Article | Received 11 November 2023; Accepted 22 May 2024; Published 27 June 2024
https://doi.org/10.55092/am20240010

Transfer learning methods for Three-Axis CNC anomaly detection

Eugene Li[1,*], Yang Li[1], Sasank Kakarla[2], Sanjeev Bedi[1] and William Melek[1]

[1]University of Waterloo, 200 University Avenue West, Waterloo, Ontario, N2L3G1, Canada
[2]Hurco Companies Inc, Indiana 46268, USA

* Correspondence author; E-mail: eugene.li@uwaterloo.ca.

Abstract: Transfer learning is a machine learning method used to train a network using prior knowledge from a source network. This method has been applied for anomaly detection in CNC machines to detect anomalies such as chatter. Before transfer learning can be applied, the Maximum Mean Discrepancy (MMD) score between the source and target data sets should be evaluated. A low MMD score indicates that the system can be transferred, but it is unclear what the physical implication of this score is. Experiments were conducted on a variety of CNC machines and materials to determine a relationship between this score and observed physical outputs and other measures. Through experimental testing, it was found that a low MMD score will indicate that the source and target machines will respond to an input signal in a similar fashion. Furthermore, the MMD scores and cross-entropy followed the same trend, but this did not correlate with the accuracy of the system.

Keywords: Artificial intelligence, transfer learning, CNC, industrial, anomaly detection

1. Introduction

Accelerated by recent trade tensions, conflicts and the COVID-19 pandemic, there has been an increased attention on re-shoring manufacturing capabilities within North America [1]. Re-shoring involves the returning of production and manufacturing capabilities back to a company's original country and has the potential to provide many benefits, such as decreased transportation and logistics and reduced supply chain disruptions [2]. However, an impediment to many re-shoring initiatives is the lack of skilled labor and manufacturing knowledge. To address this gap, there has been an increased effort in automating the manufacturing process.

CNC milling machines are the ideal tool to implement more automated manufacturing processes. Much of the current operation of CNC machines has already been automated, with one operator able to monitor several machines. Although modern CNC machines are capable of significant automation, they still operate in a primarily open-loop configuration and are not able to detect or react when anomalies occur [3]. There have been many attempts to address this gap, such as the machine learning-based tool condition monitoring in [4] and the indirect tool condition monitoring through machine learning in[5], but this problem still remains an issue yet to be widely solved.

In recent years, there have been significant advances in artificial technology and machine learning. These advances have produced many successful data-driven approaches to anomaly detection. Examples of this are the deep transfer learning of the machining process in [6], the deep learning anomaly detection using spindle current in [7], and the deep learning approach to learn tool dynamics in [8]. While these methods work well in a lab setting, they often do not extend well to production environments due to their need for large amounts of training data before they can be implemented. For users in the industry to spend time collecting potentially millions of training data points instead of producing useful parts results in significantly reduced productivity and wasted operator time, a costly proposition.

The authors proposed to use transfer learning as a solution to collecting a large amount of training data in production. In [9], it was found that users could invest a large amount of time upfront in a controlled lab environment to collect the necessary training data and then train an anomaly detection network. This network could then be transferred to a production machine using a fraction of the training data that could be quickly collected within several dozen sample cuts. This approach significantly reduces the amount of necessary training data that needs to be collected outside of the lab, and allows for many advanced AI methods to be implemented quickly.

Although the transfer learning methods outlined in our previous work [9] are effective, the limitations of transfer learning were not clear. For example, for transfer learning to be successful from a source to a target domain, there must be overlap between the source and target data. This is measured by comparing the Maximum mean discrepancy (MMD), which is a kernel based statistical test that is used to determine the similarity between two given distributions and described in [10] and [11]. Although this has a mathematical representation, the physical interpretation of this test is unclear.

The objective of this paper is to provide physical intuition on what the MMD score represents, and to detail the conditions for which transfer learning for anomaly detection in three axis CNC machines will be most successful. This includes investigating transferring networks between machines of different manufacturers, transferring networks between machines with different but similar configurations, and applying networks on different materials on the same machine and machines of different sizes. This paper will provide the reader with some preliminary background on LSTM-Autoencoders for anomaly detection, transfer learning and some discussion on mechanistic modelling. The organization of this paper is as follows: In section 2, the necessary background information on the LSTM Autoencoder algorithm, transfer learning and mechanistic modelling will be outlined. In section 3, the methodology implemented will be discussed. In section 4, the experimental design and results will be presented and discussed. Section 5 will provide a conclusion of these findings.

2. Background

2.1. Anomaly detection of CNC machines using mechanistic modeling

In traditional machining approaches, cutting conditions are selected in a conservative manner based on operator experience. Mechanistic cutting force models for high-speed machining have been developed to calculate in-process chip geometry. These models can be used to calculate tool vibration, tool deflection, and cutting forces [12]. As outlined in [13], the mechanistic model of a CNC machine creates a model of the cutting forces from the end mill and are related to the uncut chip thickness bh through empirical cutting force coefficients K_t and K_r. These coefficients are related to the normal cutting force F_r and tangential cutting force F_t by

$$F_t = K_t bh \qquad (1)$$

$$F_r = K_r bh \qquad (2)$$

Where the coefficients are measured during cutting experiments and extracted by linear curve fitting with varying chip thickness h [14]. These models have been employed successfully in the analysis of machine tool dynamics but need to be re-obtained for each milling operation with different cutter geometry and workplace material [14]. The above equations can also be modified to represent basic non-linearity in the cutting force expressions when the edge forces are neglected in the mechanistic models [13]. This is achieved by modifying the cutting constants as follows

$$K_t = K_T h^{-p} \qquad (3)$$

$$K_f = K_F h^{-q} \qquad (4)$$

Where p and q are also cutting force constants obtained from experiments conducted at different feed rates. These mechanistic models can then be used to model the cutting chip thickness as

$$h(t) = h_0 - [y(t) - y(t-T)] \qquad (5)$$

Where h_0 is the intended chip thickness, which is equal to the feed rate of the machine, and y(t)-y(t-T) is the dynamic chip thickness produced during a spindle revolution [13]. The equations of motion for the machine can be described as

$$m_y\ddot{y}(t) + c_y\dot{y} + k_y y(t) = K_f\alpha[h_0 + y(t-T) - y(t)] \qquad (6)$$

In [15], it was identified that the main source of chatter and vibrational instability in machining was due to regeneration in subsequent cuts. This regeneration results in the end mill bouncing away from the cutting surface between tool revolutions due to an increase in energy in the system. This results in a zero-chip distance and zero cutting force when the tool disengages with the cutting surface, resulting in chatter marks being left on the machining part, producing undesirable parts with poor finish. To address these issues, researchers used the dynamics described in Equation. 5 to attempt to predict the conditions when chatter would occur. It was determined that the phase difference between two consecutive waves could be modeled as a result of the structural dynamics of the machine tool, and the mechanics of the orthogonal cutting, resulting in a linear delay differential equation [13]. Solving this equation produced what is known as the "stability lobe". The stability lobe produces a plot that outlines the relationship between cutting depth and spindle speed, indicating where the dynamics of the system are stable and when chatter may occur. Although these methods are very effective and have been used within the literature and in production for decades, these stability lobes can be time-consuming to obtain. Matters are further complicated when the machine dynamics change with extended use, causing the stability lobes to become inaccurate. To avoid this potential, many users in production run their machines in very conservative modes and significantly restrict the output potential of the machines.

2.2. *Anomaly detection using LSTM-Autoencoders*

Recent advances in machine learning algorithms have produced very powerful techniques, capable of modeling complex systems. One such technique is the Long Short-Term Memory (LSTM) network. This network is capable of learning complex patterns over a period of time, and capturing both the long-term and short-term dynamics of a system in the network.

As outlined in [16], the LSTM architecture can be defined with matrices $W \in \mathbb{R}^{h\times d}$ and $U \in \mathbb{R}^{h\times h}$ and vector $b \in \mathbb{R}^{h}$, which contain, respectively, the weights of the input and recurrent connections, where the subscript q can either be the input gate i, output gate o, the forget gate f, or the memory cell c. These weights and bias vector parameters are learned during training, where the superscripts d and h refer to the number of input features and number of hidden units, respectively.

$$g_t = \tilde{C}_t = tanh(W^g x_t + U^g h_{t-1} + b^g) \quad (7)$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f) \quad (8)$$

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i) \quad (9)$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o) \quad (10)$$

Where the operator $\odot$ denotes the Hadamard multiplication, σ represents a sigmoid function and the subscript t indexes the time step. The variables used are summarised in Table 1.

