even out of the plotting range with high errors on WC_TC-AN. For the adapted dataset, 1D-CNN and 2D-CNN still perform better than SVR but worse than our approach, and both of them meet an invisible lower bound around 0.03 in estimation on WC_TC-AN. As a result, our approach demonstrates great adaptability and generality while being applied to the datasets generated with different materials in workpieces or tools.

V. DISCUSSION

We have conducted tests on DeepMachining across different types of products in different manufacturing factories. Some lessons learned are addressed below. (1) Sensor installation position: the placement of sensors on CNC machines is crucial. Incorrect positions may lead to ineffective signal reception, resulting in weak signal amplitudes or noise caused by sensor wire pulling. (2) Sensor sampling rate: different sampling rates between pre-trained and fine-tuned stages can impact the model's performance. To ensure the accurate functionality of

DeepMachining, the sampling rate shall be identical at the pre-trained and the fine-tuned stage. (3) Decimal precision: for workpieces with high machining precision, low measurement precision, e.g., measuring a workpiece with a required tolerence of 0.001mm but with a precision of only 0.01mm, it cannot accurately reflect the differences in machining error among workpieces. This lack of precision in data can hinder the model's ability to give precise estimation of machining errors.

In this study, we have only tested the performance of DeepMachining in outer diameter machining tasks on a lathe machine, limiting its application scope to such tasks. However, CNC machining typically encompasses a range of processes, including internal cutting and drilling. Additionally, CNC machines come in various types, such as milling and planing machines. More data from different machining tasks need to be collected and investigated to understand how DeepMachining can be applied to a wider range of machining tasks. In addition to the machining error metric, the workpiece's roughness is another important metric commonly used in CNC machining. However, DeepMachining cannot tackle roughness currently. Addressing this issue by DeepMachining is necessary in the future. In addition to the machining error metric, another important metric commonly used in CNC machining is the workpiece's roughness. Currently, DeepMachining has not yet been applied to handle this aspect. However, it's necessary for DeepMachining to address this issue in the future. Furthermore, applying DeepMachining in learning CNC machining representations for predicting the remaining useful life of cutting tools is also a significant practical research topic. Ultimately, the research goal is to build the entire DeepMachining

system as an open-source intelligent manufacturing platform to have a greater impact on the CNC machinery industry.

VI. Conclusion

This paper proposes DeepMachining, a deep learning-based approach for estimating machining errors for outer diameter processing in horizontal CNC lathe machines. DeepMachining consists of two stages: 1) pre-training a deep learning model and 2) employing few-shot learning (typically two-shot learning) to adapt the pre-trained model to new machining tasks for estimating machining errors.

The core of DeepMachining is pre-trained with machining data from one single tool and workpiece material where the data span from a new tool to completely worn-out, i.e., covering the whole life cycle of a tool. The pre-trained model can be adapted to machining tasks with different tools and workpiece materials by applying the collected data from the first two workpieces in new machining tasks for fine-tuning, also known as two-shot learning, whenever the machining configuration is changed. In summary, based on practical experiments, DeepMachining surpasses all the other baseline methods in terms of estimation accuracy and generality.

Besides, the core of DeepMachining is relatively small with around *260,000* parameters, and only *6.5%* of the parameters need to be fine-tuned with *12.5%* of epochs compared to pre-training for adaptation to machining tasks with different tools, workpieces, and machining configurations. DeepMachining can operate under limited computational resources, while also aligning with the processes of the manufacturing industry using CNC machinery. It is believed that DeepMachining will be a paradigm shift for CNC manufacturers and customers, guided by the principles of deep learning in artificial intelligence and intelligent manufacturing. As manufacturing processes become more complex, more potential relationships exist within manufacturing data among processes, workpieces, and machines. Vast amounts of textual data from machine logs correspond to the quality and yield of workpieces. A large language model (LLM) can excel in a range of natural language processing tasks that understand the machine logs and process recipes. Utilizing LLM in complicated production contexts can help manufacturers suffer less machine downtime and better product quality. Therefore, studying LLM in intelligent manufacturing is a crucial area of future study.
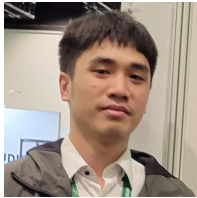
Acknowledgment

References

[1] A. Hamdan, A. A. Sarhan, and M. Hamdi, "An optimization method of the machining parameters in high-speed machining of stainless steel using coated carbide tool for best surface finish," *The International Journal of Advanced Manufacturing Technology*, vol. 58, pp. 81–91, 2012.

