



Monitoring the misalignment of machine tools with autoencoders after they are trained with transfer learning data

Mustafa Demetgul¹ · Qi Zheng¹ · Ibrahim Nur Tansel² · Jürgen Fleischer¹

Received: 16 April 2023 / Accepted: 24 July 2023 / Published online: 19 August 2023
© The Author(s) 2023

Abstract

CNC machines have revolutionized manufacturing by enabling high-quality and high-productivity production. Monitoring the condition of these machines during production would reduce maintenance cost and avoid manufacturing defective parts. Misalignment of the linear tables in CNCs can directly affect the quality of the manufactured parts, and the components of the linear tables wear out over time due to the heavy and fluctuating loads. To address these challenges, an intelligent monitoring system was developed to identify normal operation and misalignments. Since damaging a CNC machine for data collection is too expensive, transfer learning was used in two steps. First, a specially designed experimental feed axis test platform (FATP) was used to sample the current signal at normal and five levels of left-side misalignment conditions ranging from 0.05 to 0.25 mm. Four different algorithm combinations were trained to detect misalignments. These combinations included a 1D convolution neural network (CNN) and autoencoder (AE) combination, a temporal convolutional network (TCN) and AE combination, a long short-term memory neural network (LSTM) and AE combination, and a CNN, LSTM, and AE combination. At the second step, Wasserstein deep convolutional generative adversarial network (W-DCGAN) was used to generate data by integrating the observed characteristics of the FATP at different misalignment levels and collected limited data from the actual CNC machines. To evaluate the similarity and limited diversity of generated and real signals, t-distributed stochastic neighbor embedding (T-SNE) method was used. The hyperparameters of the model were optimized by random and grid search. The CNN, LSTM, and AE combination demonstrated the best performance, which provides a practical way to detect misalignments without stopping production or cluttering the work area with sensors. The proposed intelligent monitoring system can detect misalignments of the linear tables of CNCs, thus enhancing the quality of manufactured parts and reducing production costs.

Keywords CNC linear feed axis · Convolution neural network (CNN) · Generative adversarial network (GAN) · Long short-term memory neural network (LSTM) · Autoencoder (AE) · Transfer learning (TL)

1 Introduction

Modern factories use many different machine tools for various manufacturing processes. This diversity makes monitoring and maintaining machines extremely challenging, so many traditional methods of fault detection are no

longer adequate. In recent years, many new technologies have emerged in the field of machine control and artificial intelligence (AI), enhancing the intelligence of the production process, and people also have demands for improved production efficiency, which has significantly promoted the deployment of advanced and fast error detection systems in the manufacturing industry. With suitable sensors, you can track parameters such as pressure, flow, temperature, vibration, and more.

In large machines and complex equipment, real-time health monitoring and error detection are necessary—for example, there are many works focused on detecting the faults of various components in rotating machinery, such as component failures in bearing test rigs [1] and component failures in rolling bearings [2–4], collapse of rotating

✉ Mustafa Demetgul
mdemetgul179@gmail.com

¹ Institute of Production Science, Karlsruhe Institute of Technology, Rinheimer Querallee 2, 76131 Karlsruhe, Germany

² Mechanical and Materials Engineering Department, Florida International University, 10555 West Flagler Street EC, Miami, FL 3400, USA

mechanical parts [5, 6], and many others [7, 8]. The examples above demonstrate that there have been numerous efforts and research studies aimed at mitigating error issues in machine production processes. However, recent studies have paid limited attention to detecting misalignment in the linear feed axis. In many large machines, the linear feed axis is a crucial component of the feed system, responsible for transferring the workpiece to the processing station for further production [9]. Therefore, any inaccuracies in the linear feed axis can directly impact production efficiency and the quality of the manufactured workpiece [10, 11].

Various anomaly detection techniques have been proposed in literature, including those based on feature selection [12–14], signal similarity [15], probability distribution [16], and forecasting [17].

To detect equipment failures, many traditional time series signal anomaly detection methods are introduced in [18]. A machine anomaly detection method based on analyzing acoustic signals is summarized in [19]. Demegul et al. [20] also explored the possibility of using the short-time Fourier transform (STFFT) to diagnose the misalignment problem of the linear feed axis. Anomalies can be very well identified by detecting pattern changes through machine learning and deep learning algorithms. Additionally, direct encoding of motor current time series data into images enables the classification of linear feed axis signals through the residual neural network (ResNet) with hyperparameter optimization performance of 99.76% [21]. However, installing sensors for fault detection and monitoring incurs additional costs and may complicate the overall control system, particularly with environmental conditions and noise affecting sensor accuracy. Fortunately, each linear axis motor is equipped with a PLC control system that can directly monitor the motor's current value in real-time, without requiring additional sensor installation. This provides a convenient and cost-effective means for detecting errors.

Given that the current variation of a motor can be read in real-time through a PLC control system, the primary objective of this paper is to leverage this information to detect the machine's operating condition. Time series data of the current value can be acquired using this method with a given sampling rate. The data obtained from this process has different temporal and sequential characteristics, which are dependent on the specific working conditions of the machine. Hence, it is crucial to develop suitable anomaly detection methods that are based on time series data to effectively detect errors.

The success of existing deep learning-based fault diagnosis methods is heavily dependent on a vast number of labeled samples, where feature extraction and selection are performed manually. However, collecting labeled data for complex equipment in the actual use environment is challenging, leading to the deep learning network relying

on extensive data learning not being able to achieve the expected performance. Additionally, the existing industrial dataset is seriously unbalanced, with anomaly data accounting for only 5–10% of the total data volume, rendering traditional dropout and batch normalization ineffective [22]. Consequently, data collection and training pose significant challenges to the implementation of fault diagnosis in intelligent industrial systems.

Our research investigated the feasibility of utilizing motor current time series to monitor a linear feed axis under varying operating conditions. However, to avoid interruption of the operation for hours and damaging the machine tool, a limited number of experiments were conducted, and the available data did not suffice for model training and effect validation, as previously stated. To address the challenges of obtaining anomaly signals under diverse conditions and the scarcity of available samples, we employed generative adversarial networks (GANs) to generate synthetic data that closely resembles actual data. For example, Jiang et al. [23] used a deep convolutional generative adversarial network (DCGAN) to generate rolling bearing time series data. Also, based on a deep convolutional generative adversarial network (DCGAN), Sabir et al. [24] applied it to construct a current signal dataset to achieve motor open-circuit fault detection. For generating vibration data, Luleci et al. [25] improved deep convolutional generative adversarial networks (DCGANs) and used Wasserstein distance to generate synthetic data for structural damage diagnosis. The authors named the developed model W-DCGAN and successfully generated the current data at normal and anomaly conditions. Many different GAN models were studied and compared, and the improved W-DCGAN model was selected for dataset generation because of its outstanding performance.

After augmented the data, to qualitatively measure and contrast the similarity between the generated and real signals, t-distributed stochastic neighbor embedding (t-SNE) [26] analyses were performed to visualize the similarity of the generated distribution to the original distribution. At the same time, the ability of various classifiers to distinguish between real data and generated data can be quantitatively tested. For example, by training the classifier on synthetic data [27, 28], through this method, the degree of similarity between the generated data and the original data can be fully evaluated. The W-DCGAN model is shown to be on par with state-of-the-art models in generating realistic time series.

In recent years, deep learning models have been widely used in anomaly detection due to their ability to automatically learn representative invariant features and complex nonlinear relationships from raw data [29]. However, the relationship between signals of different working conditions often involves complex mapping, leading to a reduced generalization effect. This stands in contrast to traditional intelligent fault diagnosis methods.

Deep learning methods perform well in image processing, natural language processing, etc. The most common deep learning methods are convolutional neural networks (CNNs) [29] and recurrent neural networks (RNNs) [30]. There are various implementations of RNN memory units, the most famous of which are long short-term memory (LSTM) [31, 32].

Numerous unsupervised fault detection methods based on autoencoders have been researched and developed to effectively detect faults in motor current time series datasets. These methods include CNN-based autoencoders [33], LSTM-based autoencoders [34, 35], and hybrid autoencoders [36, 37] that combine CNN and LSTM for extracting spatial features and modeling temporal information. These methods first divide the time series data into positive marks and negative marks. The positive mark data is the normal state of the equipment, and the negative mark is the fault status. The autoencoder is trained on positively labeled data, while negatively labeled data are identified as anomalies based on higher reconstruction error compared to positively labeled data. Malhotra et al. first proposed to apply an LSTM-based autoencoder to anomaly detection in time series [38]. Nogas et al. [39] used the convolution layer and ConvLSTM for spatiotemporal encoding and decoding. The proposed ConvLSTM-AE model effectively integrates the improved ConvLSTM and CNN-AE for successful feature extraction and fault detection from complex process signals. Kim and Cho proposed a C-LSTM model for time series anomaly detection and this model consists of CNN, LSTM, and Deep Neural Network (DNN) [40]. Khan et al. [36] also proposed a similar hybrid model of C-LSTM-AE synergy. The CNN layer is also used to extract spatial features, and its output is fed to LSTM-AE, followed by dense (fully connected) layers for final prediction.

Many researchers studied anomaly detection by evaluating the time-domain signals. Instead of converting the original 1D motor current signals into 2D images, it is also possible to directly detect the anomalies from the raw signal. For example, Li et al. developed an anomaly detection method for tool wear based on CNC machine tool motor current data. They used a CNN-based autoencoder model and the model has strong generalization ability [41]. In addition, Yang et al. proposed a method for detecting the abnormal operating status of linear motor feed systems based on motor current. They first used the long short-term memory (LSTM) network to extract the time series features of the current [42].