Table 1. Variables used in LSTM.

Parameter Name	Notation
x_t	Input vector to the LSTM unit
g_t /$\tilde{C}_t$	Cell input activation vector
C_t	Cell state vector
f_t	Forget gate's activation vector
i_t	Input vector to the LSTM unit
o_t	Output gate's activation vector

This network can then be arranged in what is known as an Autoencoder configuration. In an LSTM Autoencoder, we develop a network with several layers for encoding, an embedding layer and a decoding layer using LSTM layers. This network topology can be seen in Fig. 1, with the first half of the network known as the encoder, and the second half of the network known as the decoder. The encoder is responsible for reading the input sequence step-by-step and extracting the hidden information in the data, capturing the system's dynamics. The decoder then takes this hidden information and reconstructs the original signal.

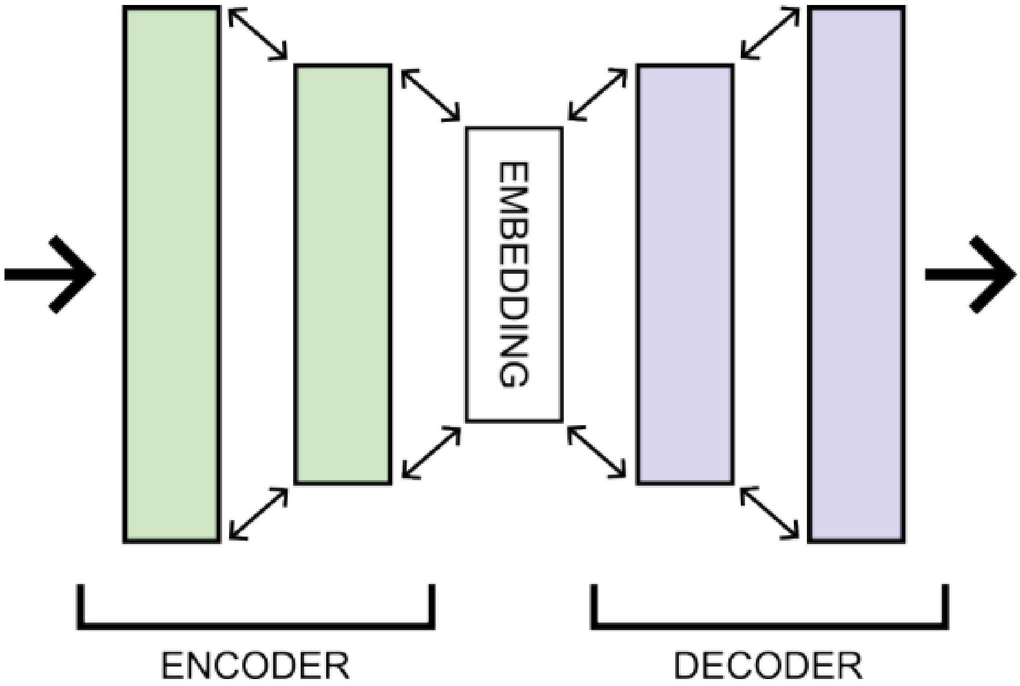

Figure 1. LSTM Autoencoder Architecture.

In anomaly detection we train the LSTM Autoencoder to reconstruct stable, proper signals based on stable cutting conditions. In this instance, if the network is given stable inputs, it will reproduce a stable output with little reconstruction error. However, if the signals original from an anomaly, the network will be unable to reconstruct the signal well, resulting in a large reconstruction loss. This increased reconstruction loss can be used to determine if an anomaly has occurred, by noting if the difference between the signals is greater than a pre-defined threshold. This threshold can be determined by plotting the loss distribution histogram from a given sample of stable conditions, and then determining the mean and three-sigma distance using traditional statistical methods. This three-sigma distance allows us to determine a threshold for reconstruction error, since reconstruction losses beyond three-sigma, most likely represent outliers and do not belong to the proper set, and thus represent anomalies.

2.3. Neural Network training using transfer learning

Transfer learning is a machine learning method where knowledge from a first system is used as a starting point for training a second related system, with similar dynamics and limited new information [17]. This method allows us to take an existing model trained on one data set, and adapt it to predict another similar data set, saving significant time and cost by re-using many of the previous learned weights [18]. This approach is often used in vision processing tasks where there may be similarity in the images. For example, a network trained to identify cars can be used as a starting point to train a network to identify trucks.

The mathematical modelling of transfer learning is expressed in detail in [19], but is presented in brief below. Following the conventions defined in the methodology of [17], we begin by defining a domain $D = \{x, P(X)\}$, containing a feature space x and the probability distribution $P(X)$, where $X = x_1, x_2, \ldots, x_n$. A task can then be presented by $J = \{Y, f(x)\}$, where y represents a label space and $f(x)$ a target function. These labels and target functions are learned from the training data consisting of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in Y$.

Given a learning task J_s from source domain D_s, and a target domain D_T and learning task J_T, where $D_S \neq D_T$, or $J_S \neq J_T$, transfer learning aims to improve the predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and J_S. This is accomplished by using the latent knowledge from D_S and J_S. In most cases the size of D_S is much larger than D_T [17].

In practice this is often accomplished by first developing a deep model based on the source domain data set. Following this, the final layers of the model are removed and replaced with new layers with randomly defined weights, or additional layers are added on top of the source model. The source model layer weights are then frozen and the new data set from the target domain is used to train the new layers [18].

In our previous work, outlined in detail in [9], we developed a transfer learning strategy for anomaly detection. This strategy involved developing an LSTM Autoencoder using a large dataset from one three axis CNC milling machine. We would then freeze the encoding layers of this Autoencoder and transfer the weights to a new network. The decoder was then trained using a fraction of the data from the target machine. This approach of transfer learning, known as transductive transfer learning, falls within the realm of domain adaptation. Namely, we aim to develop a model from a source distribution and apply that model on a different but related target distribution to achieve the same sort of tasks from the source and target domains. In this case, these tasks are the detection of anomalies in CNC milling.

However, for transfer learning to be successful from the source domain to the target domain, there must be sufficient overlap between the two data sets. This overlap ensures that both the source and target domain are reflecting similar environments, making the transferred information useful. For example, in the context of a CNC machine, if the source data contained only drilled holes but the target data was entirely curved surfaces, one would expect there to be little overlap.

This analysis can be accomplished by comparing the similarity between the training data in both the source and target domains using methods described in works such as the inner product defined in [19], or more recently, using statistical measures such as the Maximum Mean Discrepancy (MMD). The MMD metric was first described in [11] and has been used in works such as [20] and [6] to determine the feasibility of transfer learning. The MMD is a kernel based statistical test that is used to determine if two data sets represent the same distribution. In the context of transfer learning, if both the source and target training data have a low MMD score, then we can conclude that they both represent similar domains and transfer learning will be successful.

Given two distributions X and Y, we assume there is a feature map $\emptyset(*)$ that maps the distributions to a feature space F. The MMD is calculated as:

$$MMD^2(X,Y) = \parallel \mu_X - \mu_Y \parallel_F^2 \qquad (11)$$

where μ represents the mean of the distribution in the feature space.

This expression has a straightforward description, but is often difficult to implement. Thus, we use an empirical estimation by representing the system with a kernel $k(x_i, x_j)$. For this implementation, we have chosen a linear kernel, since we are not aware of any particular distribution that the data may take. The choice of a linear kernel may not be fully representative, but allows for a simple consistent baseline to evaluate the different options.

3. Methods

To evaluate the feasibility of transfer learning methods in machining applications, and to draw a more intuitive understanding of the numerical analysis, several experiments were conducted where these approaches could be applied. These experiments were conducted with the assistance of an industrial machine shop that has a wide variety of CNC milling machines. This machine shop services many customers looking for custom parts and has a variety of machines with different, but similar capabilities, providing the ideal test bench for implementing the test cases.

The transfer learning algorithms described in the methodology section of [9] were applied with a source network being trained on a large data set and then a re-trained with a smaller target data set from each of the test cases. To properly evaluate each of the different scenarios, the large data sets all had approximately 250 000 training samples, while the target sets had roughly 23 000 test samples. Although more data was available in some cases, these training parameters were kept constant to try and isolate any differences in the system to differences from the data and not for the amount of available data. During training, 80were randomly selected for training and the remainder was used for validation. These datasets were composed of a variety of cutting conditions at various depths and speeds, all chosen from stable cutting conditions. The networks were trained using mini-batch stochastic gradient descent with approximately 45 000 parameters trained over 50 epochs, with Mean Absolute Error (MAE) as the loss function.

