Anomaly Detection in CNC Milling Machines using Transfer and Incremental Ensemble Learning of LSTM Autoencoder Networks

by

Eugene Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2024

© Eugene Li 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Jaho Seo Associate Professor, Dept. of Automotive, and Mechatronics Engineering, Ontario Tech University
Supervisor(s):	Sanjeev Bedi Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo
	William Melek Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo
Internal Member:	Soo Jeon Professor, Dept. of Mechanical and Mechatronics Engineering, University of Waterloo
Internal-External Member:	Stephen Mann Professor, Cheriton School of Computer Science, University of Waterloo
	Alexander Wong Professor, Dept. of Systems Design Engineering Engineering, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Eugene Li was the sole author for Chapters 1, 2, 3, 7 and 8 which were written under the supervision of Dr. Sanjeev Bedi and Dr. William Melek and were not written for publication.

This thesis consists in part of four manuscripts written for publication. Exceptions to sole authorship of material are as follows:

Research presented in Chapter 4 and 5, and 6: This research was conducted at the University of Waterloo by Eugene Li under the supervision of Dr. Sanjeev Bedi and Dr. William Melek, with support from Hurco Companies from Dr. Paul Gray and Sasank Kakarla. Eugene Li designed and implemented the algorithms, with assistance from Dr. Bedi and Dr. Melek. Yang Li provided assistance with data collection and sections of the manuscript related to the data collection. Dr. Paul Gray provided insight into applications in industry and intellectual input. Sasank Kakarla provided expertise on finite element analysis. Eugene Li drafted the manuscript and each author provided input on the manuscript drafts.

Li E, Bedi S, Melek W, Anomaly Detection in Three Axis CNC Machines using LSTM Networks and Transfer Learning, 2023, International Journal of Advanced Manufacturing Technology, 10.1007/s00170-023-11617-5,

Li E, Li Y, Bedi S, Melek W, Gray P, Incremental Learning of LSTM-Autoencoder Anomaly Detection in Three-Axis CNC Machines, 2023, International Journal of Advanced Manufacturing Technology, 10.1007/S00170-023-12713-2/TABLES/2

Li E, Li Y, Kakarla S, Bedi S, Melek W, Transfer Learning Methods for Three-Axis CNC Anomaly Detection, 2024, Journal of Advance Manufacturing, 10.55092/am20240009

Li E, Bedi S, Melek, W, Transfer Learning in CNC Milling Machines for Chatter Detection using LSTM-AutoEncoders, 2024, AIM Conference

As lead author of these four chapters, I was responsible for conceptualizing the algorithms, developing and implementing the approaches, analyzing the data and drafting and submitting manuscripts. My coauthors provided guidance during each step of the research and provided feedback on draft manuscripts, or assisted with data collection or finite element analysis.

I understand that my thesis may be made electronically available to the public.

Abstract

Since the industrial revolution, there has been a steady and continued effort to bring more automation and efficiency to the manufacturing process. Milling machines have long been a valuable tool to create precise parts quickly and effectively. CNC Milling machines have been the next evolution of automation in manufacturing and have allowed for complex parts to be produced quickly and with high precision.

Although CNC milling machines are able to semi-autonomously produce parts effectively, they still require significant human interaction to operate. This human interaction is especially true since many tasks are completed in open loop configuration with little to no feedback. To try to address this issue there has been significant effort in the literature to develop systems to provide feedback to the machine controller. This work is often focused on detecting anomalies such as chatter, broken tools and other conditions that will impact the surface finish or machine health.

A limitation of much of the current work is that it tends to be machine or material specific. These approaches developed in the lab often do not scale well to production as they require custom setups or complex machine dynamics to be studied. To overcome this problem, this thesis proposes a machine learning based solution that leverages deep learning to create a solution that can potentially be quickly and easily transferred to machines in production.

In this thesis we demonstrate that by using simple accelerometers mounted on the spindle of a CNC milling machine, we can create an LSTM-Autoencoder to detect anomalies such as chatter. This feat is accomplished by creating an artificial neural network that is trained on sensor data from stable cutting conditions. This network aims to reproduce the original signal with as little error as possible. If the network is provided data from a stable condition it will reconstruct the signal with little error, but if it is presented data from an anomaly condition it will reconstruct with significant error, which indicates that an anomaly is present.

In Chapter 4 we show that this approach can also be achieved by implementing what is known as transfer learning. In transfer learning we begin with a network that is trained on one source data set, and then transfer the knowledge to another target data set. We investigate under what conditions this is most feasible and demonstrate that we can train a network from a robust data set on one three-axis CNC machine and then transfer it to another three-axis CNC machine. We also demonstrate that this method works for both chatter detection and broken tool detection.

In Chapter 5, we introduce an incremental learning method based on ensemble learning. This approach takes the LSTM-Autoencoder trained previously as a strong learner and has weak learners continually learn as new data is made available. This approach is shown to have comparable results to a large network trained from scratch and improves the performance of a system trained with transfer learning.

Taking these transfer learning and incremental learning algorithms, we extend the approach to anomaly detection for five axis CNC milling machines in Chapter 6. This is accomplished by introducing a stacked ensemble learning approach by transferring the encoder from the three axis CNC anomaly detection algorithm and then combining it with an encoder and decoder that is trained from the target data. Incremental learning is then integrated by adding weak learners to this strong learner. These weak learners allow the network the ability to improve the performance of the system to be comparable to a network trained from scratch with a fraction of the data.

Lastly, we demonstrate how these approaches can be used for multi-class prediction in Chapter 6. In this chapter, we use the LSTM-Autoencoder to perform dimensionality reduction. We then use this dimensionality reduced output and apply a one-versus-all SVM classifier and Platt scaling to obtain a probabilistic prediction of the classes of interest. This approach allows us the ability to both detect and differentiate cutting with broken tools and chatter conditions.

The approaches presented in this thesis demonstrate that this proposed approach is capable of not only detecting chatter in a specific lab setting, but can potentially be used to detect multiple anomalies across a variety of machines and materials. This allows users to potentially scale these approaches to many machines quickly with minimal setup and minimal configuration.

Acknowledgments

I would like to thank all the many people who made this thesis possible.

I would like to thank my advisors Sanjeev Bedi and William Melek. I have known you both for a long time, and in many capacities, but I continue to learn things from you every day. I could not have made it to this point in my career without you.

I would like to thank all the co-op students who helped with the preparation for the experimental data collection including Eric Jessee, Sarah Ahmed, Krish Patel, and, most of all, Yang Li.

I would also like to thank the Pearl Sullivan Engineering IDEAs Clinic for allowing me to use their computing resources to train the networks, and Dr. Paul Gray and all of the engineers at Hurco Companies Inc for their generous support and access to machines to complete the experimental work.

I would like to thank Keona Pang for helping me review and edit my thesis.

I would like to thank my family for their understanding and support while I completed my work.