Transfer learning has become a research focus in the field of fault detection. For practical applications, training datasets distributions (source domain) and test datasets (target domain) are often different because they are derived from different working conditions or various devices. In this case, it is difficult for a deep learning model trained in a

single scene to achieve good recognition results in all the scenarios [43]. To overcome this problem, transfer learning is proposed for fault diagnosis. Transfer learning methods can effectively reduce the distribution difference between source and target domains and extract common features. Li et al. have reviewed deep transfer learning for mechanical fault diagnosis in recent years [43]. Mao et al. constructed a three-channel dataset with time/frequency/time-frequency domain information. Then based on the pre-trained VGG-16 model and support vector machine, effective online detection of early bearing failures is realized [44]. Wan et al. proposed a gear fault diagnosis method based on transfer learning. This method uses a cohesive evaluation method to select sensitive features [45]. Shao et al. achieved fast and accurate machine fault diagnosis using transfer learning [46].

Although the aim of this paper is to detect misalignment of linear feed axis in various scenarios, the complex and multifaceted nature of actual working conditions makes it impossible to gather data and train models for every operational state. Furthermore, as anomalies cannot be simulated in CNC machines without risking damage to the equipment, alternative methods of generating data must be considered.

Our approach involves gathering data on misalignment from a single experimental setup, using this data to train algorithms and construct models. We can then apply these pre-trained models to other scenarios. To adapt the experimental anomaly data to the normal state data of the CNC linear feed axis, we must make some adjustments since it is extremely costly to simulate anomalies on the CNC machine table. This method allows for a short training time for the models to reach convergence, and it has been shown to have a highly effective diagnostic effect.

In this study, normal data, and anomaly data of varying degrees of misalignment were collected from a constructed feed axis test platform (FATP). Subsequently, normal condition CNC signals were collected from a real CNC machine, given a particular speed and stroke. However, the inability to generate anomaly signals on the CNC machine without damaging the equipment resulted in the synthetically generated anomaly CNC signals. The difference between normal data and different levels of anomaly data collected on the linear feed axis experimental platform were added to normal condition CNC data to create these synthetic anomaly CNC signals. Furthermore, data augmentation was conducted through the W-DCGAN model. Finally, pre-trained autoencoder models on the linear feed axis data were employed to detect anomalies from CNC data using transfer learning.

The study makes the following key contributions:

Introduction of a hardware twin for data collection: The required data was obtained through a few tests using a CNC machine during normal operating conditions. To generate extensive training data, FATP [20] was utilized under both normal and desired anomaly conditions.

Synthetic data generation using an AI tool called W-DCGAN: W-DCGAN was employed to generate extensive synthetic data using the experimental data from the CNC machine and its twin FATP.

Utilization of t-SNE for evaluating the significance of the characteristics of the entire data (experimental and synthetic): CNC controller signals were studied for the first time at the misalignment condition.

Using and comparing performances of multiple AI tool combinations: Comparison of the performance of different combinations, namely CNN-AE, TCN-AE, LSTM-AE, and CNN-LSTM-AE for identification of misalignment.

Introduction of a low cost and convenient misalignment detection tool: Various geometric error measurement tools and methods [47–49] exist to assess the condition of machine tools, including misalignments. These testing methods require the installation of specialized instruments on the CNC machines and involve conducting multiple tests after the machine has been taken out of production. In contrast, the proposed method, consisting of above four steps, offers significant advantages as it eliminates the need for additional equipment installation and only requires monitoring the signals at the controller of the CNC machine after the AI tools have been trained.

2 Theoretical background

In this study, 4 neural networks were used. These neural networks were selected based on the gained student experience in last 4 years. In this decision, availability, training speed, performance, convenience, and compatibility were the main factors.

2.1 1D CNN

Convolution is a special kind of linear operation. A neural network that uses convolution operations instead of normal matrix multiplication operations is called a convolutional neural network. 2D CNN is generally used in image recognition to extract features from images. The classic CNN models are LeNet, ResNet, etc. Since our current signal input is one-dimensional, a 1D CNN is used to extract the relevant features of the input current signal. Hence, its convolutional kernel also adopts a one-dimensional structure. Each commonly used convolutional network usually consists of three types of layers: convolutional layer, activation layer, and pooling layer. The output of each layer also corresponds to a one-dimensional feature vector [11].

The basic structure of the one-dimensional convolution unit is shown in Fig. 1, and its core components are described below.

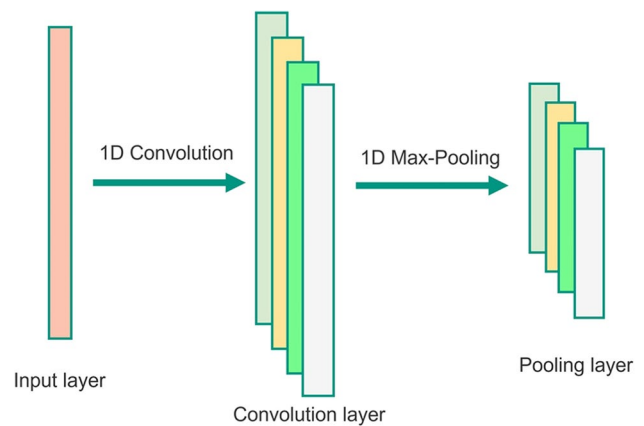


Fig. 1 Structure of the 1D CNN unit

- 1 Convolutional layer: It is the main building block of CNN. It consists of a set of filters (or kernels) whose parameters are learned throughout training, and input samples are convolved with multiple parameter-learnable filters, respectively. The convolution operation can be expressed as follows (equation 1).

$$x_i^k(j) = w_i^k * x_{i-1}(j) + b_i^k \quad (1)$$

where $*$ denotes the convolution operator, i is the index of the number of network layers, k is the index of the output feature maps, j represents the index of current region to participate in the convolution, $x_{i-1}(j)$ is the input feature map, and $x_i^k(j)$ is the k -th output feature map after the convolution calculation is completed. w_i^k and b_i^k are the weight and the bias.

- 2 Activation function: The activation function is a nonlinear function that performs nonlinear transformation on the input signal to enhance the ability of the convolutional neural network to extract feature vectors. Common activation functions are Sigmoid, Tanh, and ReLU. Among them, ReLU is widely used because of its fast convergence speed and its ability to overcome gradient vanishing. The following formula (Eq. 2) can describe the ReLU activation function.

$$A_i^k(j) = f(x_i^k(j)) = \max\{0, x_i^k(j)\} \quad (2)$$

where $x_i^k(j)$ is the output value of the convolution operation, $A_i^k(j)$ is the output value of $x_i^k(j)$ after the activation function.

- 3 Pooling layer: A pooling operation is adopted to reduce the computational complexity and effectively control the risk of overfitting. The calculation process of max pooling is given by equation 3.

$$P_i^k(j) = \max_{(j-1)S+1 < t \leq jS} \{A_i^k(t)\} \tag{3}$$

where $A_i^k(t)$ represents the value of t th neuron in the k th output feature map of the i th layer. S represents the pooling factor size, and P_i^k is the output feature map after max pooling.

2.2 LSTM

Recurrent neural networks (RNNs) are a popular deep learning architecture. In traditional neural network models, nodes between adjacent layers are interconnected. But they are not connected in the same layer. However, the correlation between the data is strong for time series data. Therefore, the traditional neural network structure cannot effectively learn the features of time series data because the nodes in each layer are not connected. The biggest difference between RNN and the traditional fully connected model is that the nodes of each layer are also interconnected, and the connections between units and the sequence information from the input form a directed graph. The biggest problem of RNN is that the gradient disappears, making it easy to forget the previous information [38]. So as an enhanced version of RNN, LSTM overcomes the vanishing gradient problem through gates (input, forget, and output) and storage units. It uses storage units capable of representing long-term dependencies in sequence data. Compared to RNNs, LSTMs exhibit a more complex independent module structure, with more tunable parameters in the threshold unit.

As shown in Fig. 2, the LSTM memory cell comprises four gates, namely the input gate, output gate, forget gate, and self-recurrent neuron. These gates regulate the interaction between memory cells, ensuring the effective storage and retrieval of relevant information. The functioning of the gate structure and specific calculation processes are elaborated below [39].

- 1 Input gate: The input gate computes the cell state \tilde{C}_t to input based on $[h_{t-1}, x_t]$, and computes the vector i_t to control what information to be input into the cell state

\tilde{C}_t . The control function of the input gate is given by equations 4 and 5.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

where \tilde{C}_t , W_c , and b_c represent the cell state value to be input from the new input, the updated weight of the cell state, and the cell state bias value, respectively. Tanh is the hyperbolic tangent activation function, and i_t , W , and b_i are the output vector, weight matrix, and bias value of the input gate, respectively.

- 2 Forget gate: The vector f_t represents what information should be forgotten from the cell state C_{t-1} at time $t-1$. Which information needs to be forgotten is controlled by Eq. 6.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{6}$$

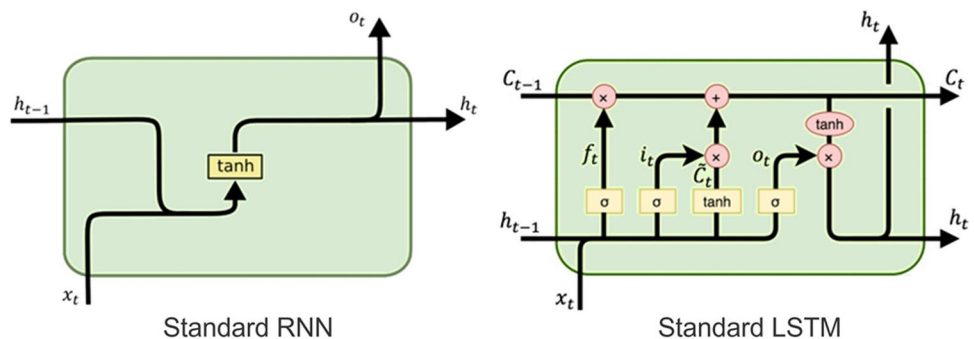
where σ is the sigmoid function, W_f is the weight vector of the forget gate, and b_f is the offset value of the forget gate. $[h_{t-1}, x_t]$ is the concatenation of hidden layer h_{t-1} and input x_t

- 3 Cell state: After calculating the forget gate f_t , the information to be forgotten C_{t-1} is determined. Subsequently, the cell state C_t is then derived from the data at time t . By combining this with the input gate i_t calculation results, the information that can be input is determined. The control function of the cell state C_t at time t is represented in Eq. 7.