Each of the test scenarios were evaluated by comparing the MMD score between the two machine data sets, namely, the source data set and the target data set of each machine. Traditional machine learning algorithms are compared using the accuracy score, which is derived by adding the true positives and true negatives, divided by all classifications (i.e. total of true positives, true negatives, false positives and false negatives). This score provides an understanding of how the MMD score can relate to the accuracy of the network, and provides further insight into how the similarities between systems impacts the accuracy of a network, prior to training. The accuracy was determined by establishing a ground truth using the methods described in [21] and [22].

The test cases performed involved the following scenarios:

1. Train source network on Tongtai TMV-720A and transfer for Hurco VMX42Di
2. Train source network on Hurco VMX42Di and transfer to Hurco VMX42i
3. Train source network on Hurco VMX42Di and transfer to Hurco VMX6030
4. Train network on Hurvo VMX42Di on steel and test on aluminum

For the reader's convenience, the output from a pre-trained anomaly detection system has been included in Fig. 2. This system was originally trained with the data available from the Tongtai TMV-720A and had a network accuracy of 95own machine. However, when the same network is applied to data from the Hurco VMX42Di without transfer learning applied, it is completely unable to detect any anomalies.

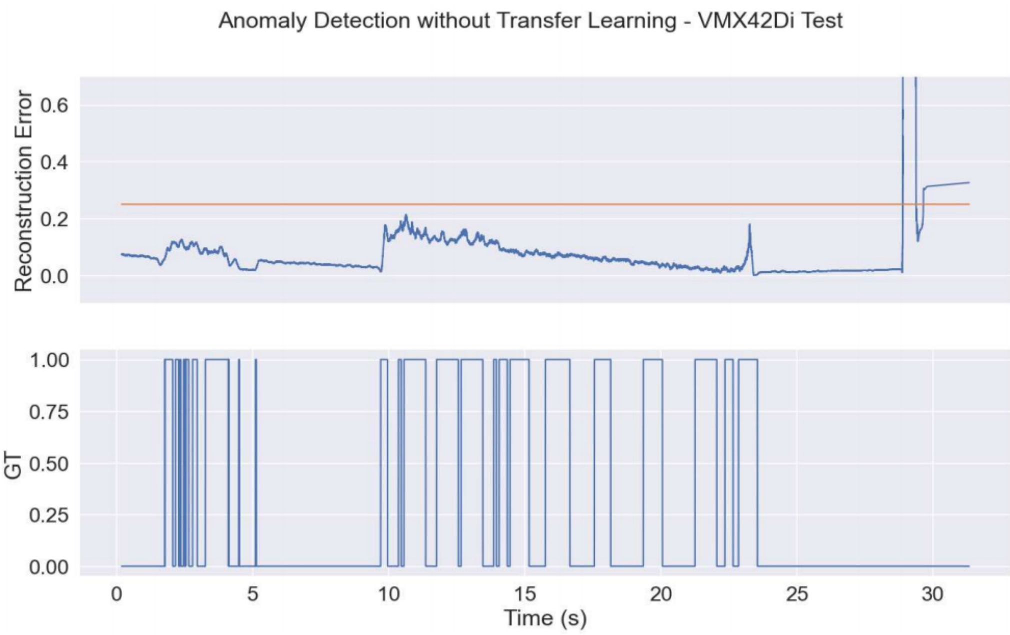

Figure 2. Anomaly detection system output without transfer learning.

3.1. Experimental design

To produce chatter indicator values following the method presented in [21], machining vibration data was collected on a variety of Hurco three-axis CNC machines using a custom-built vibration measurement system consisting of three major components. This system included a pair of uniaxial piezoelectric accelerometers for sensing vibrations, a DAQ for collecting electrical signals from the accelerometers and then transmitting the information via USB, and a computer for running a Python script to produce chatter indicator values.

3.1.1. Vibration measurement system components

The accelerometers used were PCB Piezotronics miniature piezoelectric unixial accelerometers, with one being the 352C22 model and the other being the 352A21 model. Different models were used due to supply issues, but both had the same technical specifications, with ±15sensitivity of 10 mV/g (1.0 mV/(m/s²)), measurement range of ±500 g pk (±4900 m/s² pk), broadband resolution of 0.004 g rms (0.04 m/s² rms), and ±510000 Hz.

Aside from the accelerometers, the DAQ used was a LabJack T7. Analog input (AIN) ports on the LabJack T7 were used to stream input from the piezoelectric accelerometers. Since the maximum sample rate of the LabJack T7 was 100 000 samples per second, it was able to stream input at a suitable speed to the computer running the Python script. The computer used to generate chatter indicator values from collected accelerometer readings was an Asus ROG G551J laptop with a processor base frequency of 2.60 GHz.

3.1.2. Vibration measurement system setup

Due to the need to set up the vibration measurement system on a variety of CNC machines, the DAQ was mounted inside of a plastic control box along with the circuitry required for the piezoelectric accelerometers. This plastic control box had permanent magnets attached to its underside so that it could be quickly mounted onto different CNC machines. The inside of the plastic control box is shown in Fig. 3 below.

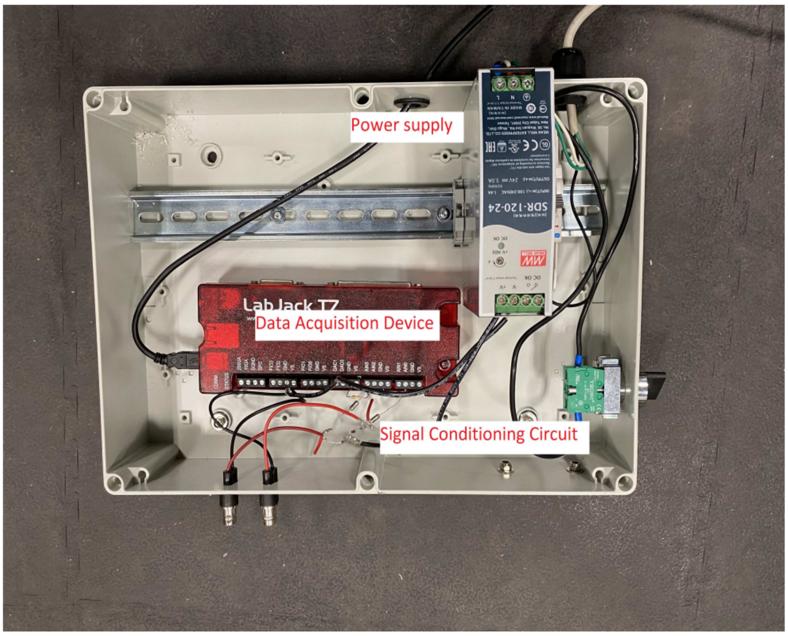

Figure 3. The inside of the plastic control box.

To set up the vibration measurement system on a Hurco machine, the plastic control box housing the DAQ and sensor signal conditioning circuitry was first mounted to the outer metal shell of Hurco three-axis CNC machines using the permanent magnets its back. This is shown in Fig. 4 below.

Figure 4. The plastic control box mounted onto a Hurco VMX42Di CNC machine.

The accelerometers were then threaded through gaps between the CNC machine's head-stock assembly and its outer shell, before either being glued directly to the spindle assembly or being glued to small permanent magnets that were then attached to the spindle assembly. An example of accelerometers set up on a CNC machine is shown in Fig. 5 below.

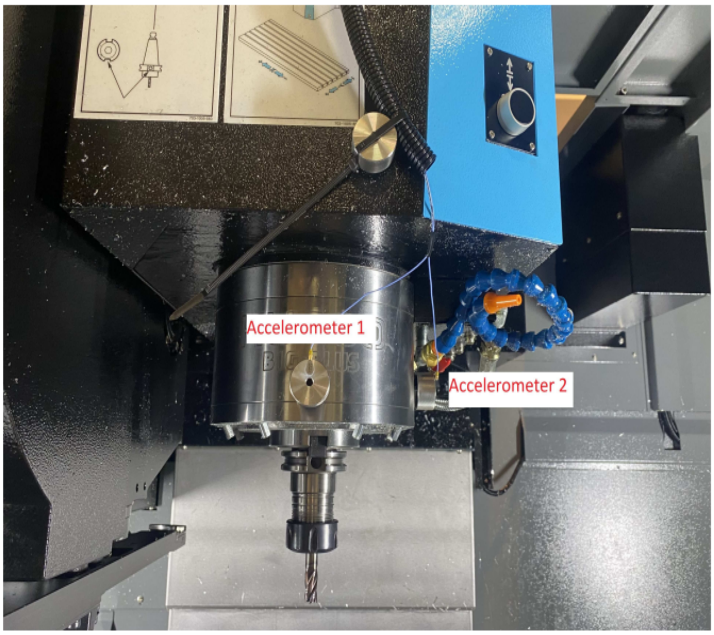

Figure 5. The piezoelectric accelerometers attached to a Hurco VMX42Di CNC machine.