[2] S. Mekid and T. Ogedengbe, "A review of machine tool accuracy enhancement through error compensation in serial and parallel kinematic machines," *International Journal of Precision Technology*, vol. 1, no. 3-4, pp. 251–286, 2010.

[3] C.-F. Chien and C.-C. Chen, "Data-driven framework for tool health monitoring and maintenance strategy for smart manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, no. 4, pp. 644–652, 2020.

[4] S. B. Ramezani, L. Cummins, B. Killen, R. Carley, A. Amirlatifi, S. Rahimi, M. Seale, and L. Bian, "Scalability, explainability and performance of data-driven algorithms in predicting the remaining useful life: A comprehensive review," *IEEE Access*, 2023.

[5] T. Benkedjouh, K. Medjaher, N. Zerhouni, and S. Rechak, "Health assessment and life prediction of cutting tools based on support vector regression," *Journal of intelligent manufacturing*, vol. 26, pp. 213–223, 2015.

[6] A. Gavahian and C. K. Mechefske, "Motor current-based degradation modeling for tool wear hybrid prognostics in turning process," *Machines*, vol. 11, no. 8, p. 781, 2023.

[7] Y. Ding, J. Zhuang, P. Ding, and M. Jia, "Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings," *Reliability Engineering & System Safety*, vol. 218, p. 108126, 2022.

[8] M. Fernandes, J. M. Corchado, and G. Marreiros, "Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review," *Applied Intelligence*, vol. 52, no. 12, pp. 14 246–14 280, 2022.

[9] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, 2020.

[10] M. Ntemi, S. Paraschos, A. Karakostas, I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris, "Infrastructure monitoring and quality diagnosis in cnc machining: A review," *CIRP Journal of Manufacturing Science and Technology*, vol. 38, pp. 631–649, 2022.

[11] S. Schwendemann, Z. Amjad, and A. Sikora, "A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines," *Computers in Industry*, vol. 125, p. 103380, 2021.

[12] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects," *Applied Intelligence*, vol. 52, no. 10, pp. 10 934–10 964, 2022.

[13] M. Soori, B. Arezoo, and R. Dastres, "Machine learning and artificial intelligence in cnc machine tools, a review," *Sustainable Manufacturing and Service Economics*, p. 100009, 2023.

[14] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE systems journal*, vol. 13, no. 3, pp. 2213–2227, 2019.

[15] P. Kumar, S. Khalid, and H. S. Kim, "Prognostics and health management of rotating machinery of industrial robot with deep learning applications—a review," *Mathematics*, vol. 11, no. 13, p. 3008, 2023.

[16] V. Nasir and F. Sassani, "A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges," *The International Journal of Advanced Manufacturing Technology*, vol. 115, no. 9-10, pp. 2683–2709, 2021.

[17] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver, "Review of tool condition monitoring in machining and opportunities for deep learning," *The International Journal of Advanced Manufacturing Technology*, vol. 109, pp. 953–974, 2020.

[18] C.-Y. Lee and C.-F. Chien, "Pitfalls and protocols of data science in manufacturing practice," *Journal of Intelligent Manufacturing*, pp. 1–19, 2022.

[19] M. Marei, S. El Zaatari, and W. Li, "Transfer learning enabled convolutional neural networks for estimating health state of cutting tools," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102145, 2021.

[20] M. Marei and W. Li, "Cutting tool prognostics enabled by hybrid cnn-lstm with transfer learning," *The International Journal of Advanced Manufacturing Technology*, vol. 118, no. 3, pp. 817–836, 2022.

[21] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, and X. Chen, "Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing," *IEEE transactions on industrial informatics*, vol. 15, no. 4, pp. 2416–2425, 2018.

[22] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," *arXiv preprint arXiv:2106.10199*, 2021.

[23] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, "A review on soft sensors for monitoring, control, and optimization of industrial processes," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12 868–12 881, 2020.

[24] J. A. Duro, J. A. Padget, C. R. Bowen, H. A. Kim, and A. Nassehi, "Multi-sensor data fusion framework for cnc machining monitoring," *Mechanical systems and signal processing*, vol. 66, pp. 505–520, 2016.

[25] D.-E. Lee, I. Hwang, C. M. Valente, J. F. G. d. Oliveira, and D. A. Dornfeld, *Precision manufacturing process monitoring with acoustic emission*. Springer, 2006.