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{7}$$

- 4 Output gate: The final output of the LSTM model is the output vector of the output gate o_t . At the same time, the cell state C_t at time t determines which information in the output model will be output as the hidden state h_t . The control function of the output gate is equation 8 and 9.

Fig. 2 Standard architecture of RNN and LSTM



$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = o_t \times \tanh(C_t) \quad (9)$$

where o_t , W_o , and b_o are the output vector, weight vector, and offsets of the output gate. h_t is the output hidden state vector of the LSTM model at time t .

2.3 GAN

The GAN framework comprises two adversarial neural networks that compete against each other. The first network, known as the discriminator, is trained to differentiate between real and fake input data, assessing the authenticity of the generated data. The second network, the generator, receives random noise $z \in R^r$ as input and attempts to produce synthetic data that closely resembles the training data distribution, thereby evading detection as false by the discriminator. Consequently, the generator's objective is to maximize the discriminator's error rate, while the discriminator strives to minimize it [48]. The figure below shows a simple example of a game played by a GAN architecture and a neural network model, with the discriminator and generator labeled D and G , respectively. This leads to a two-player minimax game defined by the value function $V(D, G)$. The input of D is either the generated data $G(z)$ or the real data x . In the training process, D maximizes the cost function while correctly identifying between real and fake data. Meanwhile, the training process of G generates more and more realistic data closer to the real samples by minimizing the cost function. The training is brought to a stop when Nash equilibrium is reached, and D and G cannot be further improved by way of further training [49, 50] (Eq. 10).

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}} [\log D(x)] - E_{z \sim p_z} [\log (1 - D(G(z)))] \quad (10)$$

The architecture of the proposed generator and the discriminator model of GAN is shown in Fig. 3. This architecture is an improved version of a one-dimensional deep convolutional generative adversarial network based on Wasserstein distance. The signal amplitude is normalized between 0 and 1 before feeding it to the 1d W-DCGAN for training.

In the model, the generator takes a Gaussian noise tensor (z) of dimension $[1 \times 100]$ and convolves it through 5 one-dimensional transposes, then it creates a generated current signal of $[1 \times 320]$. A ReLU activation function and a batch normalization are used for each layer, except for the last layer, which uses a Sigmoid activation function. Both the generated signal from the generator and the one-dimensional input data within the same range are fed to the discriminator, referred to as the critic in the W-GAN paper. The critic processes the input 1D tensor through five transposed 1D convolutions, generating a decision score for subsequent backpropagation and network optimization. LeakyReLU activation functions and batch normalization are employed in each layer of the critic, except for the final layer, which omits an activation function. Notably, the generator and critic exhibit inverse structures, utilizing the same filter, stride, and padding values in contrasting manners. More details on the computation process and more information on the original model can be found in the GAN [51], DCGAN [52], and W-GAN [53] papers.

2.4 Autoencoder

An autoencoder is an unsupervised artificial neural network composed of three core layers arranged sequentially: an

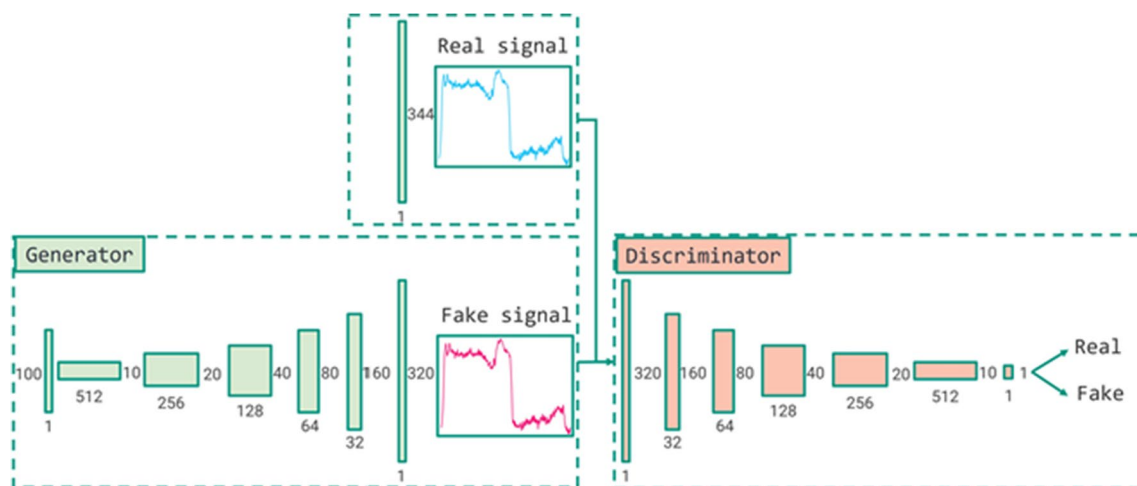


Fig. 3 Architecture of the generative adversarial network

input layer, a hidden layer, and an output (or reconstruction) layer. The primary objective of an autoencoder is to perform dimensionality reduction and learn a compact representation of the dataset. As shown in Fig. 4, The training process of AE consists of two stages: In the encoding stage, the model learns a compressed representation (or latent variable) of the input. Instead, in the decoding stage, the model reconstructs the object from the compressed representation in the encoding stage [54].

Autoencoders are trained to minimize reconstruction loss, which is usually given by Eq. 11.

$$\text{Reconstruction loss} = \|x - \hat{x}\|^2 \tag{11}$$

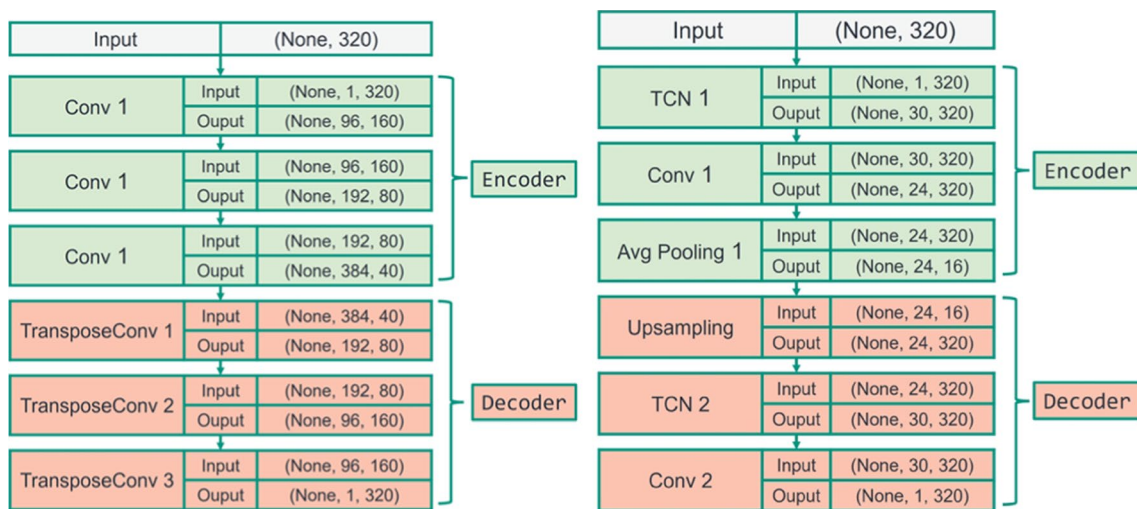
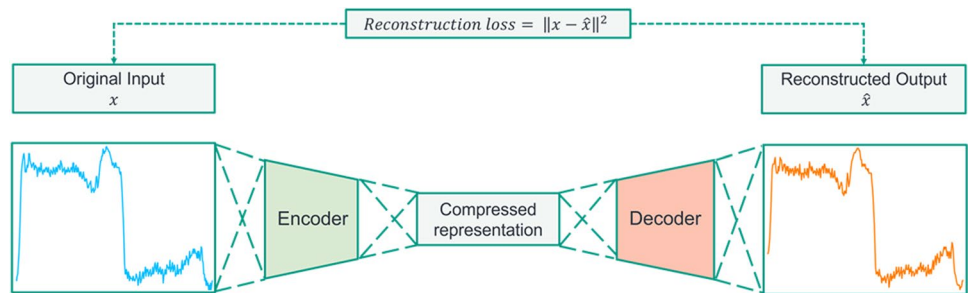
2.5 Ensemble structures

The first model to be used was CNN-AE, the model structure is shown in Fig. 5a. It includes six convolutional units. Each convolutional unit in the encoder consists of a ReLU activation function, MaxPoolin1D, and Batch normalization. Each

convolutional unit in the decoder consists of a ReLU activation function, UpSampling1D, and Batch normalization. A 20% dropout was applied to the output of each convolutional unit to prevent overfitting. The kernel size is both 3, the MaxPoolin1D and UpSampling1D size is 2, and the stride of all layers is set to 1. The Adam optimizer and mean squared error loss function are used in the model. The model results are presented in the next sections. A TCN can be described by four elements: a list of dilation rates ($q_1, q_2, q_3, \dots, q_n$), the number of filters (nfilters), the kernel size (k), and the number of convolutional layers, where for each layer the number of filters is the same as the kernel size. The TCN unit used in this thesis is four stacked convolutional layers, the number of filters (nfilters) is 30, the convolution kernel size (k) is 30, and each layer sequentially uses a dilation rate list (1, 2, 4, 8) to achieve dilated convolution.