Most importantly, I would like to thank my wife, Alison Shaw, for her understanding and patience, and for helping me with vector images.

Dedication

This thesis is dedicated to the memory of my father Joseph Li, who encouraged me from a young age to ask questions and be curious. You prepared and supported me for the journeys and challenges of life and I only wish I could share this with you.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgments vii

Dedication viii

List of Figures xiii

List of Tables xvii

1 Introduction 1

1.1 Motivation 1

1.2 Problem statement 4

1.3 Thesis objective 5

1.4 Proposed contributions 5

1.5 Thesis organization 6

2 Background and literature review **8**
2.1 Background . . . 8
2.2 Literature review . . . 9
2.2.1 Mechanistic modeling based anomaly detection . . . 9
2.2.2 Sensor based methods of anomaly detection . . . 12
2.2.3 Data driven methods for anomaly detection . . . 16
2.3 Preliminaries . . . 20
2.3.1 Recurrent Neural Networks . . . 20
2.3.2 LSTM Autoencoder . . . 23
2.3.3 Data augmentation of training data for machine learning . . . 25
2.3.4 Ensemble learning . . . 26

3 Anomaly Detection in CNC Milling Machines using Sensors and Machine Learning **28**
3.1 Motivation for anomaly detection algorithm in CNC milling machines . . . 28
3.2 Sensors used for anomaly detection in CNC machines . . . 29
3.3 Machine learning algorithm for anomaly detection in CNC Machines . . . 31
3.4 Experimental implementation of anomaly detection algorithm . . . 33
3.5 Discussion and summary of contributions . . . 37

4 Transfer Learning for Anomaly Detection in CNC Machining **38**
4.1 Motivation . . . 38
4.2 Transfer learning algorithm mathematical definition . . . 39
4.2.1 Maximum Mean Discrepancy and transfer learning . . . 41
4.3 Transfer learning experimental implementation . . . 42
4.3.1 Closed loop implementation of anomaly detection using transfer learning . . . 47
4.3.2 Anomaly detection of broken tool bit using transfer learning . . . 51
4.3.3 Meaning of Maximum Mean Discrepancy in relation CNC Machining 55
4.4 Discussion and summary of contributions . . . 67

5 Incremental Learning for Anomaly Detection in CNC Machining 70
5.1 Motivation . . . 70
5.2 Incremental learning algorithm definition . . . 71
5.2.1 Incremental learning problem formulation . . . 72
5.2.2 Incremental learning algorithm description . . . 73
5.2.3 Incremental learning algorithm mathematical representation . . . 74
5.3 Incremental learning experimental implementation . . . 76
5.4 Discussion and summary of contributions . . . 82

6 Multi-Axis Anomaly Detection with Transfer and Ensemble Learning 84
6.1 Motivation . . . 84
6.2 Multi-axis anomaly detection algorithm using transfer and ensemble learning 86
6.2.1 Incremental Ensemble Learning for multi-axis anomaly detection . . 88
6.3 Experimental implementation . . . 90
6.4 Discussion and summary of contributions . . . 96

7 Multi-Class Anomaly Prediction 97
7.1 Motivation . . . 97
7.2 Multi-class anomaly prediction algorithm . . . 98
7.2.1 Dimensionality Reduction using a LSTM-Autoencoder . . . 99
7.2.2 Multi-class classification with Linear SVC . . . 100
7.2.3 Multi-class label prediction with Linear SVC . . . 102
7.3 Multi-Class anomaly prediction with transfer learning . . . 104
7.3.1 Experimental implementation of multi-class anomaly prediction . . 104
7.4 Discussion and summary of contributions . . . 106

8 Conclusions and Future Work 108
8.1 Future Work . . . 110

References 112

APPENDICES 135

A Network design for anomaly detection systems 136

List of Figures

2.1 Resistor model of cutting tool used in CNC machining 10
2.2 Example of traditional stability lobe [184] 12
2.3 Example of accelerometer signal from stable CNC cutting conditions 13
2.4 Example of accelerometer signal from chattering CNC cutting conditions . 14
2.5 Example Frequency Analysis from stable and unstable CNC cutting conditions 15
2.6 Example of WPD signal from stable CNC cutting conditions 16
2.7 Example of WPD signal from chattering CNC cutting conditions 17
2.8 Recurrent Neural Network architecture 20
2.9 LSTM Cell layout 22
2.10 LSTM Autoencoder Architecture 24
2.11 Histogram of reconstruction loss from LSTM Autoencoder 25

3.1 Piezoelectric accelerometers attached to a Hurco VMX42Di CNC machine. 31
3.2 Autoencoder structure for LSTM Autoencoder for anomaly detection . . . 34
3.3 Test bench on X-Carve Desktop CNC 35
3.4 Test results for anomaly detection Autoencoder with small sample set . . . 36

4.1 Data flow for transfer learning approach 40
4.2 Two-Dimensional visualization of MMD metric 42
4.3 Tongtai TMV-720A setup [215] 43
4.4 Roland MDX-40A experimental setup 44

4.5 Sample Acceleration Signal 45
4.6 Reconstruction Plot generated without Transfer Learning 46
4.7 Reconstruction Plot generated from Transfer Learning 47
4.8 Chatter Indicator Comparison 47
4.9 Accelerometer signal from closed loop cutting experiment 49
4.10 Accelerometer signal with anomaly detection indicators 50
4.11 Plot of spindle speed (RPM) and anomaly detection indicator 51
4.12 Comparison between new end mill and damaged end mill 52
4.13 Algorithm output with new tool bit at 7000 RPM 52
4.14 Algorithm output with broken tool bit at 7000 RPM 53
4.15 Algorithm output with new tool bit at 4500 RPM 54
4.16 Algorithm output with broken tool bit at 4500 RPM 54
4.17 The inside of the plastic control box. 56
4.18 The plastic control box mounted onto a Hurco VMX42Di CNC machine. . 57
4.19 Hurco VMX42Di Acceleration Signal 58
4.20 Hurco VMX42Di Test comparing transfer learning from different sources . 58
4.21 Hurco VMX42Di Aluminum Acceleration Signal 60
4.22 Hurco VMX42Di Aluminum Test comparing transfer learning from different sources 61
4.23 Hurco VMX42i Comparison of transfer learning to the same model of machine 62
4.24 Hurco VMX42i Acceleration Signal 62
4.25 Hurco VMX42i Test comparing transfer learning from different sources . . 63
4.26 Hurco VMX6030 Acceleration Signal 63
4.27 Hurco VMX6030 Test comparing transfer learning from different sources . 64
4.28 FE Analysis of Hurco VMX42Di 65
4.29 FE Analysis of Hurco VMX42i 66
4.30 FE Analysis of Hurco VMX6030 67