Coaxial cables 10 feet in length were then partially threaded into the machine and used to connect the accelerometers to the plastic control box. The computer for producing chatter indicator values was then placed beside the CNC machine on a cart so that it could be connected to the DAQ within the plastic control box via a USB cable. Finally, an Ethernet cable was used to connect the computer with the CNC machine, this connection allowing for information from the machine's motor drives to control the duration of chatter indicator generation.

The setup process for the vibration measurement system described above took an average of 15 minutes for each machine. To produce chatter indicator values for a machining operation, the feed rate of the CNC machine would have to be manually set to zero and then returned to a non-zero value. This manipulation of the feed rate caused a momentary stop to movement on all axes of the machine, and this specific state was detected by the computer via Ethernet connection and used as a signal to begin generating chatter indicator values. Similarly, chatter indicator value generation was halted once the computer detected that all axes of the machine and its spindle drive had stopped moving, this being a sign that the machining operation had completed. Each completed series of chatter indicator values for a machining operation was saved to the computer as a CSV file, with each chatter indicator value being accompanied with the time of the machining operation at which it was recorded. The CSV files produced by the vibration measurement system were used as the measure for ground truth in determining the efficacy of the transfer learning networks presented in this paper.

3.1.3. Chatter indicator data

The data collected on the Hurco CNC machines were time series of chatter indicator values. To determine these values, the method presented in [21] and [22] was used. This method involved taking the piezoelectric accelerometer values, performing initial filtering to denoise, and then numerically integrating the acceleration time series of the piezoelectric accelerometers to get displacement time series of the headstock assembly in the X and Y directions. After plotting the trajectory of the headstock using these displacement time series, the standard deviation of the headstock's trajectory can be used to estimate the position of the tool in the spindle, and thus the chatter indicator for said tool can be calculated.

Chatter indicator time series were generated for a variety of models of Hurco CNC machines using the vibration measurement system. For each machine, a series of straight cuts through material were programmed. These cuts were performed with standard 0.5 inch end mills and would have varying depths of cut and tool engagement percentages so that periods of stable cutting and unstable cutting would be induced. The material used throughout these tests was standard 1018 steel that was loaded into the CNC machine as a 4.12 inch long, 3 inch wide, and 1 inch tall piece of rectangular stock material.

4. Results

The experiments in the industrial machine shop were completed as detailed in Section. 3.1.. From these experiments, the main measurements taken were the MMD score between the source and target data sets, the minimum learning loss from re-training the network through transfer learning, the cross entropy score, and the network accuracy from testing conditions. These results have been summarized in Table 2 for the results for the tests completed on multiple machines.

Table 2. Comparison of MMD scores and accuracy for experimental test cases on different machines.

Test Description	MMD Score	Learning Loss	Cross Entropy	Network Accuracy
TMV to VMX42Di	55.62	$0.8{\times}10^{-2}$	0.14	61.58%
VMX42Di to VMX42i	15.31	1.33×10-2	0.06	83.00%
VMX42Di to VMX6030	37.59	$20.96{\times}10^{-2}$	0.09	98.03%
VMX42Di on VMX42Di Al	5.67	N/A	0.04	86.39%

From reviewing the table results alone, it is difficult to draw a conclusion as to a relationship between the MMD score, the learning loss, cross-entropy, and the network accuracy. There is no clear correlation between the MMD score, learning loss, or accuracy. We can see that the MMD score follows the same trend as the well-known cross-entropy, but neither has a strong correlation between learning loss and network accuracy. However, when we evaluate the trend between the experiments we see that applying transfer learning between machines from the same manufacturer consistently produces higher accuracy.

To further investigate this trend, we apply the same source input signal amongst the various networks. For example, we apply the acceleration signal seen in Fig. 6 to the network trained with source data sets from the Tongtai TMV-720A, Hurco VMX 42Di, VMX42i and VMX6030 to another Hurco VMX42Di. These networks are each trained using the data sources and targets described in Table. 2 above. The results from these retrained networks are compared to the ground truth and can be seen in Fig. 7.

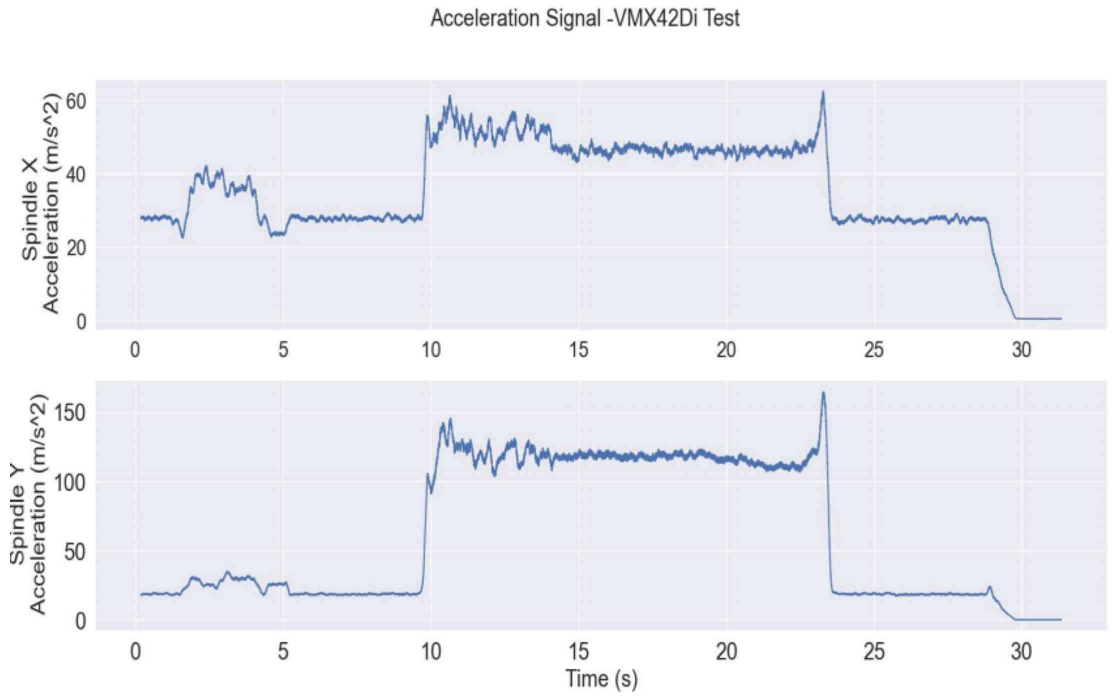

Figure 6. Hurco VMX42Di acceleration signal.

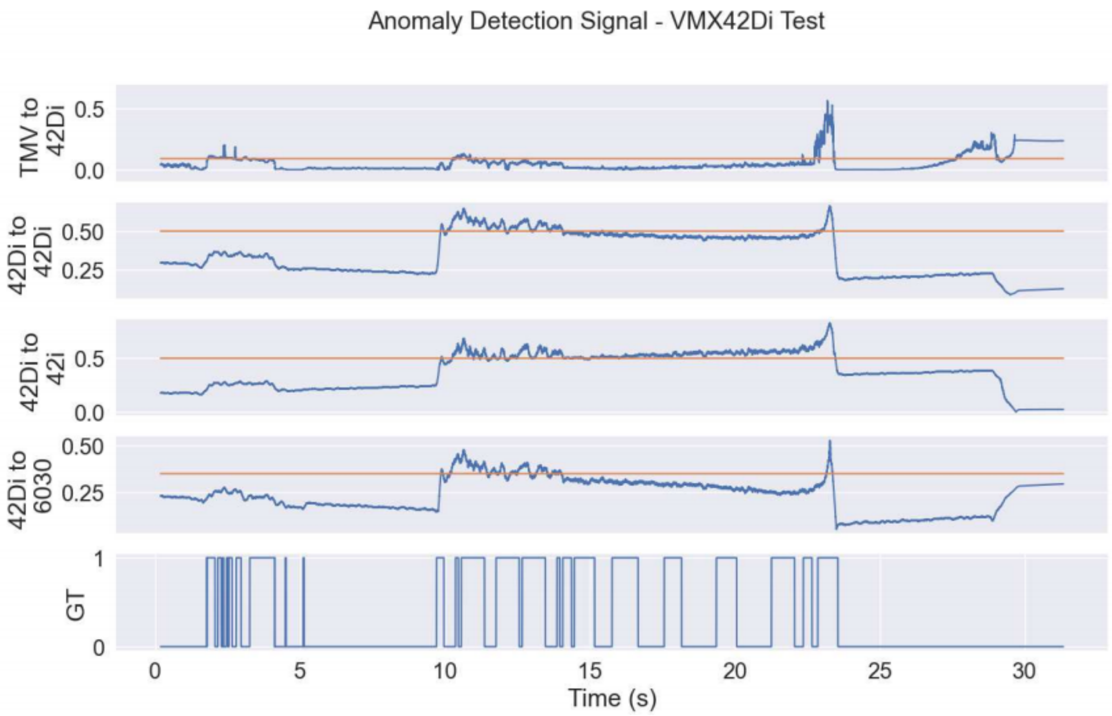

Figure 7. Hurco VMX42Di test.