The architecture used in this section is a TCN-based autoencoder called TCN-AE [55], as shown in Fig. 5b. Like other autoencoders, TCN-AE also consists of an encoder and a decoder. The encoder initially uses a TCN unit to process an input sequence of length 320 and dimension 1. Then

Fig.4 Architecture of autoencoder



a) Architecture of the proposed CNN-AE model b) Architecture of the proposed TCN-AE model

Fig.5 Different ensemble deep learning structures

the input sequence is input into a TCN unit, and the output sequence length is still 320, but the dimension is 30 according to the number of TCN filters. Subsequently, since the input sequence is not particularly complex, a 1D convolution ($q = 1, k = 1, \text{nfilters} = 24$) is used to reduce the dimension size of the feature map output by the TCN to 24. Then for each feature map, an average pooling layer with a down sampling factor size of 16 is used to down sample the sequence. The extracted compressed representation has a dimension of 24 and a length of 16. This down sampled compressed representation is then fed into the decoder, which initially restores its original size using an up-sampling layer based on nearest-neighbor interpolation. Subsequently, the data passes through a second TCN with identical parameters to the first TCN. Finally, the input sequence is reconstructed using a 1D convolutional layer. In this model, the Adam optimizer and mean squared error loss function are employed for training. Detailed results are presented in the next sections.

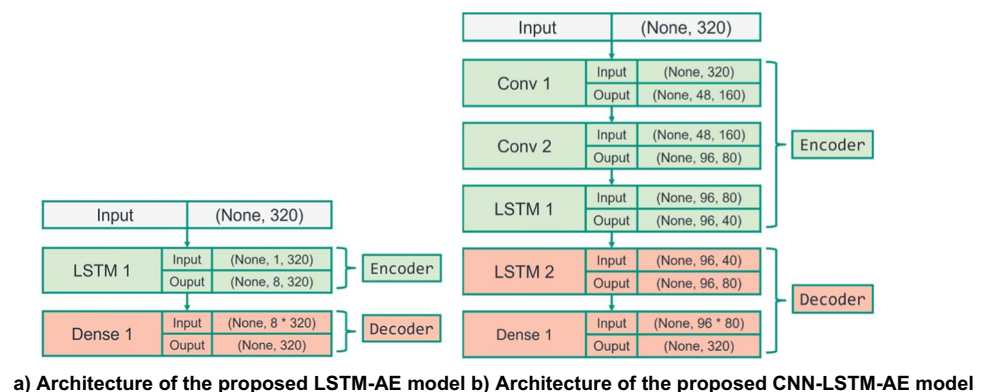
LSTM-based autoencoders are widely used to learn dependencies between time series. The simple LSTM-AE model is applied as shown in Fig. 6a. The first LSTM layer acts as the encoder, and the dropout layer is used to prevent overfitting. The input sequence is fed directly to the encoder. Without using an additional LSTM layer for decoding, the encoded sequence is input to a fully connected layer for decoding, which produces an output prediction for the input sequence. The Adam optimizer and mean squared error loss function are used in the model. This simple model works well, and detailed results are presented in the next sections. In this work, a hybrid network to detect anomalies in the input current time series is developed. LSTM can learn from the temporal dependencies of one sequence and another but needs help modeling spatial features. Therefore, in this work, CNN is used to extract spatial features from the input dataset and then feed these features to LSTM-AE. The structure is shown in Fig. 6b. The first is two one-dimensional CNN layers used to extract spatial features; each CNN unit includes a convolutional layer, dropout layer, pooling layer, ReLU layer, and batch normalization. The filter size of the first

convolutional layer is 48, while the filter size of the second layer is 96, and the kernel size of both convolutional layers is 3. The extracted features are then fed into an LSTM layer as an encoder, which encodes the input sequence for 96-time steps, these encoded sequences are fed into another LSTM layer for decoding, and finally, a dense layer is used to reconstruct the input sequence. The Adam optimizer and mean squared error loss function are used in the model. The proposed hybrid model works well and achieves convincing results, which will be evaluated in the next sections.

2.6 t-SNE

t-SNE (t-distributed stochastic neighbor embedding) is a nonlinear dimensionality reduction machine learning algorithm, particularly well-suited for visualizing high-dimensional data in 2D or 3D spaces. It helps ascertain whether high-dimensional data is separable, characterized by small intra-class distances and large inter-class distances. T-SNE nonlinearly projects high-dimensional data into 2D, or 3D space. If distinct separations are observed in the low-dimensional space, it suggests the high-dimensional data is separable. Conversely, if separations are not apparent, the data may be inseparable in high-dimensional space or unsuitable for low-dimensional projection for classification purposes. T-SNE's principle involves converting data point similarity into conditional probabilities. In the original space, data point similarity is represented by Gaussian joint distribution, while in the embedded space, it is represented by Student's t-distribution. Embedding quality is assessed using the KL divergence of the joint probability distributions in both the original and embedded spaces, which serves as a measure to evaluate the similarity between the two distributions. In essence, a function related to the KL divergence is employed as the loss function, and this loss function is minimized using the gradient descent algorithm, ultimately yielding the convergent result. t-SNE can be considered one of the most effective methods for data dimensionality reduction

Fig. 6 Different ensemble deep learning structure



and visualization. Its primary drawback is its higher memory consumption and longer execution time [26].

3 Experimental setups and data processing

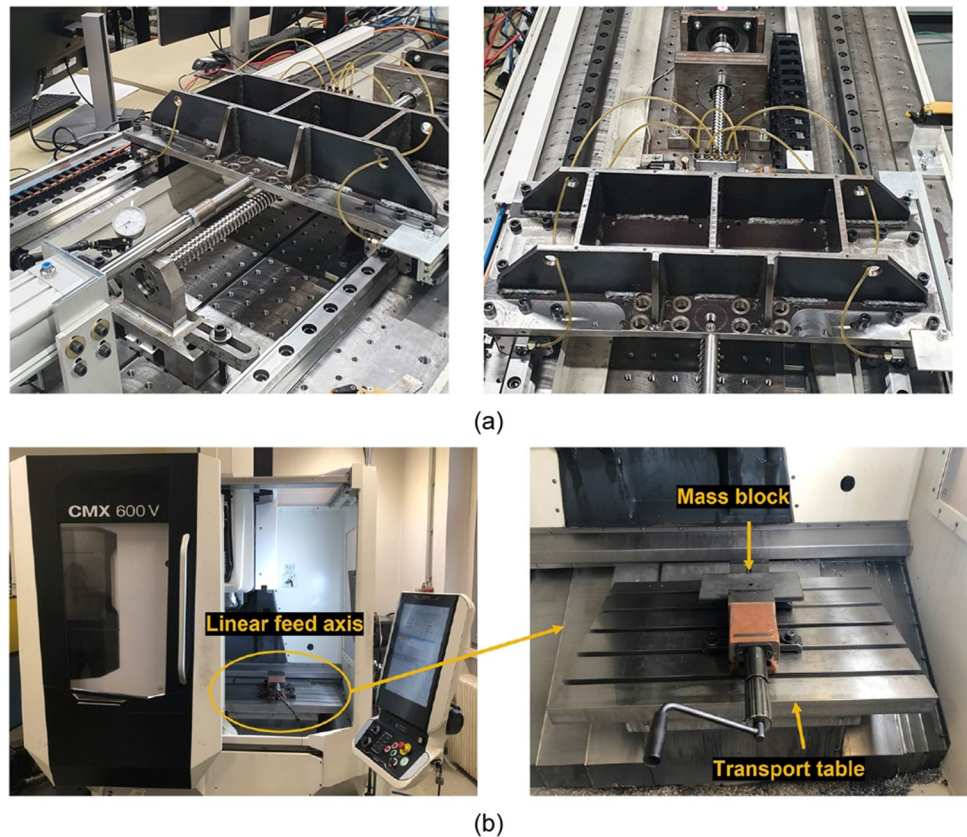
Mechanically forcing anomalies on CNC linear axes damages the machine. Not to damage the machines, anomaly experiments were performed on a different set of experiments and adapted to the normal state data obtained from the CNC.

A separate platform was prepared to simulate the motor which moves the linear table. This platform was called feed axis test platform (FATP) and presented in Fig. 7a. It was used to simulate different types of misalignments, application of external forces to the system, simulation of various misalignment issues, and collection of motor current data for analysis. Demetgul et al. [18] contributed to the development of the entire experimental platform and conducted all experiments on it, gathering various types of data. This paper utilized only certain categories of data for training purposes. The experimental pillow block was driven by a motor, reciprocating at 100 mm/s and 200 mm/s, and the stroke length was 400–500 mm. Five levels of horizontal left side misalignments (0.05 mm, 0.10 mm, 0.15 mm, 0.05 mm, and 0.25 mm) were introduced into

the linear feed axis. As the horizontal misalignment shifts to the left, the motor current magnitude slightly increases. The extent of this increase is positively correlated with the degree of horizontal misalignment introduced to the left.

After obtaining the data of the linear feed axis for the pre-training model, experiments were carried out to obtain motor current data from a real CNC machine. The experimental platform is the CMX 600V vertical milling CNC machine of the DMG MORI brand (Fig. 7b). Our experiments were carried out to obtain the actual data from this CNC machine's linear feed axis system without any abnormality. Figure 7 is a schematic diagram of the system structure. Important components were shown in the picture. In this context, the transport table serves as the object for error detection. It conveys the workpiece to be milled to a predefined position, with its linear motion in each direction (x , y , z) controlled by a single axis. In the experiment, only the x -axis in the horizontal direction was considered, and numerous different experiments were conducted. The linear feed system of the CNC machine operates without any mass block, using one mass block (100 N), two mass blocks (200 N) and three mass blocks (300 N). And then collect each motor current data through PLC. The sampling rate of this experiment was set to 500 Hz. Through this experiment, normal condition data under different loads was obtained.