5.1 Network Structure with Strong and Weak Learner 75
5.2 Reconstruction Loss with Varying Numbers of Ensemble Learners 77
5.3 Acceleration from test signal used for testing incremental learning algorithm 78
5.4 Ground truth from test signal used for testing incremental learning algorithm 79
5.5 Reconstruction Loss from directly trained network and ensemble-inc network 80
5.6 Reconstruction Loss from transfer learning network and ensemble-inc network 80

6.1 Hurco VMXSRTi showing the B-Axis Spindle and C-Axis Rotary Table[138] 85
6.2 Block diagram of five axis ensemble learning approach 86
6.3 Typical stacking ensemble learning process flow 87
6.4 Block diagram for five-axis ensemble incremental learning architecture . . . 89
6.5 Experimental Setup on Hurco VMX42SRTi 90
6.6 Solidworks model of ramp cut completed in XYZ 91
6.7 Experimental results of ramp cut completed in XYZ 91
6.8 Frequency response from ramp cut completed on five axis CNC 92
6.9 Example Accelerometer Signal for Five Axis Machine 93
6.10 Anomaly Detection signal from directly trained network compared to Ground Truth 94
6.11 Anomaly Detection signal from Transfer Learning compared to Ground Truth 94
6.12 Anomaly Detection signal from Transfer Learning and Ensemble Incremental Learning compared to Ground Truth 95

7.1 Example Cutting Signal Containing Stable, Broken and Chatter Cutting Conditions 98
7.2 Feature Mapping of Cutting Conditions through dimensionality reduction via Autoencoder 100
7.3 SVC with RBF Kernel and SVC with Polynomial 101
7.4 Linear SVC classifier with class boundaries for mapped features 102
7.5 Predicted Class Probabilities based on input signal 103

7.6 Feature Mapping of Cutting Conditions through dimensionality reduction via Autoencoder trained with transfer learning 105
7.7 Linear SVC classifier with class boundaries for mapped features with Autoencoder trained with transfer learning 106
7.8 Linear SVC classifier with class boundaries for mapped features 107

A.1 Network Configuration for Five Axis Anomaly Detection 137
A.2 Network Configuration for Five Axis Anomaly Detection with Ensemble Incremental Learning 138
A.3 Network architecture for dimensionality reduction of multi-class network . 139

List of Tables

2.1 Variables used in LSTM 23

3.1 Typical sensors used in CNC milling operations 30

4.1 Confusion Matrix for Anomaly Detection without Transfer Learning 48

4.2 Confusion Matrix for Anomaly Detection using Transfer Learning 48

4.3 Comparison of MMD Scores and Accuracy for Experimental Test Cases on Different Machines 59

4.4 Comparison of MMD Scores and Accuracy for Experimental Test Cases on the same model of machine 61

4.5 Comparison of network training parameters with and without transfer learning 69

5.1 Accuracy Comparison of Different Machine Learning Architectures 81

6.1 Number of weak learners vs learning loss for multi-axis anomaly detection . 89

6.2 Comparison of network performance metrics 95

Chapter 1

Introduction

1.1 Motivation

Since the beginning of the industrial revolution there has been a steady effort to increase the autonomy in the manufacturing environment. Increased autonomy can lead to higher productivity and increased operator safety [37]. This automation has led to significant advances in manufacturing, and significant impact on the manufacturing industry [66].

These advances in manufacturing ability were accompanied by increased globalization and reliance on off-shore manufacturing in other countries. Although this approach of globalization served North American manufacturing industries well for decades, in recent years there has been an impetus to bring more of these manufacturing capabilities back to North America, known as onshoring. This push has been exasperated by recent supply chain issues, delays, price hikes and export/import restrictions [177] [24]. However, high labor costs and a growing skills shortage are often a barrier to effective onshoring of manufacturing [67]. To address this gap, there has been an increased effort to automate more manufacturing processes to compensate for the lack of experienced affordable operators in the last few decades.

In addition to these onshoring efforts, there has been renewed interest in developing a "lights-out" manufacturing facility. First coined in the 1980s, in lights-out manufacturing the machines run completely autonomously without any human intervention, facilitating the idea of a factory running without need for lighting for humans [16]. This ideal would allow for reduced accidents and increased worker safety, improved operational efficiency, reduced scrap and rework and allow for more advanced maintenance strategies [152]. In the

1980s, this work was envisioned as having automatic part loading and unloading, with the same part manufactured over and over. This goal has been recently accomplished with the introduction of collaborative robots, known as cobots, and other palletization techniques [151]. These machine tending technologies have increased flexibility, throughput, system performance while reducing operation costs and improving employee safety and satisfaction [39]

Although these new technologies and systems do solve many issues, not all of the problems preventing a fully autonomous manufacturing facility have been solved. Furthermore, the introduction of these connected systems provides opportunities for greater innovation beyond simply loading and removing parts. One limitation in our current manufacturing systems is that they do not know how to react when deviations from normal operating conditions occur. Therefore, the occurrence of deviations, or risk of deviations, limits the machines from truly running without human interaction. To achieve this lofty goal of a machine that can run without any human intervention, we need to develop smarter machines that are not only capable of achieving the desired performance that we demand of them, but are also able to adapt to changing demands such as the presence of anomalies or the wear of tools overtime. This is required to ensure that a safe and continuous operating environment can be maintained without human intervention. Without this ability, machines running autonomously may produce unacceptable parts or damage itself or its surroundings; at worst, others in the surrounding area may be injured.

In many manufacturing environments, CNC milling machines are used to produce precise parts. Once these milling machines have been configured, they can be run with little human intervention [16]. However, a limitation for these machines is that they run in an open-loop configuration, namely, there is no built-in feedback to the machine when deviations occur. Although some machines have rudimentary feedback, such as spindle current detection, many of them are unable to obtain quality measurements of the cuts themselves. This is a limitation of many production machines, as they are unable to detect or react to anomalies when they occur [59]. To address this issue, there has been significant effort in the literature to develop different algorithms and methods to detect anomalies during the machining process. These methods have ranged from model based methods, described in detail in seminal works such as [5], to sensor based methods [35][89][124][42][150][223], and recently to data based approaches [52][62][6][21][77][126][221].

Although these approaches have shown considerable success in lab environments, they have not been implemented widely in production. The methods developed in labs are for particular machines, materials or machining operation such as milling, drilling and more. Unless the specific machines, materials and operations are used in production, the users are unable to achieve the same results as those observed in the lab. To achieve the

benefits seen in lab setups the users have to invest significant time and capital recreating the systems on the production floor. Furthermore, this approach requires all machines to be instrumented, and the modeling procedures repeated to generate the specific models required. This process is both expensive and time consuming. As a result of this, many operators run the machines in overly conservative modes, since they do not have an accurate model of their specific machine.