[26] C. Du, C. L. Ho, and J. Kaminski, "Prediction of product roughness, profile, and roundness using machine learning techniques for a hard turning process," *Advances in Manufacturing*, vol. 9, pp. 206–215, 2021.

[27] B. Denkena, B. Bergmann, and M. Witt, "Material identification based on machine-learning algorithms for hybrid workpieces during cylindrical operations," *Journal of Intelligent Manufacturing*, vol. 30, pp. 2449–2456, 2019.

[28] M. Papananias, T. E. McLeay, O. Obajemu, M. Mahfouf, and V. Kadirkamanathan, "Inspection by exception: A new machine learning-based approach for multistage manufacturing," *Applied Soft Computing*, vol. 97, p. 106787, 2020.

[29] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.

[30] P.-M. Huang and C.-H. Lee, "Estimation of tool wear and surface roughness development using deep learning and sensors fusion," *Sensors*, vol. 21, no. 16, p. 5338, 2021.

[31] D. F. Hesser and B. Markert, "Tool wear monitoring of a retrofitted cnc milling machine using artificial neural networks," *Manufacturing letters*, vol. 19, pp. 1–4, 2019.

[32] A. Proteau, A. Tahan, R. Zemouri, and M. Thomas, "Predicting the quality of a machined workpiece with a variational autoencoder approach," *Journal of Intelligent Manufacturing*, vol. 34, no. 2, pp. 719–737, 2023.

[33] K. Zhu, "Big data oriented smart tool condition monitoring system," in *Smart Machining Systems: Modelling, Monitoring and Informatics*. Springer, 2021, pp. 361–381.

[34] H. Wu, M. J. Triebe, and J. W. Sutherland, "A transformer-based approach for novel fault detection and fault classification/diagnosis in manufacturing: A rotary system application," *Journal of Manufacturing Systems*, vol. 67, pp. 439–452, 2023.

[35] Y. Li, Z. Zhou, C. Sun, X. Chen, and R. Yan, "Variational attention-based interpretable transformer network for rotary machine fault diagnosis," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[36] W. Li, H. Fu, Z. Han, X. Zhang, and H. Jin, "Intelligent tool wear prediction based on informer encoder and stacked bidirectional gated recurrent unit," *Robotics and Computer-Integrated Manufacturing*, vol. 77, p. 102368, 2022.

[37] B. Bhandari, G. Park, and N. Shafiabady, "Implementation of transformer-based deep learning architecture for the development of surface roughness classifier using sound and cutting force signals," *Neural Computing and Applications*, vol. 35, no. 18, pp. 13 275–13 292, 2023.

[38] H. Liu, Z. Liu, W. Jia, X. Lin, and S. Zhang, "A novel transformer-based neural network model for tool wear estimation," *Measurement Science and Technology*, vol. 31, no. 6, p. 065106, 2020.

[39] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[40] T. Zhang, J. Chen, F. Li, K. Zhang, H. Lv, S. He, and E. Xu, "Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions," *ISA transactions*, vol. 119, pp. 152–171, 2022.

[41] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[42] P. Wang and R. X. Gao, "Transfer learning for enhanced machine fault diagnosis in manufacturing," *CIRP Annals*, vol. 69, no. 1, pp. 413–416, 2020.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[45] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 7316–7325, 2019.

[46] A. Bahador, C. Du, H. P. Ng, N. A. Dzulqarnain, and C. L. Ho, "Cost-effective classification of tool wear with transfer learning based on tool vibration for hard turning processes," *Measurement*, vol. 201, p. 111701, 2022.

[47] N. S. Ross, P. T. Sheeba, C. S. Shibi, M. K. Gupta, M. E. Korkmaz, and V. S. Sharma, "A novel approach of tool condition monitoring in sustainable machining of ni alloy with transfer learning models," *Journal of Intelligent Manufacturing*, pp. 1–19, 2023.

[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[49] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and A. Askell, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[50] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[51] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[52] H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce activations, not trainable parameters for efficient on-device learning," *arXiv preprint arXiv:2007.11622*, 2020.

[53] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[54] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," *arXiv preprint arXiv:2110.04366*, 2021.

[55] R. N. Bracewell and R. N. Bracewell, *The Fourier transform and its applications*. McGraw-Hill New York, 1986, vol. 31999.

[56] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[57] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[59] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[60] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[61] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets: Time-Frequency Methods and Phase Space Proceedings of the International Conference, Marseille, France, December 14–18, 1987*. Springer, 1990, pp. 286–297.

[62] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[63] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[65] H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce activations, not trainable parameters for efficient on-device learning," *arXiv preprint arXiv:2007.11622*, 2020.