Fig. 7 Experimental setup. **a** Feed axis test platform (FATP). **b** CNC



The relevant experimental parameters are shown in the following Table 1. In all experiments, the sampling rate of the motor current is kept constant at 500 Hz, and all collected data are normal data, except that the load is divided into four levels from 0 to 300N; under normal working conditions, it is normal to have a load, so the data collected under load also belongs to the normal state data. The speed is 100 mm/s, 200 mm/s, and 300 mm/s, respectively, and stroke is 100 mm, 200 mm, and 300 mm, respectively. As can be seen in the table, only the normal state and normal state tests with vertical load were carried out on the CNC machine. Anomaly experiments were adapted from the first set of experiments to the data of the CNC machine.

Figure 8 shows the overall flow of this work. First, the motor current time series data was collected from the linear feed axis experimental platform (Fig. 7a) and CNC machining center (Fig. 7b) through PLC. The data collected from the linear feed axis experimental platform was divided into single-period signals and normalized. Normal condition data, normal condition data with horizontal force applied, and different types of left-side misalignment anomaly data were obtained from this platform. However, only normal condition data and normal condition data under vertical force can be obtained from the CNC machining center, and no anomaly data and normal condition data with horizontal force application were available. To address this issue, the difference between the normal signal and other types of signals on the linear feed axis experimental platform was added to the normal signal of the CNC to synthetically create usable signals for training and testing. W-DCGAN models were then trained to enhance data for all types of data. t-SNE and TSTR evaluation indicators were used to test the ability of the W-DCGAN to generate similar signals. The normal condition linear feed axis current data was used to

pre-train various autoencoder models and complete hyperparameter optimization. The training results under the best hyperparameter combination were used as the standard for evaluating the pre-training model. Finally, the parameters of the pre-trained model were transferred to detect anomalies in CNC data. By training the model in a single scene and transferring the parameters to other scenes, the cost was significantly decreased.

3.1 Generation of anomaly data and normal data with horizontal forces

As mentioned in the previous section, normal state data can be collected from real CNC machines. Still, since there is no way to damage the machine or put force on the shaft, and most of the time it runs normally, it is difficult to capture and obtain anomaly data and normal condition data under load, so it was necessary to create the data synthetically.

Figure 9 shows how the synthetic signals for the CNC machine were created based on the difference between the normal signal and the other signals of the linear feed axis. The upper part of the diagram shows how a PLC was used to read a normal state current signal from a CNC machine and divide it into cycles. In the lower section of the depicted figure, the normal signal of the CNC machine was augmented with the difference between each left misalignment signal of the linear feed axis and its corresponding normal signal. Synthetic data was created with this approach and utilized for the purpose of training and testing. The anomaly signal will retain the amplitude change in the middle of the original linear axis signal. This amplitude change is the focus of our detection. The same method was used to create the normal state data under load. On the left column of the second and third rows, the normal and anomaly signals obtained by a specially designed feed axis test platform (FATP) were shown without and with the horizontal force application. At the third column from the left, in the same rows, the CNC data was shown.

In these experiments, the table of the CNC machining center was driven by the motor, at the speeds of 100 mm/s, 200 mm/s and 300 mm/s, while the stroke was set to 100–300 mm. Motor current signals were collected without and with vertical loads (Fig. 10b). The motor current signals under normal conditions with horizontal forces (Fig. 10a) and the motor current signals with different levels of horizontal misalignment (Fig. 10c) were both synthetically generated and added. Figure 10 a and b show the motor current without any abnormality under different horizontal and vertical loads, respectively. Figure 10c shows the current when the data was taken at different misalignment levels. In all cases, the table speed and stroke distance were 100 mm/s and 100 mm, respectively. It was observed that the current signal did not change under different vertical loads.

Table 1 Experimental parameters of CNC machine

Experiment	Normal	Normal with vertical loads
Horizontal misalignment (mm)	0	0
Vertical misalignment (mm)	0	0
Horizontal force (N)	0	0
Vertical force (N)	0	100 200 300
Speed (mm/s)	100 200 300	100 200 300
Stroke (mm)	100 200 300	100 200 300
Sampling rate (HZ)	500	500

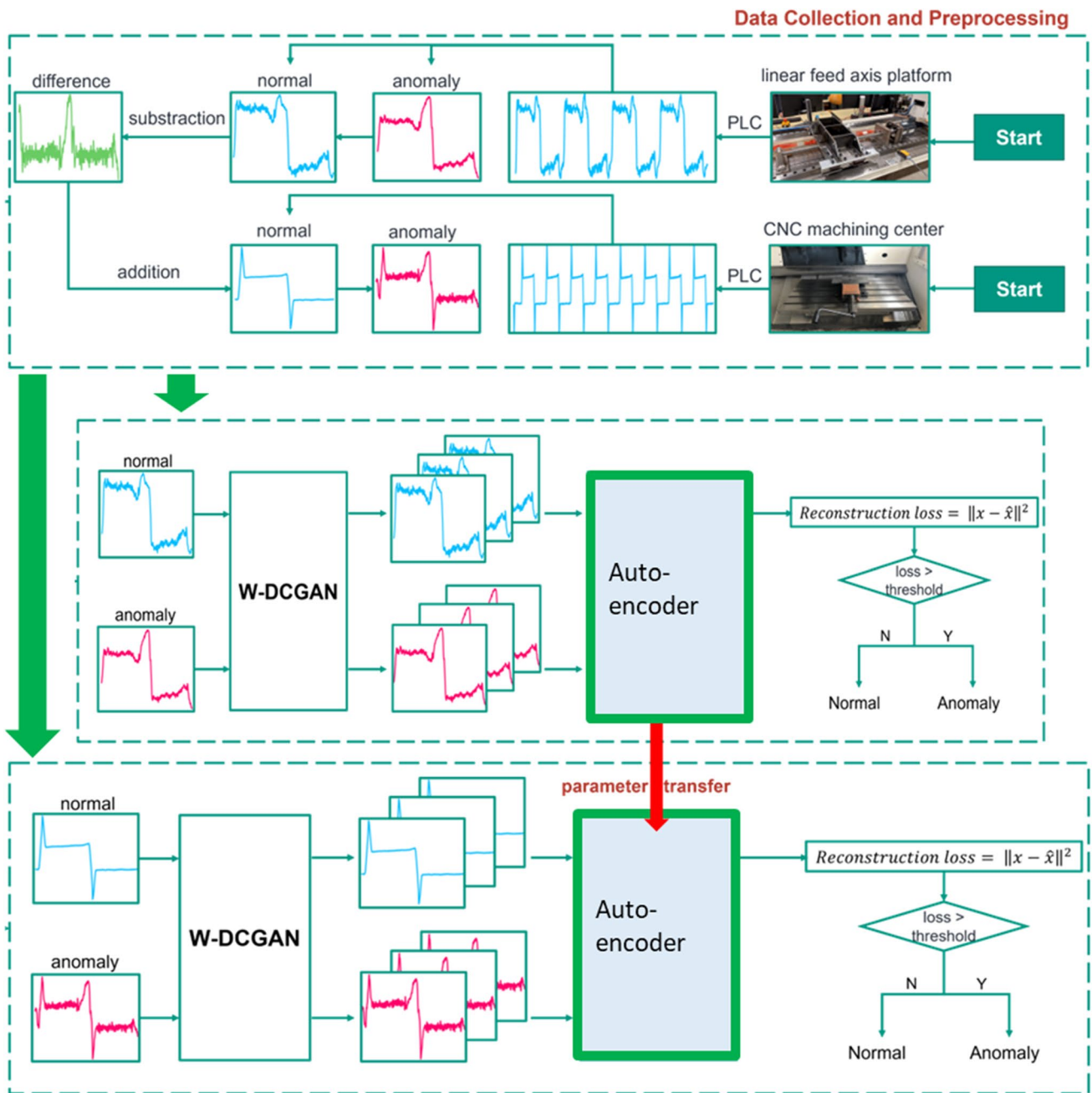


Fig. 8 Outline of the overall work. Experimental data collection at two platforms (FATP and CNC), transfer learning, and classification

4 Results and discussion

4.1 Data generation with W-DCGANs

W-DCGAN model was used to impose abnormalities into the data. In the experiment, 42 experimental data sets were collected by using the CNC machine with and without vertical loads, Since there was no difference in the signals, they were grouped in the same category here. Anomaly data and normal data with horizontal forces were synthetically

created, and a separate GAN model was trained for each one of the considered cases. Figure 11 shows the experimental data without any load and the same signal after the effect of 10 N horizontal force was simulated.

t-Distributed stochastic neighborhood embedding (t-SNE) is a statistical method for visualizing high-dimensional data by assigning each data point a location in a 2D or 3D map and has been used to evaluate the similarity between the data sets [56]. If the data points after t-SNE dimensionality reduction are very close in the two-dimensional or