In recent years there have been significant advances in machine learning algorithms. A day-to-day example of this is the automatic translation capabilities of many smartphones and computers. These algorithms work by analyzing the sequences of words and looking for patterns and then translating these patterns to a target language. This is achieved by using neural networks with two parts, an encoder and decoder, and sequence-to-sequence learning with networks such as a Recurrent Neural Network (RNN) [15]. The encoder portion of these networks attempt to capture the meaning of the sequence in a numerical representation, abstracted from the original input into a latent representation space. The decoder, then translates this numerical representation to a target system. The abstraction behind the translation system bears similarity to the scalability challenges faced in CNC machines. These concepts can be used in manufacturing environments to abstract the information captured from one machine to a numerical representation, and then translate this generic model to a specific machine. This approach will allow for a general model to be developed and quickly moved to another system. Thus, these machine learning methods have the potential to overcome the limitations of developing expensive machine specific models.

Motivated by Industry 4.0, machine learning algorithms have been used to overcome limitations in modeling accuracy and to implement new approaches to analyze and predict the output of machining operations. These methods have shown significant promise [94] [3], and many in industry are keen to take advantage of these approaches. Although effective, these algorithms require a significant amount of data to be collected on the specific machines of interest to train machine learning algorithms capable of obtaining results useful in practice and comparable to those in the literature. This requires a substantial investment, as time and resources must be spent obtaining the data and instrumenting machines with expensive sensors, rather than producing useful parts; ultimately limiting the implementation and adoption of these technologies outside of lab environments.

Nevertheless, machine learning is an in-demand and emerging field, not only in manufacturing but in many domains as well. This has resulted in a wide variety of new and advanced machine learning algorithms being developed. Some of these new algorithms, namely techniques such as transfer learning, ensemble learning and incremental learning, have changed the way that many machine learning algorithms work. These algorithms have

shifted the focus from obtaining data from a single application, to developing a connected and evolving framework that leverages data from a network of machines for training. This change in approach offers an opportunity for developing a black box model that can be applied on any machine in general, without expert knowledge of the specific machine. In addition, this change in methodology aligns well with the changes in Industry 4.0 technology, which have created new interconnected manufacturing facilities, allowing data to be collected and shared quickly. This evolution of technology presents a unique opportunity that takes advantage of these new algorithms in an industrial setting to address the shortcomings of the approaches described previously, allowing anomaly detection systems to be developed which are scalable to production environments without the need for extensive experimentation and expensive experimental setups. This methodology has the potential to be applied to CNC machining, if the problem is properly studied.

1.2 Problem statement

There is a desire to have smart CNC machines, capable of autonomously producing parts with little to no human intervention. To achieve this goal, we must design machines with systems that allow them to not only produce accurate parts, but to detect anomalies as well. These systems should not be machine or material specific, less they be limited to lab environments. These systems also should not require complex or expensive equipment, to ensure that they have the greatest reach possible. Only by designing systems with these criteria in mind, can we make progress towards the goal of a smart CNC machine.

To achieve this goal, we have identified, what we believe, to be four key issues that are preventing these smart CNC machines from being implemented in production. These issues are as follows

- Implementing machine learning algorithms requires substantial training data before they can be used. This is often impractical or expensive for end users and those in production settings, leading to low adaptation.
- Models trained for one instance of time do not adapt to changes in machine dynamics, especially as a machine ages. This leads to models not accurately reflecting the scenarios they are operating in and being ignored.
- The majority of research is focused only on three-axis CNC milling machines. Although these are the largest segment of machines, five-axis CNC milling machines represent a neglected area of the literature.

- Anomaly detection alone is useful, but given the wide variety of potential anomalies, and the different recovery formats, it is helpful to know what the anomalies are. For example, the recovery method for chatter (change in spindle speed), is vastly different from the recovery method for a broken tool (changing the tool).

1.3 Thesis objective

The main objective of this thesis is to present an approach that can be used to develop a system that can be implemented across a large spectrum of CNC machines and cutting materials by creating a robust Autoencoder anomaly detection neural network on one machine, and then transferring it to another machine via transfer learning. The network can then be extended to add enhanced capabilities such as incremental learning, multi-axis anomaly detection or multi-class anomaly prediction capabilities through the use of ensemble learning. This allows advanced technologies to be deployed quickly with a fraction of the data required and is scalable to a wide variety of CNC milling machines with minimal effort from the operator.

1.4 Proposed contributions

This thesis identifies four potential problems and four potential solutions to these problems. Each of these proposed solutions addresses their respective problem described above. These solutions provide contributions to the literature, as they represent new methods to solve these potential problems. These contributions will be explored in detail in the subsequent chapters, but are presented here in brief.

- Development of methods and machine learning parameters for applying transfer learning for LSTM anomaly detection in CNC milling machines: In this work we identify the key conditions necessary to apply transfer learning for deep networks such as the LSTM-Autoencoder. This involves identifying network parameters, such as network size and architecture, methods and relationships between the statistical measures and their physical meaning, to effectively train an anomaly detection system using transfer learning.
- Development of a incremental learning method for anomaly detection in LSTM-Autoencoders : In this work we develop an extension to train the anomaly detection

algorithm through incremental learning. We then demonstrate that this approach significantly improves the accuracy of the system trained via transfer learning.

- Development of transfer and ensemble learning methods for multi-axis anomaly detection in CNC milling machines: In this work we show how transfer and ensemble learning methods can be extended for anomaly detection beyond simply three-axis CNC machines to multi-axis CNC machines. We demonstrate this work on a five-axis CNC machine and show how incremental learning can be applied to significantly improve the accuracy of this system.

- Development of a multi-class prediction method using Autoencoder dimensionality reduction: In this work we show how the LSTM-Autoencoder designed previously can be modified to not only detect anomalies, but can be used to predict multi-class labels via dimensionality reduction. This work is also extended to systems trained via transfer learning.

1.5 Thesis organization

This thesis is divided into three main parts, a description of an anomaly detection method developed on a training data from a single source obtain from a sole CNC machine, machine learning techniques for anomaly detection developed using data from a variety of machines and sources, and then extensions of these techniques to more advanced applications. This structure allows the reader to understand the current state of techniques and the contributions that this thesis makes towards the literature.

The first part of the thesis provides the reader with an understanding of the anomaly detection methods proposed and a comparison of different network and sensors options within a limited scope. In Chapter 2 the reader is provided with the background and motivation to the problem at hand, as well as providing a cursory overview of the common approaches currently being applied and their limitations, and some preliminary information. Chapter 3 provides an in-depth discussion to the key anomaly detection algorithm implemented in this thesis, as well as a discussion on the sensing approaches applied.

The second part of this thesis describes the machine learning methods developed and applied in this thesis. In Chapter 4 we introduce the concept of transfer learning and demonstrate how it can be applied to connect machines to reduce the training data required to bring an anomaly detection methods to industry. In Chapter 5, we discuss incremental learning and how it, combined with transfer learning provide significant power for anomaly detection in CNC milling machines.