As the figures show, the ground truth generated using Brecher's chatter indicators align with the expected peaks and valleys seen in the acceleration signal. These points also align with the signals seen for the VMX42Di, VMX42i and VMX6030. Although the network trained on the TMV-720A does pick up some of the similar trends, it does not transfer the knowledge from the original system as well as the other networks designed on data from the same manufacturer. The similarity in response to signals can be noted by visual inspection.

We see a similar result when this test is repeated with acceleration inputs from a cut done on aluminum, seen in Fig. 8, with the results of these networks can be seen in Fig. 9. Since all of the previous networks were trained on steel, one would expect that the networks would have greater difficulty evaluating stability conditions on aluminum. Since aluminum typically has a lower cutting force coefficient, its stability lobes tend to show similar stability conditions as steel for the same depth of cut and RPM machining conditions [23]. This result is confirmed when we evaluate the performance of the various networks seen in Fig. 9. Most notably, the networks with the largest MMD scores, the TMV-720A and the VMX6030, track the input signal and ground truth poorly, while the ones with low MMD scores, the VMX42Di and VMX42i, are able to track the signal, with the VMX42Di tracking the signal the most accurately.

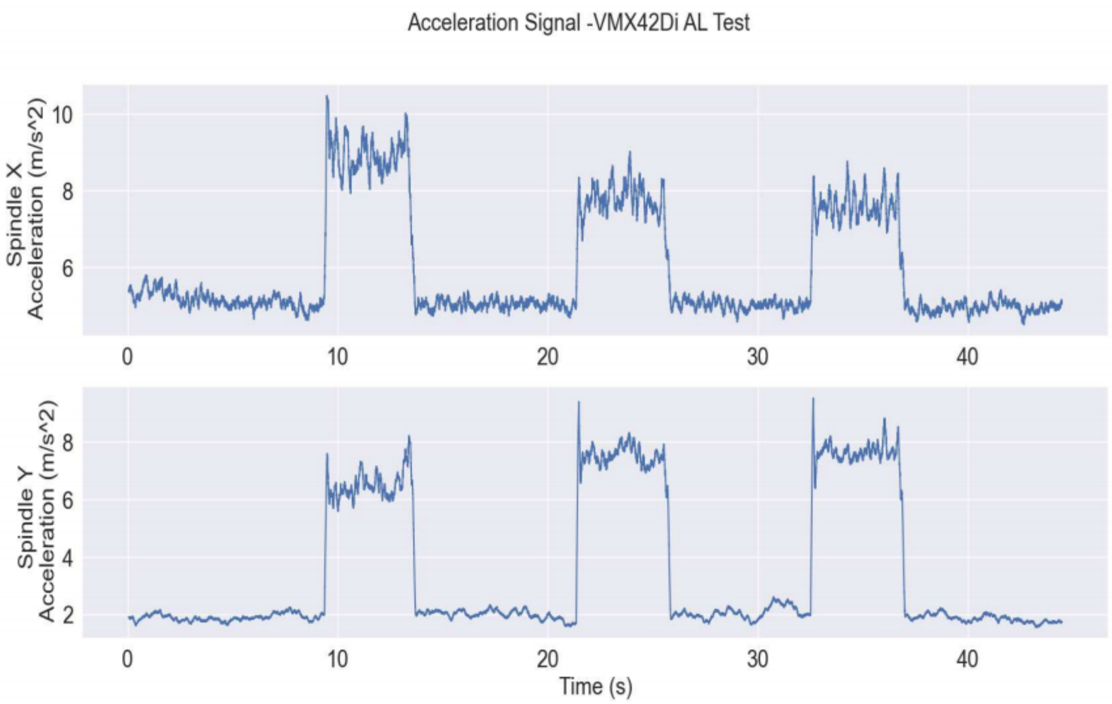

Figure 8. Hurco VMX42Di aluminum acceleration signal.

To evaluate the repeatability of this approach, the same experiment was done on three machines of the same make and model. Data was collected for each machine individually and then the networks were transferred from the VMX42Di to each of the VMX42i, numbered PFC1, PFC2 and PFC3. The summary of results can be seen in Table 3, and the resulting system outputs can be seen in Fig. 10. Consistent with the results seen in the other tests, PFC1 has the largest MMD score and fails to capture some of the peaks seen in the response from the VMX42Di, whereas PFC2 has the lowest score and captures the peaks as well as the relative magnitudes. These results are consistent with the findings from the other experiments.

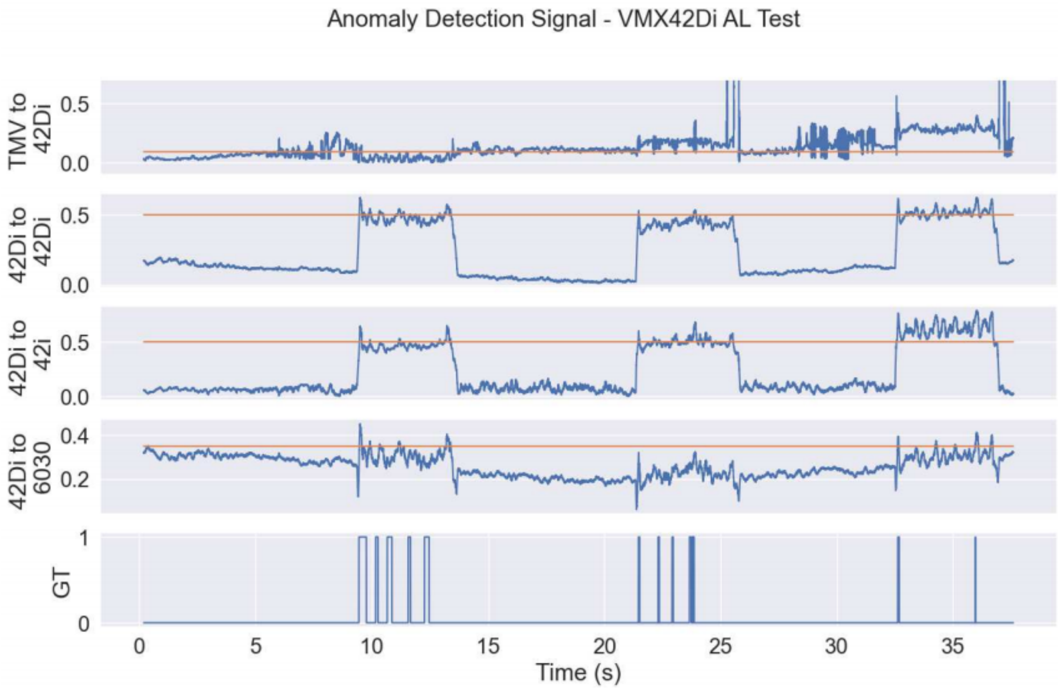

Figure 9. Hurco VMX42Di aluminum test.

Table 3. Comparison of MMD scores and accuracy for experimental test cases on the same model of machine.

Test Description	MMD Score	Learning Loss	Network Accuracy
VMX42Di to VMX42i 1	39.58	4.02×10^{-2}	95.58%
VMX42Di to VMX42i 2	3.53	4.04×10^{-2}	88.08%
VMX42Di to VMX42i 3	22.28	3.04×10^{-2}	97.05%

The results from these experiments match our intuition, where the greater the difference between the source and target sets, the greater the difficulty the network will have in distinguishing the stability of the system. Similar results can be seen for the tests completed on other VMX42i machines and the VMX6030, seen in Fig. 11, Fig. 12, Fig. 13 and Fig. 14. These discrepancies also align with the structure of the mechanistic models, as described in Section .2.. As the mechanistic models show, the cutting parameters representing the tangential and normal forces will vary based on the cutter geometry and material parameters. Thus, we can expect that these differences in geometry and material parameters would also have an impact on the MMD score, and the ability of the network to transfer.