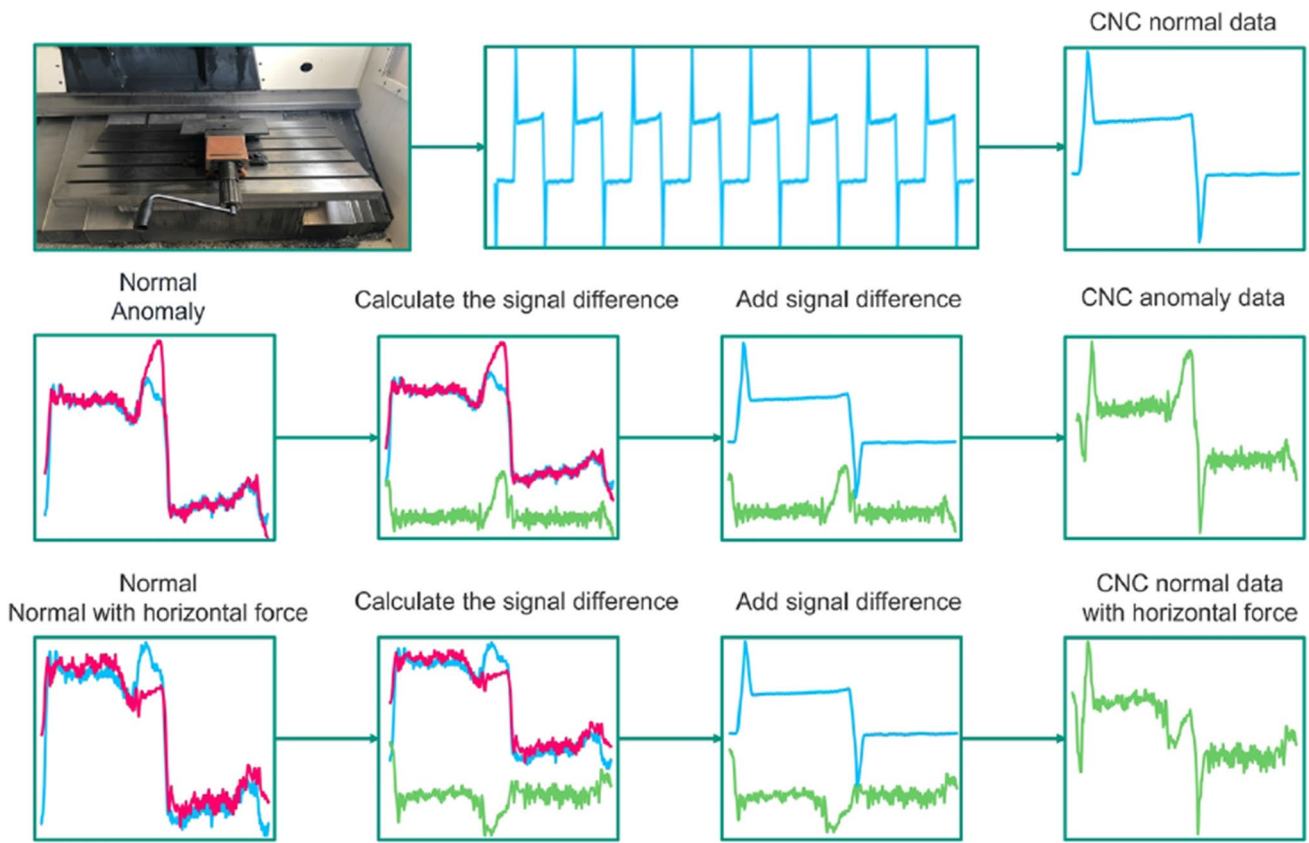
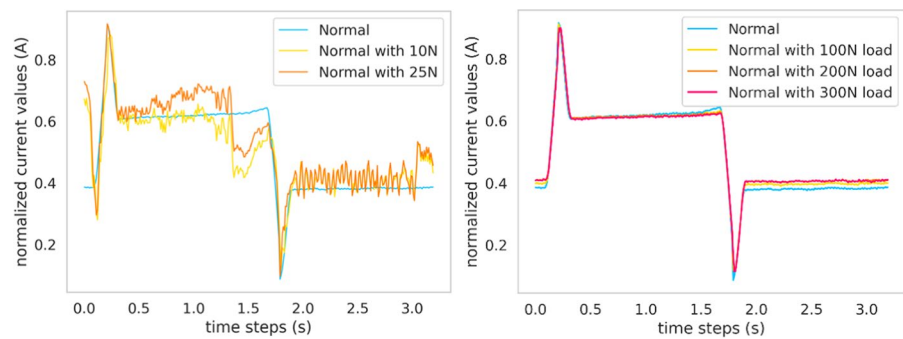
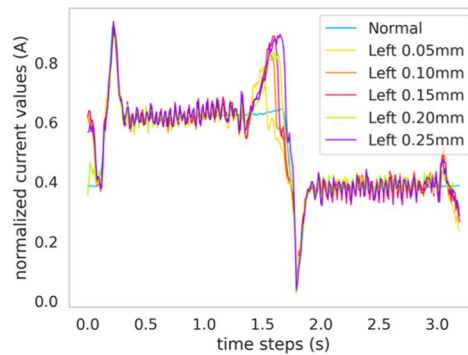


Fig. 9 Flowchart of synthetic data generation from the FATP and CNC data

Fig. 10 Preprocessed motor current signals of CNC machine at speed 100 mm/s



(a) Normal condition data with horizontal force (b) Normal condition data with vertical force



(c) Horizontal misalignment data

three-dimensional maps, it indicates that the spatiotemporal characteristics of the data are very similar. This approach was used to evaluate the similarity between synthetically generated and the experimental data.

Figure 12 shows the data point positions and corresponding similarities of experimental and synthetically generated cases for each type of data after dimensionality reduction by t-SNE. For example, in the figure, there is overlap between normal data points and other anomaly data points, this is because some anomaly signals were only slightly different from normal signals. However, it is evident that every type of data possesses a distinctive cluster center, indicating that the generated data is strikingly similar to the actual data and indistinguishable from it. This finding suggests that the

synthetically generated data exhibited the characteristics of the experimental data, but its diversity remained limited.

Figure 13 displays the positions of all real and synthetically generated data points, along with the similarity of normal data and anomaly data after dimensionality reduction through t-SNE, when all anomaly data are classified into one category.

Only 42 sets of data were recorded at the normal operating conditions by using the CNC without any abnormality. To obtain well-balanced descriptive training and test data, 12,000 synthetically generated cases without any anomaly and 10,000 synthetically generated cases with anomalies were generated by using the W-DCGAN model. Eighty percent of cases were used for training, and the remaining 20%

Fig. 11 Generation results for normal condition data with 10 N horizontal force applied

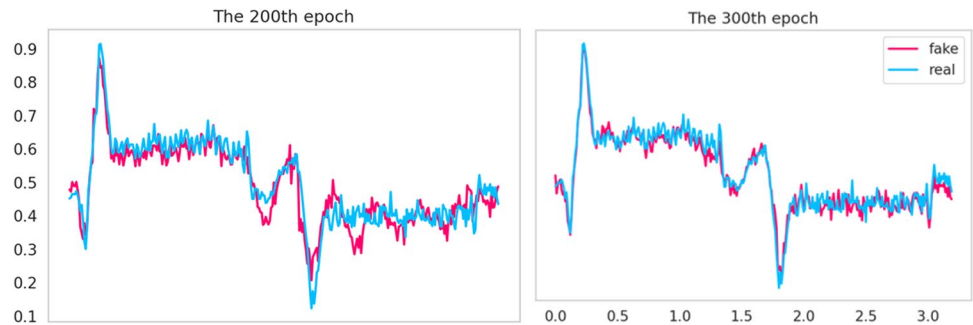


Fig. 12 Visualizations after reducing all the real and generated fake data points of each type of data to two- or three-dimensional map by t-SNE

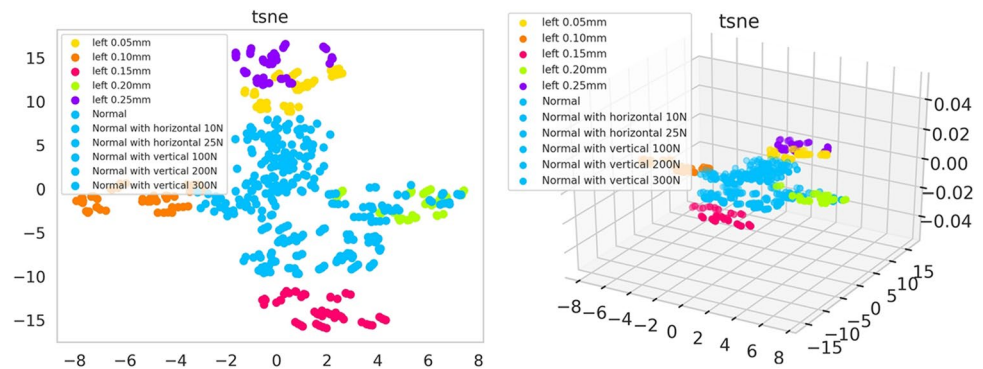
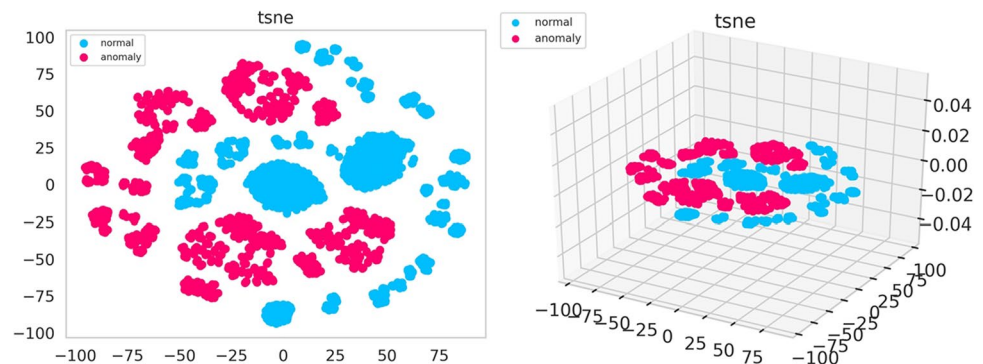


Fig. 13 Visualizations after reducing all the real and generated fake data points of normal and anomaly data to two- or three-dimensional map by t-SNE



were used for validation and testing purposes. The exact data numbers are listed in Table 2.

4.2 Threshold selection

The reconstruction loss distribution of each autoencoder model for all data is shown in Fig. 14: from top to bottom, (a) CNN-AE, (b) TCN-AE, (c) LSTM-AE, and (d) CNN-LSTM-AE. Blue points represent reconstruction errors for normal signals, and red points represent reconstruction

errors for anomaly data. In the figure, it is also easy to draw a threshold between the reconstruction errors of normal data and anomaly data to distinguish them. The threshold selection is based on maximizing the classification accuracy of all data.