The third part of this thesis investigates some potential applications of this transfer and ensemble learning approach. In Chapter 6 we explore how this framework allows users to leverage the algorithms developed for three axis CNC milling machines to higher dimensions such as four or five axis milling machines. In Chapter 7, we show how the system can be extended beyond simple anomaly detection, but can be applied to multi-class anomaly prediction as well. This is concluded in Chapter 8 where conclusions are presented and potential future work is discussed.

Chapter 2

Background and literature review

2.1 Background

Beginning in the late 1700s, milling machines were introduced as a way to avoid manual filing and cutting with hand tools [174]. Prior to this point, the production and manufacture of parts was slow and labor intensive. Thus, from its inception, milling machines have been seen as a way to effectively reduce time and effort spent in the production of parts, allowing more parts to be made and for greater production overall.

This continual development of milling machines lead to the production of CNC milling machines in the 1960s. This computerization allows for significantly more automation to be implemented and has led to reduced human oversight in the production of high quality components. However, this automation is not without its own limitations.

The current architecture of CNC milling machines allows for precise control of the cutting tool of a machine. This control is accomplished by advanced modeling of the machine and motor dynamics along with the implementation of sophisticated control algorithms. Although this approach is effective at producing parts, this architecture does not allow the machine to detect or react to anomalies that may occur during the machining process. Without the ability to detect or react to anomalies, the machine may encounter an anomaly and will attempt to continue its cutting operations. Ignoring the anomaly often results in unacceptable and wasted parts being produced, or significant damage to the machine or its surroundings. To avoid this problem, operators are often required to watch the machines to catch the anomalies that may or may not occur, which results in wasted labor and lower productivity.

To address the limitation of the current architecture, there has been significant effort within the research community to try to detect, predict and react to these anomalies. In this thesis, these approaches have been classified into three classes: mechanistic modeling based approaches, sensor based approaches and data driven approaches. Each of these approaches have produced powerful techniques for anomaly detection in CNC machines, but they have yet to gain mainstream adoption in production settings. Each of the key aspects of the various classes will be summarized in the following sections, along with a brief discussion on some of the limitations of each approach.

2.2 Literature review

To provide a reader with an understanding of the different approaches that have been explored to solve this problem, the following literature review summarizes the contributions in three main categories: mechanistic modeling based approached, sensor and statistical based approaches and other data driven approaches.

2.2.1 Mechanistic modeling based anomaly detection

In traditional machining approaches, cutting conditions are often selected in a conservative manner based on operator experience and rules of thumb passed on from master to apprentice during the training of a tradesmen. This often results in somewhat arbitrary settings being chosen, which may not make optimal use of a milling machine's capabilities. One early approach to address this was to develop a mechanistic model of the CNC machine. Mechanistic models aim to model the cutting forces of the machine end mill to calculate the in-process chip geometry. These models can then be used to calculate tool vibration, tool deflection and cutting forces using the model seen in Figure 2.1[153].

In mechanistic modeling, the cutting forces are related to the uncut chip thickness bh through empirical cutting force coefficients K_t and K_r [5]. These coefficients are related to the normal cutting force F_r and tangential cutting force F_t by

$$F_t = K_t bh \tag{2.1}$$

$$F_r = K_r bh \tag{2.2}$$

where the coefficients are measured during cutting experiments and extracted by linear curve fitting with varying chip thickness h [1].

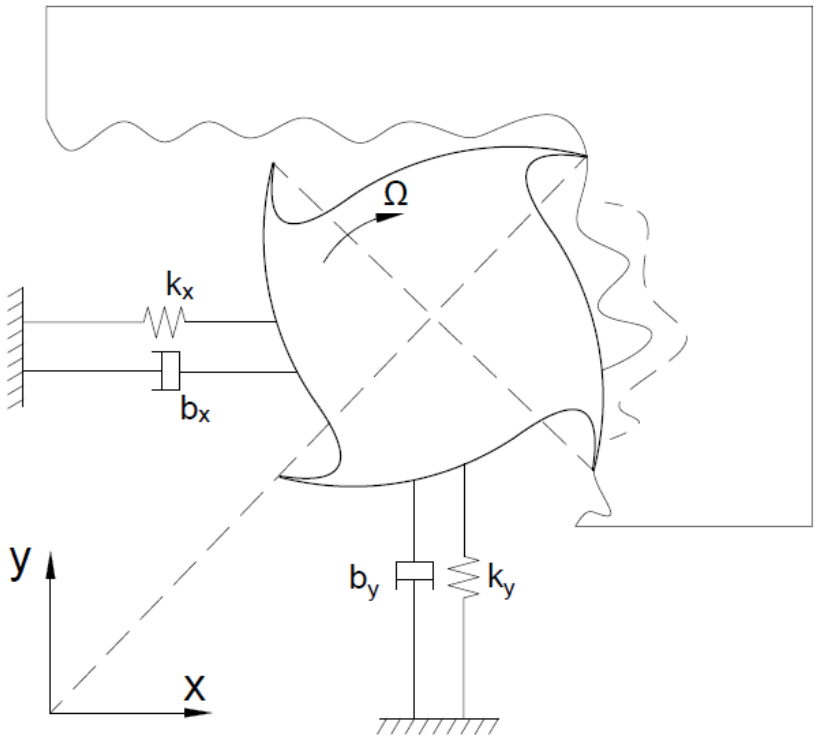


Figure 2.1: Resistor model of cutting tool used in CNC machining

Equation (2.1) and Equation (2.2) can also be modified to represent basic non-linearities in the cutting force expressions when the edge forces are neglected in the mechanistic models [4]. This is achieved by modifying the cutting constants as follows

$$K_t = K_T h^{-p} \tag{2.3}$$

$$K_r = K_R h^{-q} \tag{2.4}$$

where p and q and K_T and K_R are 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

$$h(t) = h_0 - [y(t) - y(t - T)] \tag{2.5}$$

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 [5]. The equations of motion for the machine can be described as

$$m_y\ddot{y}(t) + c_y\dot{y}(t) + k_y y(t) = K_f\alpha[h_0 + y(t-T) - y(t)] \tag{2.6}$$

By using these models, we can calculate the intended chip thickness and then monitor for any differences. Although these models have been employed successful in the analysis of machine tool dynamics, they are only applicable for specific conditions, thus needing to be re-obtained for each milling operation with different cutter geometry and workplace material [1].

One study identified that the main source of vibrational instability in machining was due to regeneration in subsequent cuts [189]. 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, leaving chatter marks on the machined part, and undesirable parts with poor finish.

To address these issues, researchers used the dynamics described in Equation 2.6 to attempt to predict the conditions when chatter would occur. They 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 [1]. Solving this equation produced what is known as the *stability lobe*. An example of a stability lobe diagram is shown in Figure 2.2.

Stability lobes provide a relationship between the cutting depth and the spindle speed for which the cuts will result in either stable or unstable conditions. Knowing this information allows an operator to push their machines to the limits without unstable cutting conditions.