[66] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[67] A. Yeganefar, S. A. Niknam, and R. Asadi, "The use of support vector machine, neural network, and regression analysis to predict and optimize surface roughness and cutting forces in milling," *The International Journal of Advanced Manufacturing Technology*, vol. 105, pp. 951–965, 2019.

[68] M. Schwenzer, T. Auerbach, K. Miura, B. Döbbeler, and T. Bergs, "Support vector regression to correct motor current of machine tool drives," *Journal of Intelligent Manufacturing*, vol. 31, pp. 553–560, 2020.

[69] S. Sayyad, S. Kumar, A. Bongale, K. Kotecha, G. Selvachandran, and P. N. Suganthan, "Tool wear prediction using long short-term memory variants and hybrid feature selection techniques," *The International Journal of Advanced Manufacturing Technology*, vol. 121, no. 9-10, pp. 6611–6633, 2022.

[70] Y. Liao, I. Ragai, Z. Huang, and S. Kerner, "Manufacturing process monitoring using time-frequency representation and transfer learning of deep neural networks," *Journal of Manufacturing Processes*, vol. 68, pp. 231–248, 2021.

H. T. Kung is William H. Gates Professor of Computer Science and Electrical Engineering at Harvard University. He conducts research on topics related to the application of artificial intelligence in manufacturing and healthcare, AI accelerators, VLSI design, high-performance computing, parallel and distributed computing, computer architectures, and computer networks.

Professor Kung received his bachelor's degree from National Tsing Hua University in Taiwan in 1968. He received his Ph.D. degree from Carnegie Mellon University in 1973. He taught at Carnegie Mellon for 19 years before joining Harvard in 1992.

Professor Kung is an elected member of the US National Academy of Engineering for introducing the idea of systolic computation, contributions to parallel computing, and applying complexity analysis to very-large-scale-integrated (VLSI) computation. In addition, Professor Kung is an ACM Fellow, Guggenheim Fellow, an elected member of the Academia Sinica in Taiwan, as well as the president and co-founder of the Taiwan AI Academy, a non-profit organization which has cultivated over 10,000 AI talents for industries since 2019.

Xiang-Li Lu received his B.E. and M.S. degrees in Information Engineering and Computer Science from Feng Chia University, Taiwan, in 2020 and 2024. He is currently serving as a research assistant of professor Hwai-Jung Hsu in the Department of Information Engineering and Computer Science at Feng Chia University. His research interests include graph neural networks, speech analysis, computer vision, and deep learning.

Chen-Hsin Lee received his Duo-Bachelor's Degree in Business Management, with a major in Business Information Systems, in 2023 from The University of Queensland, Australia, and Feng Chia University, Taiwan. He is currently pursuing a Master's Degree in Information Technology, specializing in Data Analytics at The University of Technology Sydney, Australia.

Hwai-Jung Hsu received his B.E., M.S., and Ph.D. degrees in Computer Science from National Chiao Tung University, Taiwan in 2001, 2003, and 2011. He is now the director of the artificial intelligence reasearch center and an associate professor in the Department of Information Engineering and Computer Science at Feng Chia University. His research interests include data analytics , computer vision and deep learning, agile development,and psychophysiology.

Sheng-Mao Cheng received two M.S. Degree in Physics and Industrial Engineering and Engineering Management from National Tsing Hua University, Taiwan, in 2011 and 2022. He worked as a process engineer at TSMC for 3 years from 2012 to 2015. From 2015 he switched to the precise machine industry. Currently, he is the manager of Victor Taichung Machinery Company. He has participated in many important major projects of Victor Taichung, such as production line improvement, information system integration, and new generation products and application technology development.

Che-Wei Chou is an assistant professor at the Department of Industrial Engineering and Systems Management at Feng Chia University. He received a Ph.D. degree in Industrial Engineering and Engineering Management from National Tsing Hua University, Taiwan, in 2016. He also received a B.S. degree and M.S. in physics from Industrial Engineering Management from Yuan Ze University, Taiwan, in 2001 and 2003, respectively. He served as the CEO of Artificial Intelligence for Intelligent Manufacturing Systems (AIMS) Research Cen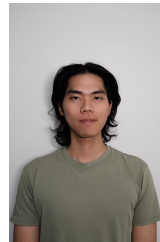ter, MOST, Taiwan from 2019 to 2021. His research interests include intelligent manufacturing, manufacturing large language models, deep learning, decision analysis, and big data analytics.