4.3 Classification results

Almost all researchers have been using confusion matrices in recent studies to evaluate the performance of the classification method since their compact diagrams show the number of tests and number of the correct and wrong estimations at a single glance. After selecting the optimal threshold based on the classification accuracy, the confusion matrices of the four models are presented in Fig. 15. All models achieved excellent results. The CNN-LSTM-AE combination obtained the best accuracy. This excellence also confirmed the validity of the proposed synthetic data generation procedure in this paper.

4.4 Discussion

Generation of training data by using analytical models for manufacturing applications started in early 1990s. Tansel's group used analytical model-based simulations for training conventional neural networks to detect chatter in turning [57] and tool breakage in milling operations [58]. Recently, transfer learning was used in milling [59, 60] and turning [61] operations. Unver and Sener [59], and Postel et al. [60] used analytical models to generate all or most

Table 2 Number of experimental and synthetic data in different categories

Type	Original	Synthetic	All	Ratio
Normal	21	2000	2021	5.07%
Normal with vertical 100 N	7	2000	2007	5.04%
Normal with vertical 200 N	7	2000	2007	5.04%
Normal with vertical 300 N	7	2000	2007	5.04%
Normal with horizontal 10 N	882	2000	2882	7.23%
Normal with horizontal 25 N	882	2000	2882	7.23%
Anomaly 0.05 mm	3360	2000	5360	13.45%
Anomaly 0.10 mm	3360	2000	5360	13.45%
Anomaly 0.15 mm	3150	2000	5150	12.92%
Anomaly 0.20 mm	3150	2000	5150	12.92%
Anomaly 0.25 mm	3024	2000	5024	12.61%
Total normal	1806	12,000	13,806	34.65%
Total anomaly	16,044	10,000	26,044	65.36%
Total	17,850	22,000	39,850	100.0%

Fig. 14 Distribution of reconstruction loss using various autoencoders after transfer learning

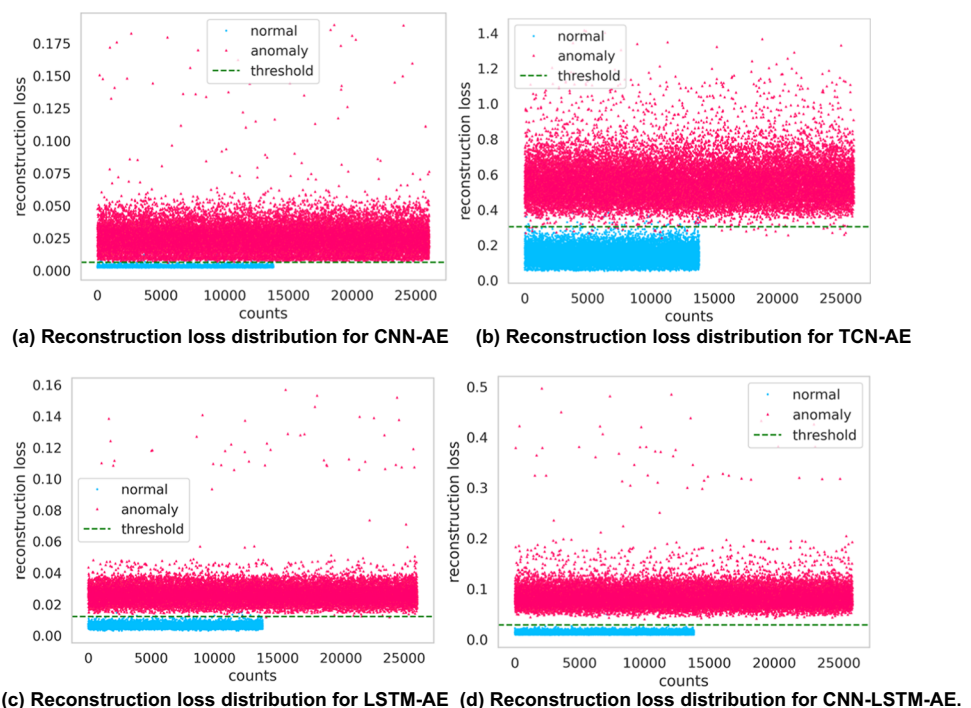
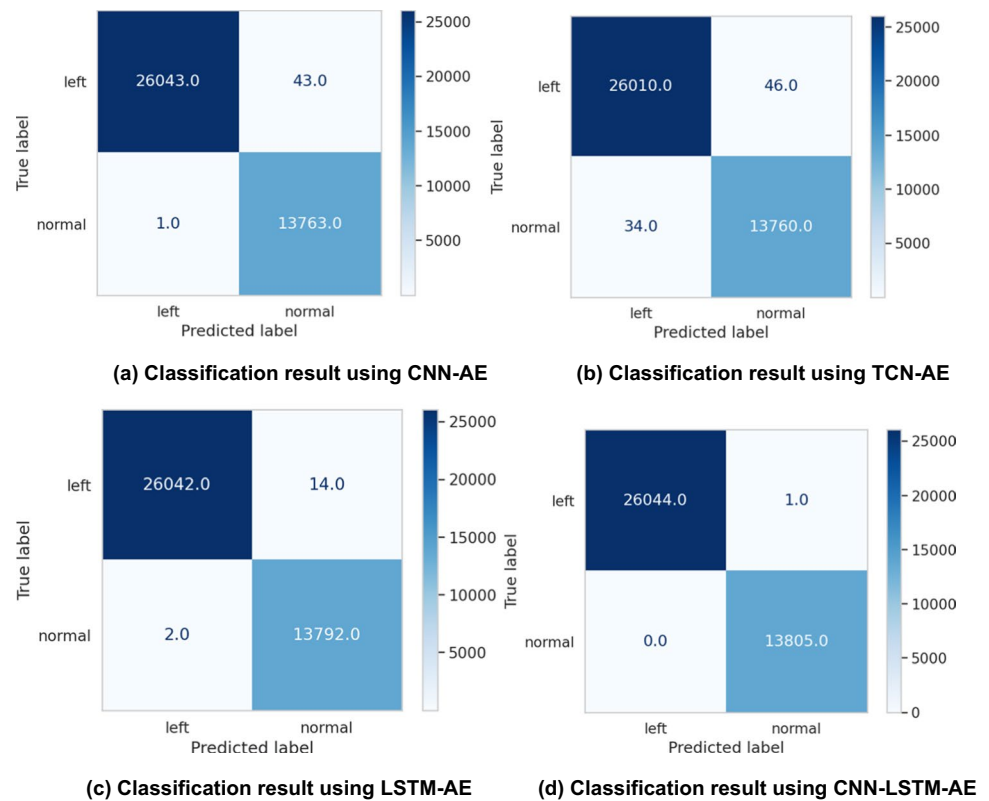


Fig. 15 Horizontal misalignment classification results using various autoencoders based on optimal thresholds after transfer learning



of their training data respectively. Yesilli et al. [61] trained their classifiers with 5.08 cm stickout cases. They tested the classifier with 11.43cm stickout cases to evaluate the transfer learning capability of the classifiers.

In this study, only experimental data was used for transfer learning. W-DCGAN model used experimental data to generate the 22,000 synthetically generated cases. Experimental data was collected by performing 42 tests by using the CNC machine and 17,808 tests by using FATP as shown in Table 2.

5 Conclusion and future work

An innovative method was proposed to estimate the misalignments of the linear tables of CNC machines without attaching any sensors on the machine tool. Use of the motor current signals collected by the controller was proposed. Experimental data was collected for the study by using an experimental platform which was constructed to induce different levels of horizontal misalignments mechanically and a CNC machine. Data was collected when the linear table was in normal operating condition and when different misalignments were imposed. Additional data was collected by using a computer numerical control (CNC) machine without creating any anomaly.

A Wasserstein deep convolutional generative adversarial network has been developed to generate synthetic data that captures the effect of anomalies experienced on the experimental platform, using data collected from a CNC machine. The generated signal was compared to the original signal in real-time by using t-SNE. Additionally, data augmentation was applied to improve the robustness and generalization ability of models trained on this dataset. t-SNE showed that the synthetically created data had the same characteristics with the experimental data collected on the CNC machine.

Autoencoders were used to classify motor current signals, and random search and grid search techniques were used to find their optimal parameters. The CNN-AE, TCN-AE, LSTM-AE, and CNN-LSTM-AE combinations were used to classify the test cases after training. All models achieved good classification results. The highest accuracy was achieved by combining CNN, LSTM, and AE models. These results also demonstrated the feasibility of the proposed synthetic data generation approach and great performance of the combining CNN, LSTM, and AE models.

In summary, the models proposed can detect axial misalignment in the horizontal direction very well on CNC machines, with 0.3% error. In addition, Wasserstein deep convolutional generative adversarial network was an excellent tool for transfer learning.

The results demonstrated that monitoring the motor loads of machine tool controllers can effectively detect certain

geometric errors, such as misalignment. These findings can be further extended to estimate tool offsets and identify spindle errors associated with radial, axial, and angular deviations. Moreover, aside from geometric errors, there have been extensive studies focusing on tool wear estimation, chatter detection, and tool breakage identification using signals from CNC controllers. These studies could be integrated to develop intelligent CNC controllers capable of evaluating both machine geometric errors and machining performance simultaneously.

Acknowledgements Mr. David Perez edited the paper with ChatGPT's suggestions. Authors appreciated the capabilities of this large language model trained by OpenAI; thanks to the organization for providing open access to public. Detailed information about this study is presented at the following Web site: https://github.com/Macallen117/current_based_anomaly_detection_with_GAN_and_AE.

Author contribution Not applicable.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability Not applicable.