These methods have proven effective and have been used within the literature and in production for decades. However, as the machine conditions change, either due to age or changes in material, the cutting parameters also change [188]. These changes can cause cutting forces to change by as much as 30% [229]. With such significant potential variance in cutting force, the machines cannot safely be run at the limit of the stability lobes, causing operators to run them in overly conservative modes. Furthermore, to obtain the data for stability lobes, an operator needs to have a high speed data acquisition card, impact hammers, accelerometers, microphones and sophisticated software, often at a cost in the tens of thousands of dollars. These limitations, along with the high cost to obtain the

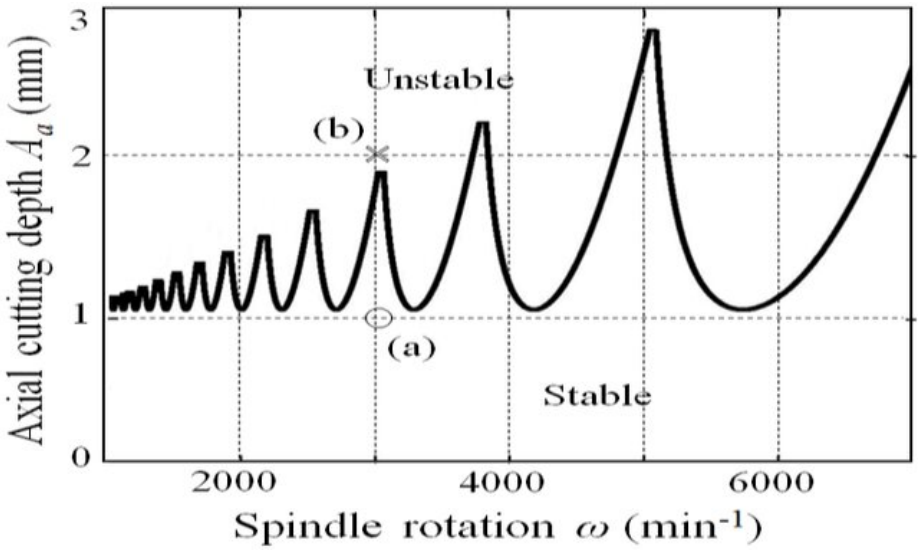


Figure 2.2: Example of traditional stability lobe [184]

stability lobes, restricts the use of many of these techniques to only expensive materials, such as titanium, where the investment in this investigation may be worthwhile.

2.2.2 Sensor based methods of anomaly detection

When mechanistic models and stability lobes were first being developed, the primary sensor used to develop these models was the dynamometer. Dynamometers are high precision force sensing devices, that require careful calibration to use. Since that time, there have been many different sensors developed, tested and integrated for anomaly detection in CNC machines. These sensors present the possibility of changing the paradigm for data collection in CNC machines.

Given the many types of sensors currently available, researchers have postulated it was possible to obtain more information about the machining operations by looking at the intersection of the data produced by these sensors. Sensor fusion techniques allow us to close the loop between the mechanistic models and the state of the machine by leveraging this intersection of information. By integrating these methods, we can achieve a clearer representation of the physical state of the system, rather than simply relying on modeling alone [223], [84]. This allows for a more complete picture of both the machining processes and the machine state [42]

Multi-sensor approaches for anomaly detection in CNC machining rely on taking multiple measurements from varying types of sensors. As shown in [87], systems made of three sensors provide an optimal picture of the machining processes and allow for the most information to be gained. Using these sensors, we aim to capture the dynamics of stable cutting processes, and then observe changes from the sensors to identify when anomaly conditions have occurred. For example, if an accelerometer is mounted onto the spindle of a milling machine, there will be certain accelerations expected caused by the vibrations from cutting. However, when an anomaly condition such as chatter occurs, we expect the vibrations to be greater resulting in larger magnitudes of acceleration. An example comparing accelerometer data from stable and unstable cutting conditions can be seen in Figure 2.3 and Figure 2.4 respectively.

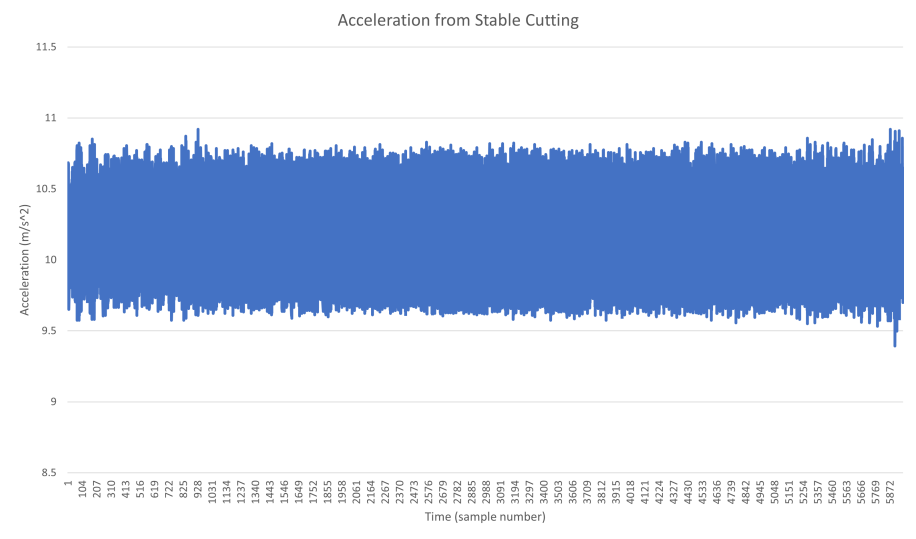


Figure 2.3: Example of accelerometer signal from stable CNC cutting conditions

By using these sensors and monitoring the outputs, we can detect when anomalies occur; but monitoring time varying signals at various frequencies and different thresholds is a time consuming and challenging task. To address this issue, researchers have implemented signal feature extraction methods. These approaches allow for the key features of the signals to be identified, thereby reducing the size of the original processed signal, but maintain the relevant information in the extracted features [89], [42]. To achieve signal feature extraction, researchers use signal processing techniques to decompose the sensor signals, and then, depending on the signal processing technique, attempt to identify key features relating to the desired anomalies.