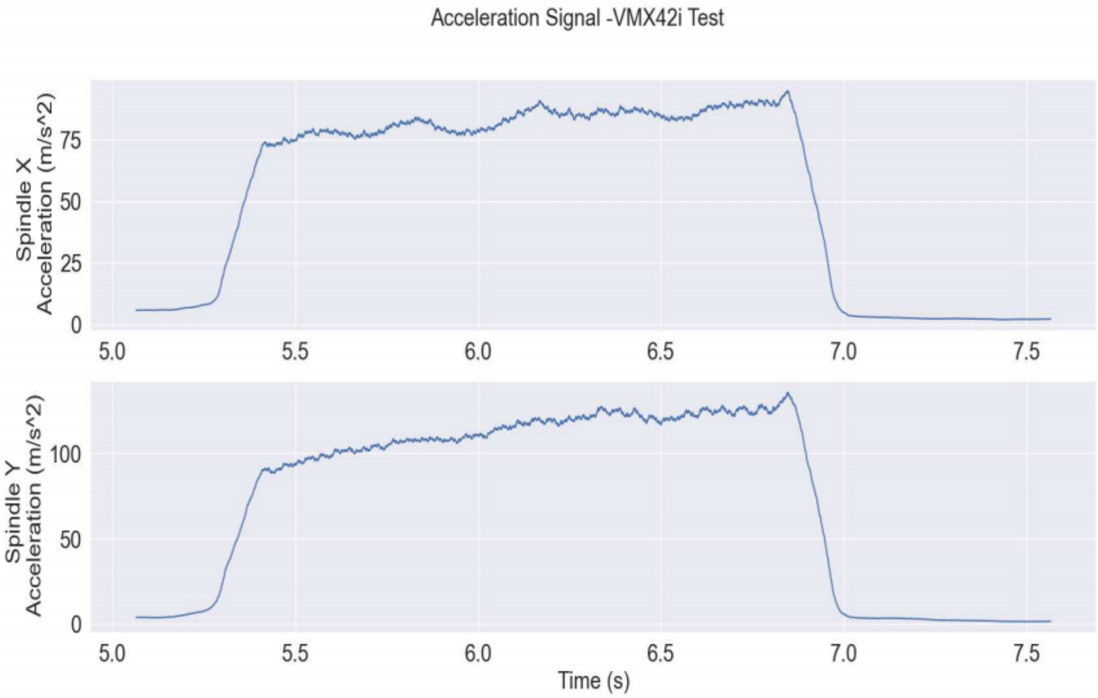

Figure 10. Hurco VMX42i acceleration signal.

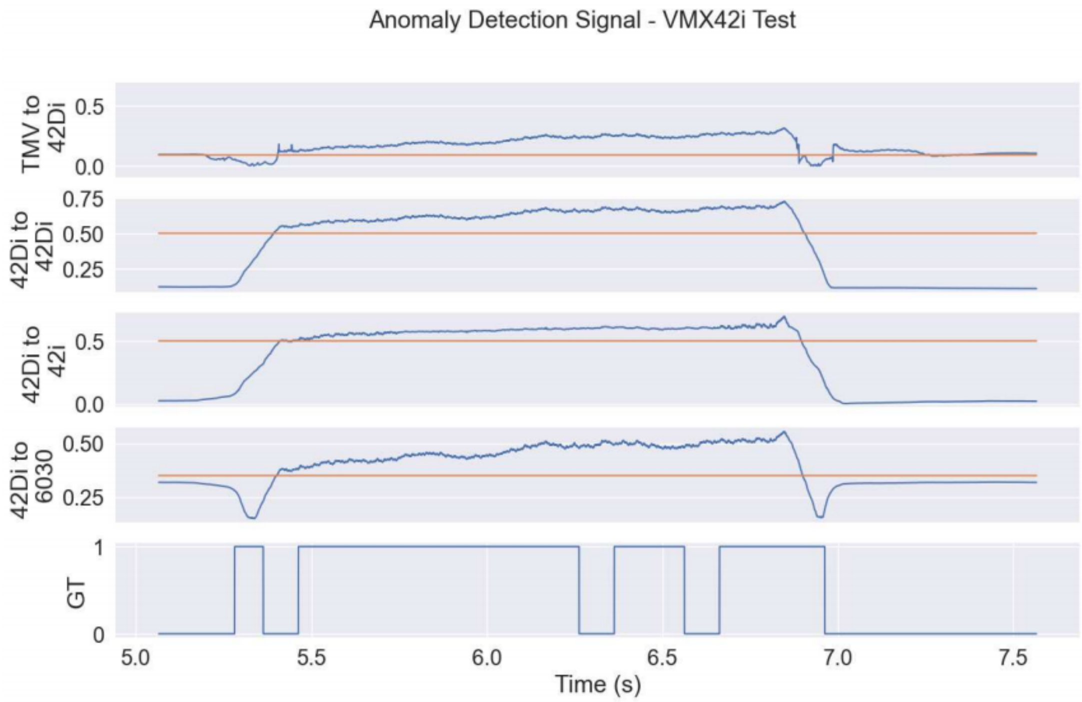

Figure 11. Hurco VMX42i test.

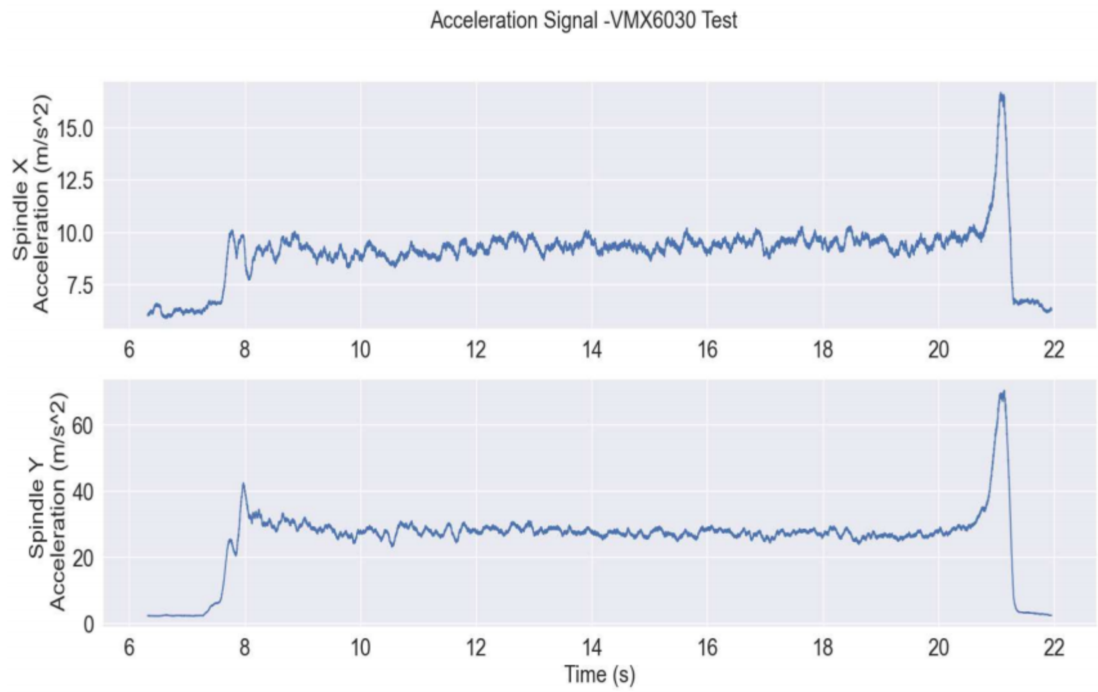

Figure 12. Hurco VMX6030 acceleration signal.