Code availability Not applicable.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Lu C, Wang ZY, Qin WL, Ma J (2017) Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification. *Signal Process* 130:377–388
- Chen Z, Deng S, Chen X, Li C, Sanchez RV, Qin H (2017) Deep neural networks-based rolling bearing fault diagnosis. *Microelectron Reliab* 75:327–333
- Shao H, Jiang H, Zhao H, Wang F (2017) A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. *Mech Syst Signal Process* 95:187–204
- Shao H, Jiang H, Lin Y, Li X (2018) A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mech Syst Signal Process* 102:278–297
- Shao H, Jiang H, Wang F, Zhao H (2017) An enhancement deep feature fusion method for rotating machinery fault diagnosis. *Knowl Based Syst* 119:200–220
- Chen Z, Li W (2017) Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. *IEEE Trans Instrum Meas* 66(7):1693–1702
- Song X, Sun P, Song S, Stojanovic V (2023) Finite-time adaptive neural resilient DSC for fractional-order nonlinear large-scale systems against sensor-actuator faults. *Nonlinear Dyn* 111:12181–12196
- Djordjevic V, Tao H, Song X, He S, Gao W, Stojanovic V (2023) Data-driven control of hydraulic servo actuator: An event-triggered adaptive dynamic programming approach. *Math Biosci Eng* 20(5):8561–8582
- Huang S, Tang J, Dai J, Wang Y (2019) Signal status recognition based on 1DCNN and its feature extraction mechanism analysis. *Sensors* 19(9):2018
- Zhang X, Han P, Xu L, Zhang F, Wang Y, Gao L (2020) Research on bearing fault diagnosis of wind turbine gearbox based on 1DCNN-PSO-SVM. *IEEE Access* 8:192248–192258
- Shao Y, Yuan X, Zhang C, Song Y, Xu Q (2020) A novel fault diagnosis algorithm for rolling bearings based on one-dimensional convolutional neural network and INPSO-SVM. *Appl Sci* 10(12):4303
- Bi F, Ma T, Wang X (2019) Development of a novel knock characteristic detection method for gasoline engines based on wavelet-denoising and EMD decomposition. *Mech Syst Signal Process* 117:517–536
- Tong Z, Li W, Zhang B, Zhang M (2018) Bearing fault diagnosis based on domain adaptation using transferable features under different working conditions. *Shock Vib* 2018. <https://doi.org/10.1155/2018/6714520>
- Chen H, Jiang B, Chen W, Yi H (2018) Data-driven detection and diagnosis of incipient faults in electrical drives of high-speed trains. *IEEE Trans Ind Electron* 66(6):4716–4725
- Vashisht RK, Peng Q (2021) Online chatter detection for milling operations using LSTM neural networks assisted by motor current signals of ball screw drives. *J Manuf Sci Eng* 143(1):011008
- Cho SB, Park HJ (2003) Efficient anomaly detection by modeling privilege flows using hidden Markov model. *Comput Secur* 22(1):45–55
- Sagheer A, Kotb M (2019) Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems. *Sci Rep* 9(1):1–16
- Giannoni F, Mancini M and Marinelli F (2018) Anomaly detection models for IoT time series data Preprint at <https://arxiv.org/abs/1812.00890>
- Shaikh KB, Jawarkar NP, Ahmed V (2021) Machine diagnosis using acoustic analysis: A review. In: 2021 IEEE Conference on Norbert Wiener in the 21st Century (21CW). IEEE, pp 1–6
- Mustafa D, Yicheng Z, Minjie G, Jonas H, Jürgen F (2022) Motor current based misalignment diagnosis on linear axes with short-time Fourier transform (STFT). *Procedia CIRP* 106:239–243
- Demetgul, M., Zihan, M., Heider, I., & Fleischer, J. (2022). Sensorless Misalignment Detection on Linear Feed Axis with Revised ResNet and Transfer Learning Using Motor Current.

22. Park P, Marco PD, Shin H, Bang J (2019) Fault detection and diagnosis using combined autoencoder and long short-term memory network. *Sensors* 19(21):4612
23. Jiang W, Cheng C, Zhou B, Ma G and Yuan Y (2019) A novel gan-based fault diagnosis approach for imbalanced industrial time series. Preprint at <https://arxiv.org/abs/1904.00575>
24. Sabir R, Rosato D, Hartmann S, Gühmann C (2021) Signal generation using 1d deep convolutional generative adversarial networks for fault diagnosis of electrical machines. In: In 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, pp 3907–3914
25. Luleci F, Catbas FN, Avci O (2022) Generative adversarial networks for labelled vibration data generation. In: Special Topics in Structural Dynamics & Experimental Techniques, Volume 5: Proceedings of the 40th IMAC, A Conference and Exposition on Structural Dynamics 2022. Springer International Publishing, Cham, pp 41–50
26. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J Mach Learn Res* 9(11):2579–2605
27. Jordon J, Yoon J, Van Der Schaar M (2019, May) PATE-GAN: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations
28. Yoon J, Jarrett D, Van der Schaar M (2019) Time-series generative adversarial networks. *Adv Neural Inf Process Syst* 32
29. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
30. Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science
31. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
32. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. Preprint at <https://arxiv.org/abs/1412.3555>
33. Son J, Kim C, Jeong M (2022) Unsupervised learning for anomaly detection of electric motors. *Int J Precis Eng Manuf* 23(4):421–427
34. Ji Z, Gong J, Feng J (2021) A novel deep learning approach for anomaly detection of time series data. *Sci Prog* 2021. <https://doi.org/10.1155/2021/6636270>
35. Shi Z, Mamun AA, Kan C, Tian W, Liu C (2022) An LSTM-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing. *J Intell Manuf* 1–17
36. Khan ZA, Hussain T, Ullah A, Rho S, Lee M, Baik SW (2020) Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework. *Sensors* 20(5):1399
37. Yin C, Zhang S, Wang J, Xiong NN (2020) Anomaly detection based on convolutional recurrent autoencoder for IoT time series. *IEEE Trans Syst Man Cybern Syst* 52(1):112–122
38. Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P and Shroff G (2016) LSTM-based encoder-decoder for multi-sensor anomaly detection. Preprint at <https://arxiv.org/abs/1607.00148>
39. Nogas J, Khan SS, Mihailidis A (2018) Fall detection from thermal camera using convolutional lstm autoencoder. In: Proceedings of the 2nd workshop on aging, rehabilitation and independent assisted living. IJCAI workshop
40. Kim TY, Cho SB (2018) Web traffic anomaly detection using C-LSTM neural networks. *Expert Syst Appl* 106:66–76
41. Li G, Fu Y, Chen D, Shi L, Zhou J (2020) Deep anomaly detection for CNC machine cutting tool using spindle current signals. *Sensors* 20(17):4896
42. Yang Z, Zhang W, Cui W, Gao L, Chen Y, Wei Q, Liu L (2022) Abnormal Detection for Running State of Linear Motor Feeding System Based on Deep Neural Networks. *Energies* 15(15):5671
43. Li C, Zhang S, Qin Y, Estupinan E (2020) A systematic review of deep transfer learning for machinery fault diagnosis. *Neurocomputing* 407:121–135
44. Mao W, Ding L, Tian S, Liang X (2020) Online detection for bearing incipient fault based on deep transfer learning. *Measurement* 152:107278
45. Wan Z, Yang R, Huang M (2020) Deep transfer learning-based fault diagnosis for gearbox under complex working conditions. *Shock and Vibration* 2020:1–13
46. Shao S, McAleer S, Yan R, Baldi P (2018) Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Trans Industr Inform* 15(4):2446–2455
47. Geng Z, Tong Z, Jiang X (2021) Review of geometric error measurement and compensation techniques of ultra-precision machine tools. *Light. Adv Manuf* 2(2):211–227
48. Zhang Z, Jiang F, Ming LUO, Baohai WU, Zhang D, Tang K (2023) Geometric error measuring, modeling, and compensation for CNC machine tools: A review. *Chin J Aeronaut*. <https://doi.org/10.1016/j.cja.2023.02.035>
49. Zhao L, Cheng K, Chen S, Ding H, Zhao L (2019) An approach to investigate moiré patterns of a reflective linear encoder with application to accuracy improvement of a machine tool. *Proc Inst Mech Eng B J Eng Manuf* 233(3):927–936
50. Radford A, Metz L and Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint at <https://arxiv.org/abs/1511.06434>
51. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S et al (2020) Generative adversarial networks. *Commun ACM* 63(11):139–144
52. Smith KE and Smith AO (2020) Conditional GAN for timeseries generation. Preprint at <https://arxiv.org/abs/2006.16477>
53. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International conference on machine learning. PMLR, pp 214–223
54. Kong X, Li X, Zhou Q, Hu Z, Shi C (2021) Attention recurrent autoencoder hybrid model for early fault diagnosis of rotating machinery. *IEEE Trans Instrum Meas* 70:1–10
55. Thill M, Konen W, Bäck T (2020) Time series encodings with temporal convolutional networks. In: Bioinspired Optimization Methods and Their Applications: 9th International Conference, BIOMA 2020. Springer International Publishing, Brussels, Belgium, pp 161–173
56. Tao H, Qiu J, Chen Y, Stojanovic V, Cheng L (2023) Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion. *J Franklin Inst* 360(2):1454–1477
57. Tansel IN, Wagiman A, Tziranis A (1991) Recognition of chatter with neural networks. *Int J Mach Tool Manuf* 31(4):539–552
58. Tansel IN, McLaughlin C (1993) Detection of tool breakage in milling operations—II. The neural network approach. *Int J Mach Tool Manuf* 33(4):545–558
59. Unver HO, Sener B (2022) Exploring the Potential of Transfer Learning for Chatter Detection. *Procedia Comput Sci* 200:151–159
60. Postel M, Bugdayci B, Wegener K (2020) Ensemble transfer learning for refining stability predictions in milling using experimental stability states. *Int J Adv Manuf Technol* 107:4123–4139
61. Yesilli MC, Khasawneh FA, Otto A (2020) On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition. *CIRP J Manuf Sci Technol* 28:118–135

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.