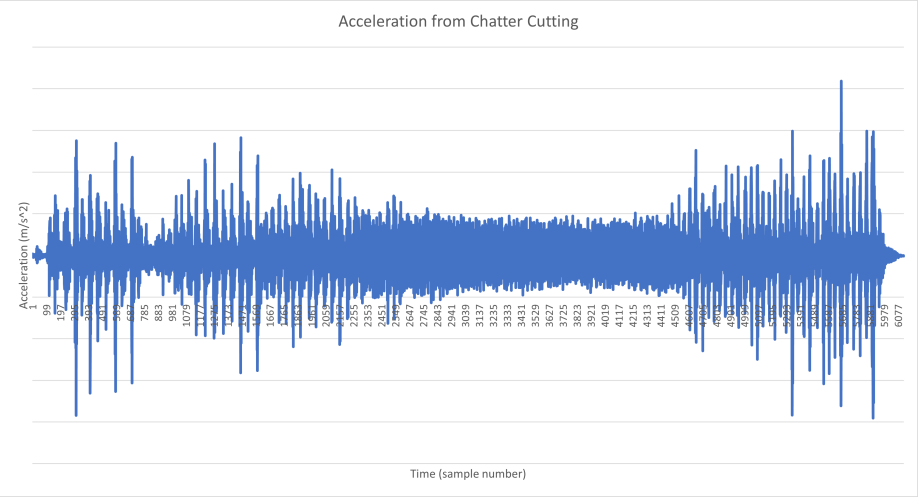


Figure 2.4: Example of accelerometer signal from chattering CNC cutting conditions

The most common signal processing techniques can be broken down into three major categories: statistical analysis, frequency analysis, and wavelet analysis [90], [150]. These three categories are broadly applied and represent many different approaches but will be summarized here at a high level.

Statistical analysis

Statistical analysis techniques aim to represent the different sensor signals with a statistical representation, an example of this is the Mahalanobis distance measure. This distance can be calculated between two points with the following

$$d_M(x, y) = \sqrt{(x - y)^T S^{-1}(x - y))} \tag{2.7}$$

where S is a positive-definite covariance matrix representing the distribution of interest.

In this approach, we are interested in measuring the magnitude of an expected signal and then attempting to see if that magnitude lies within an expected threshold. This threshold is determined by calculating the variance of a stable signal and then ensuring that the signal does not deviate greatly from an expected mean [28]. If the deviation is too large then it will be classified as an outlier caused by an anomaly.

Frequency analysis

In frequency analysis, we analyze the sensor signals in the frequency domain. This frequency domain analysis is accomplished with traditional algorithms such as Fourier transformations and Power Spectral Density transformations, where the sensor signals are decomposed into their respective frequencies. Once the sensor signal has been transformed to the frequency domain, the user can search for representative peaks at specific frequencies. Stable cutting conditions for a specific material will produce repeatable signals at specific frequencies, whereas anomalies will produce frequency peaks outside of this range [120]. An example of frequencies from stable and unstable cutting conditions can be seen in Figure 2.5.

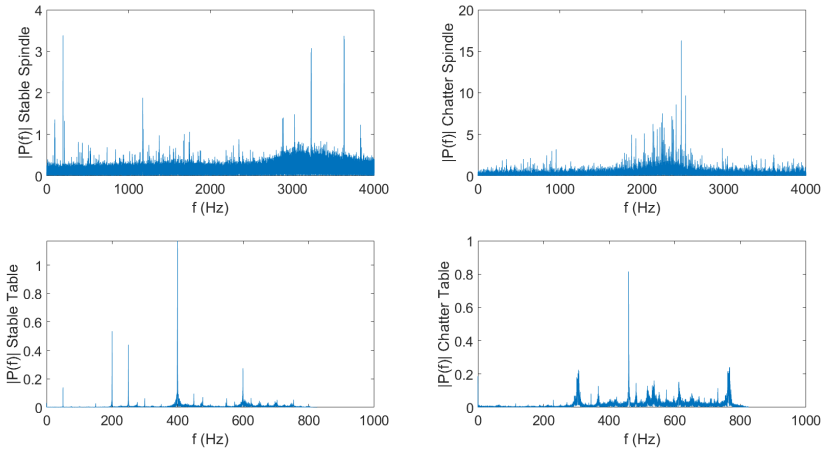


Figure 2.5: Example Frequency Analysis from stable and unstable CNC cutting conditions

Wavelet analysis

In wavelet analysis, a sensor signal is decomposed using wavelet package decomposition (WPD). In WPD, the signal is broken down into both frequency and time domain representations. This allows for both frequency and time series information to be captured [231]. With this information, the users can look for specific shifts in sensor data in either the frequency domain or the time domain to indicate the presence of anomalies. An example of this can be seen in Figure 2.6 for a stable wavelet and Figure 2.7 for a chattering wavelet. These wavelets are generated using the Daubechies wavelet that applies an orthonormal filter to the system. The figures for the stable and unstable wavelets indicate

the decomposition of the signal into its subsequent components over three levels on the left, with the signal output seen in the right. In these figures, we see the decomposition of the acceleration signal for both the stable and unstable cutting conditions, with significantly larger amplitudes noted for the chatter cutting condition.

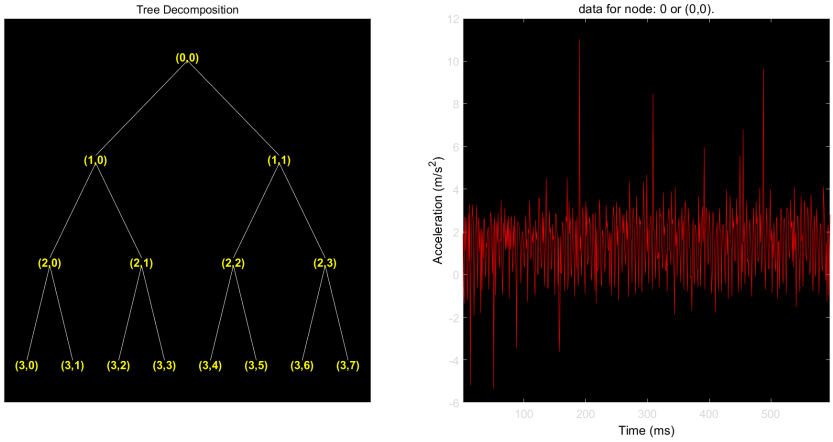


Figure 2.6: Example of WPD signal from stable CNC cutting conditions

The feature extraction methods described have many different parameters, thresholds and options that must be selected and tuned to properly detect anomalies. With such a large number of variables, the accuracy of the anomaly detection approach can vary greatly depending on the methods chosen. To complicate matters further, the thresholds and values used in these algorithms are obtained from experimental data. This makes the results sensor, material, and machine specific, making it challenging to apply to different systems [134] [193]. These restrictions limit the application beyond a single machine and requires a significant amount of time and investment to be made to obtain comparable results.

2.2.3 Data driven methods for anomaly detection

Modelling the machine dynamics of a CNC machine is a challenging task requiring extensive analysis and time to accurately develop a model of the system. These mechanistic and sensor models, as described before, have been used for many years for anomaly detection in CNC machines.