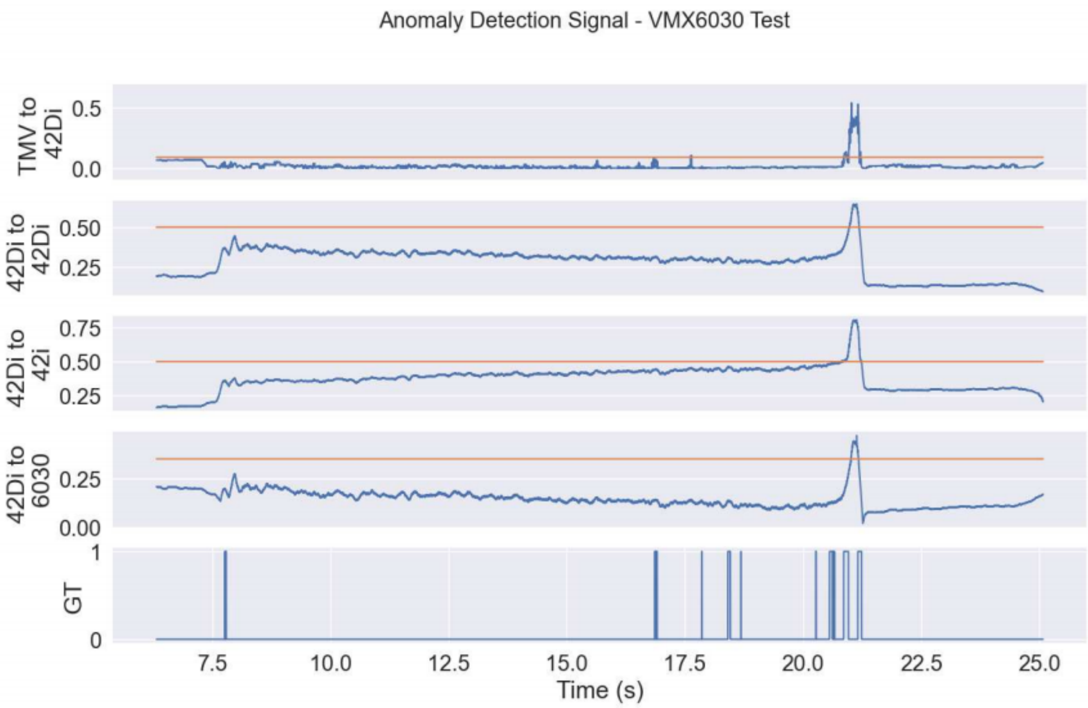

Figure 13. Hurco VMX6030 test.

These results are also consistent when we evaluate the structural differences in the machines. Following the approach detailed in [24], we can evaluate the VMX42Di, VMX42i and

VMX6030 to determine the impact that different structural elements have on the structural dynamics of the various machines, as seen in Fig. 15. Using finite element analysis, the frequency response of VMX42Di was determined in the X, Y, and Z directions to identify the most flexible mode, which influences the strucual dynamocs of the machine through the bending and twisting of the column at 63Hz and 74Hz, with a similar result was seen for the VMX42i in Fig. 16 and for the VMX6030 the column bending can be observed at 61Hz in Fig. 17.

From our analysis, we can compare the spindle drive, column casting, base casting, y-saddle, z-saddle and table configurations used for the various machines. Through this comparison we see that the VMX42Di and VMX42i have the same column casting, base casting, table configuration and y-saddle, but different spindle drives and z-saddle. The FE analysis also shows that the z-column is the most flexible element, thus has the largest impact on the structural dynamics of the machine in the range independent of spindle response. In contrast, when we compare the VMX42Di and the VMX6030, we see that they differ in terms of the spindle drive, column casting, base casting, y-saddle, z-saddle and table configuration, hence the maximum frequency response due to column bending coming at a different frequency and amplitude compared to the VMX42Di and VMX42i and the only similarity between the two machines beings the overall geometry. From this, we conclude that these differences are captured in the data, resulting in larger MMD scores.

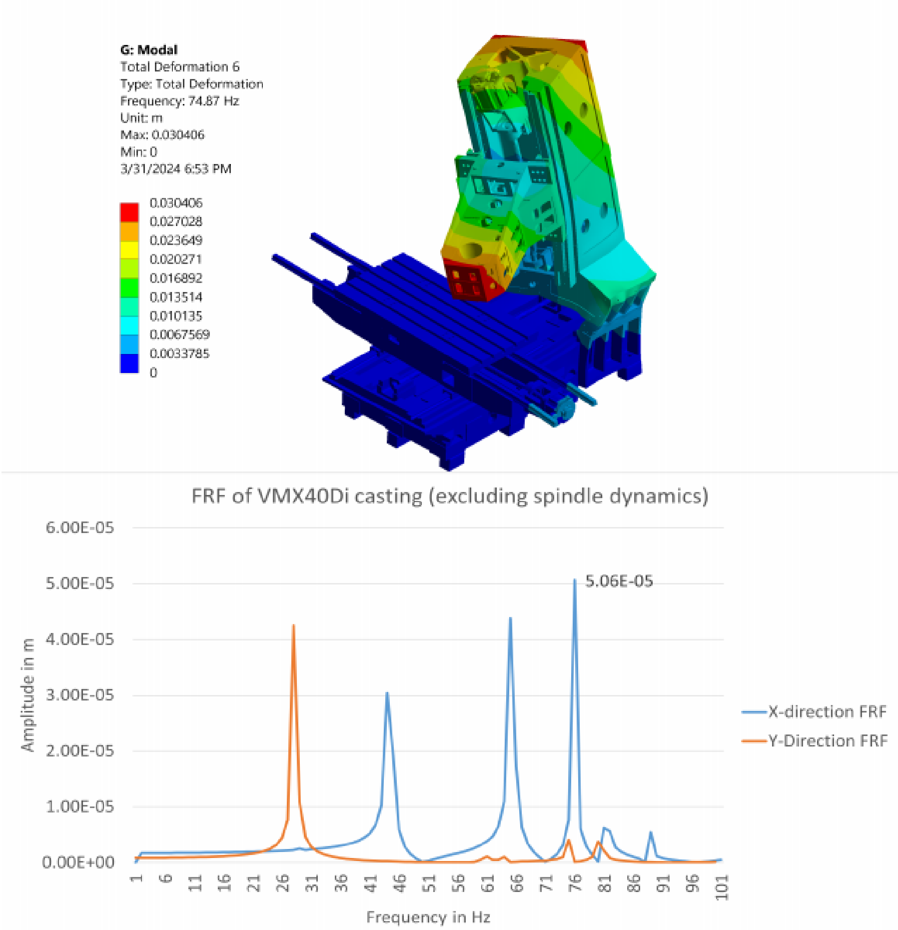

Figure 14. FE analysis of Hurco VMX42Di.

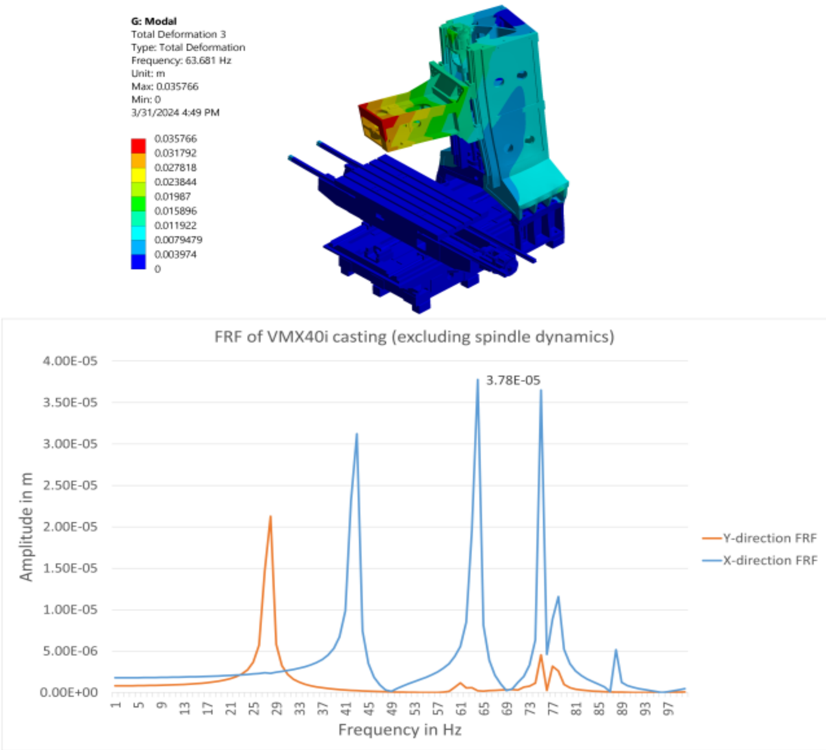

Figure 15. FE analysis of Hurco VMX42i.

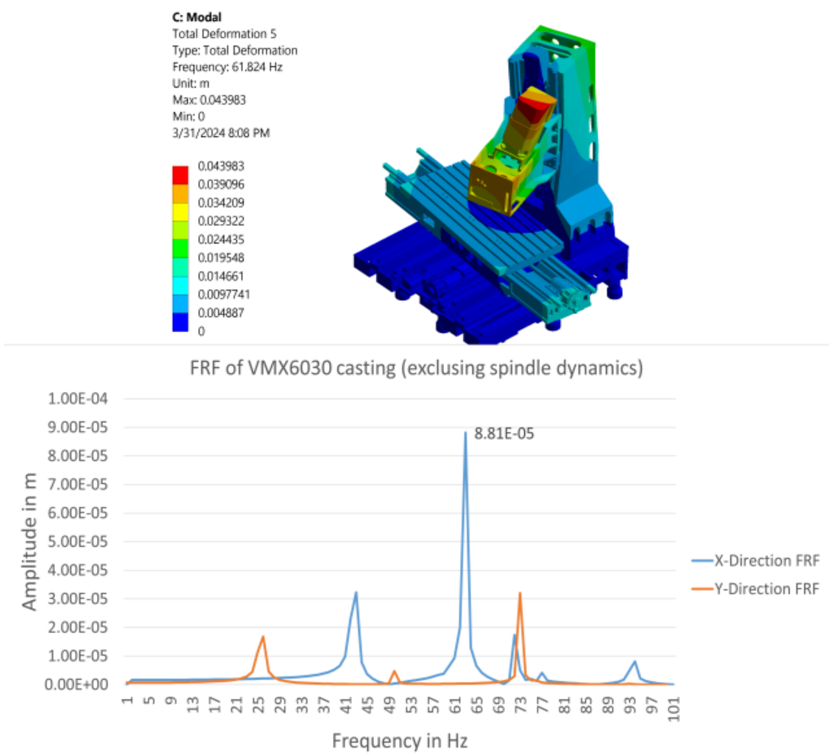

Figure 16. FE analysis of Hurco VMX6030.

Based on the literature, a low MMD score indicates that the source and target training sets have sufficient overlap for transfer learning to occur. From our experiments, we can conclude that in the context of anomaly detection in CNC machines, if the source training data produces a network that is capable of tracking an input signal to detect an anomaly, that the target machine will also produce a signal that outputs similar characteristics. Thus, the physical meaning of a low MMD score is that it indicates that the target machine will track the anomaly signal more closely, but does not necessarily indicate that the accuracy of the network will also be low. A low MMD score not only indicates whether or not transfer learning will be successful but indicates how similar the resulting network output will mimic the source network. A system with a lower MMD score will have an output that responds to the input data in a similar fashion between the source and target networks. However, this similarity does not have any relationship to the ultimate accuracy of the target network. A system with a low MMD score will not guarantee that the target network will be accurate, but will only tell us if the system can be transferred. In addition, these experiments show that networks trained on machines from the same manufacturers are more likely to have a lower MMD score. This is likely due to the fact that the structural dynamics and design criteria are more similar for machines within the same manufacturer than those from other competitors. Lastly, we see that the MMD score does not have a strong relation to the learning loss, and thus does not impact the likelihood of convergence for a system.

5. Conclusion

As the demand for more automated CNC machines continues to grow, the need for more advanced anomaly detection systems also grows. Many of these advanced systems are data-driven and often require significant amounts of data to be obtained to train powerful machine learning algorithms. One solution to this problem is to apply transfer learning to reduce the burden of obtaining large amounts of data. A key indicator for whether or not transfer learning can be successful is the statistical test, the Maximum Mean Discrepancy (MMD) score. From our experiments, we have the following conclusions:

- A low MMD score indicates that a transferred system will respond similarly to the source system
- A low MMD score does not have any correlation with system accuracy
- Networks dervied from data from machines of the same manufactures will likely have lower MMD scores due to similarities in design criteria and structural dynamics

Our experiments show that a low MMD score can be used as a prediction for how similar a source and target network will respond to a given input signal. This allows us to conclude that transfer learning will be most successful within a given family of machines. Furthermore, experiments done on one machine, in one set of conditions, will likely transfer well to another machine of the same or similar model under different operating conditions, but may not transfer as well to a machine with significant differences in its structural dynamics.

Acknowledgments

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflicts of interest

The authors have no relevant financial or non-financial interests to disclose.

Author's contribution

Eugene Li, Sanjeev Bedi and William Melek contributed to the study conception and design. Material preparation, data collection and ground truth generation were performed by Yang Li. The data analysis, algorithm development and the first draft of the manuscript was written by Eugene Li, with input from Yang Li and Sasank Kakarla. All authors commented on previous versions of the manuscript and read and approved the final manuscript.

References

[1] Pierre Cleroux. Reshoring in Canada: What is the economic data telling us | BDC.ca. 2023. URL: https://www.bdc.ca/en/articles-tools/blog/are-canadian-businesses-reshoring-production.
[2] Integrity Data Services Pvt. Ltd. Offshoring vs. Onshoring: Choosing the Right Approach for Your Financial Services. 2023. URL: https://www.linkedin.com/pulse/offshoring-vs-onshoring-choosing-right/.
[3] Li XB, Liu XL, Yue CX, Liang SY, Wang LH. Systematic review on tool breakage monitoring techniques in machining operations. *Int. J. Mach. Tools Manuf.* 2022, 176:103882.
[4] Serin G, Sener B, Ozbayoglu AM, Unver HO. Review of tool condition monitoring in machining and opportunities for deep learning. *Int. J. Adv. Manuf. Tech.* 2020, 109:953–974.
[5] Kuntoğlu M,Aslan A, Pimenov DY,Ali Usca Ü , Salur E, *et al.* A Review of Indirect Tool Condition Monitoring Systems and Decision-Making Methods in Turning: Critical Analysis and Trends. *Sens.* 2021, 21(1):108.
[6] Li WD, Liang YC. Deep transfer learning based diagnosis for machining process lifecycle. Procedia CIRP 2020, 90:642-647.
[7] Li G, Fu Y, Chen DB, Shi LL, Zhou JL. Deep anomaly detection for CNC machine cutting tool using spindle current signals. *Sens.* 2020, 20(17):4896.
[8] Li G, Yang X, Chen DB, Song AX, Fang YK. *IEEE/ACIS 3rd International Conference on Big Data, Cloud Computing, Data Science and Engineering, BCD* 2018, pp. 37–42.
[9] Li E, Bedi S, Melek W. Anomaly detection in three-axis CNC machines using LSTM networks and transfer learning. *Int. J. Adv. Manuf. Tech.* 2023, 127:5185–5198.
[10] Gretton A, Borgwardt KM, Rasch MJ, Scholkopf B, Smola A. A Kernel Two-Sample Test. *JMLR* 2012, 13:723-773.
[11] Gretton A. A Kernel Method for the Two-Sample-Problem. *Neural Information Processing Systems* 2008
[12] Roth DJ. Mechanistic Modelling of 5-Axis Machining. PhD thesis, University of Waterloo, Waterloo, 2004.
[13] Altintas Y. Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design, 2nd ed. Cambridge: Cambridge University Press. 2012
[14] Ahmadi K. Machining Chatter in Flank Milling and Investigation of Process Damping in Surface Generation. PhD thesis, University of Waterloo, Waterloo, 2011.
[15] Tlusty J, Ismail F. Basic Non-Linearity in Machining Chatter. *CIRP Ann-Manuf Techn.* 1981, 30(1): 299–304.
[16] Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Comput.* 1997, 9(8):1735–1780.
[17] Tan CQ, Sun FC, Kong K, Zhang WC, Yang C, et al. *A Survey on Deep Transfer Learning.* In A Survey on Deep Transfer Learning, 3rd ed. Springer 2018, pp.70–279.
[18] A. Burkov. The Hundred-Page Machine Learning Book. Andriy Burkov, 2019.
[19] Bozinovski S. Reminder of the First Paper on Transfer Learning in Neural Networks. *Informatica* 2020, 44(3):291–302.
[20] Wang JC, Zou B, Liu MF, Li YS, Ding HJ, et al. Milling force prediction model based on transfer learning and neural network. *J. Intell. Manuf.* 2021, 32: 947–956.

[21] Brecher C, Chavan P, Epple A. Efficient determination of stability lobe diagrams by in-process varying of spindle speed and cutting depth. Adv. Manuf. 2018, 272–279.
[22] Brecher C, Klimaschka R, Steinert A, Neus S. Efficient determination of stability lobe diagrams deploying an automated, data-based online NC program adaption. *MM Science Journal* 2021, 10:4380-4385.
[23] Schmitz TL, Smith KS. *Machining Dynamics*, 2nd ed. Springer 2019. pp.1-382.
[24] Hung JP, Lai YL, Lin CY, Lo TL. Modeling the machining stability of a vertical milling machine under the influence of the preloaded linear guide. *Int. J. Mach. Tools Manuf.* 2011, 51(9):731–739.