However, in recent years as machine learning algorithms have advanced and found their way into application in many CNC machines. Machine learning and data driven

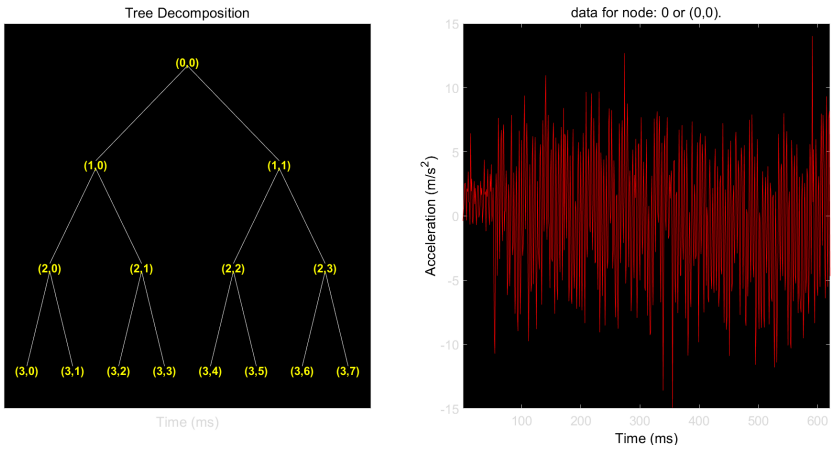


Figure 2.7: Example of WPD signal from chattering CNC cutting conditions

algorithms allow for the complex dynamics of CNC machines to be modeled using experimental data, rather than deriving mathematical models analytically. This allows for many of the difficult or unmodeled dynamics to be captured in an effective repeatable method [100] [61][180][157][141]. Of these approaches, the most common algorithms implemented are the Support Vector Machine (SVM), Convolutional Neural Network (CNN), and the Long Short-Term Memory (LSTM) variant of the Recurrent Neural Network (RNN).

Many machine learning algorithms implement the SVM approach as it is a simple and lightweight machine learning algorithm. In an SVM, the system is attempting to find a hyperplane to separate the data into different categories [15]. This is achieved by attempting to optimize the following

$$\min \frac{1}{2}|w|^2 \tag{2.8}$$

$$\text{such that } y_i(wx_i - b) \geq 0 \tag{2.9}$$

where

$$wx_i - b \geq 1 \text{ if } y_i = 1 \tag{2.10}$$

$$wx_i - b \leq 1 \text{ if } y_i = -1 \tag{2.11}$$

where w and b are network weights that are achieved by training data x_i and y_i.

This approach is simple to implement, as it forms a linear regression and allows authors to establish a boundary between stable information and anomalies. This has been

implemented in works such as [7] and [94]. Although the simplicity of this approach allows it to be applied in real time, it also limits the potential to handle more complex cases, since binary classifiers do not easily extend to more than one case at a time. This creates significant issues when more than one anomaly needs to be detected, or if the anomaly may result from several faults. Unless there is a clear separation in the sensor data, the system may make errors and be unable to detect the changes due to an anomaly.

To address the limitations of the SVM algorithm, many authors have implemented CNNs for anomaly detection. CNNs are a form of deep neural network based with special convolutional networks built in. For example, if a neural network is represented by

$$y = f_{nn}(x) \tag{2.12}$$

then the various layers of the network can be represented as

$$\begin{aligned} y &= f_{nn}(x) && (2.13)\\ &= f_n(...f_l(f_2(f_1(x)))) && (2.14) \end{aligned}$$

where each layer is represented by

$$f_l = g_l(W_l z + b_l) \tag{2.15}$$

where l represents the layer and g_l is the activation function parameters, W_l the layer weights and b_l the bias term. In this formulation, the individual inputs are selected based on the weighting matrix W_l, with some bias term b_l, feeding into an activation function g_l. In a CNN, the layers are made up of convolutional filters with trainable filter and bias parameters, which allow the CNN to pass along key feature information from one layer to the next, allowing for more complex information to be captured. This approach has found significant success in the realm of computer vision and has spawned a plethora of research.

As a result of these successes, many of the techniques developed in computer vision have been transferred to machining applications [221], [7],[147][71][117][116][224]. Although CNNs allow for greater complexity to be captured, they are only able to evaluate information as snapshots in time, and do not preserve the connection from one instance to the next. This causes the network to lose the connections through time, causing patterns to be missed.

Recurrent Neural Networks establish a connection between the different sequences in the data. This is accomplished by modifying the traditional neural networks by adding feedback loops between each layer. This is accomplished by establishing units u, for each layer l and then capturing the state in a memory unit $h(u, l)$.

The LSTM is an extension of the RNN where, as the name suggests, the network is capable of capturing both long term and short-term memory of the system of interest. This is accomplished by establishing gates that control the flow of information in and out of the cell. These gates, control the input and output of information, as well as how much information is forgotten through each sequence. LSTM networks have shown incredible promise and have been implemented in a multitude of works including [52][62][6][21][77][126][116][224][104][139][200][165][227]. The details of these networks are described in detail in Chapter 3.

Although these approaches have shown considerable promise and have addressed many of the deficiencies of their predecessors, they require a significant amount of training data to properly model the system. This training data can only be obtained by performing cutting operations on the machine of interest. In practice, this either results in a long delay before an algorithm can be used, or many unnecessary parts being produced for the algorithm to function. This restriction limits the ability of the algorithms developed in the lab to be implemented in production. To address this limitation, this thesis proposes several methods that reduce the burden of data collection from the operator to increase acceptance of the anomaly detection method. These methods are based on implementing transfer learning and incremental learning techniques to reduce the upfront data collection requirements.

Transfer learning is a technique that uses data from another data set, known as the source, to train an initial network. The knowledge from this network is then "transferred" to the network of interest, known as the target, and re-trained using data from that machine. This approach provides opportunities to significantly reduce the required amount of training data to be collected, and reduce the training time on the target machine. This proposed method is explored in detail in Chapter 4.

The second method that this thesis focuses on is *incremental learning*. As the name suggests, incremental learning has the network learn in small batches, or increments. Although there are many different types of incremental learning algorithms, a major tenet of all of them is the focus on developing a model capable of learning and enhancing its knowledge progressively, without forgetting previously acquired information. This could potentially allow users to deploy a rudimentary system that incrementally learns and improves its performance as more data is naturally made available. This method is described in greater detail in Chapter 5.

2.3 Preliminaries

This section provides the reader with some background information on popular machine learning aspects that are used heavily in this thesis. The subsections are written to provide a condensed background to easily understand the contributions in the thesis.

2.3.1 Recurrent Neural Networks

Machine learning is made up of many different types of algorithms, each with their own advantages and disadvantages. One of the most powerful types of machine learning algorithms is the artificial neural network (ANN). These networks aim to mimic the connections of the human brain for a machine. Early machine learning algorithms used shallow networks, with minimal connections, and only a feed-forward flow of information. Although this approach is lightweight and effective, it is also limited, as networks tend to only capture instances in time, and do not fully capture time dependent trends.

Recurrent neural networks (RNN) are another type of ANN that introduce a bi-directional flow of information. This means that the output from some nodes can act as subsequent input to the same node. This effectively provides a closed loop response, and creates an internal state, or memory, for each node. The RNN configuration can be seen in Figure 2.8, with the mathematical definition presented later in this section.

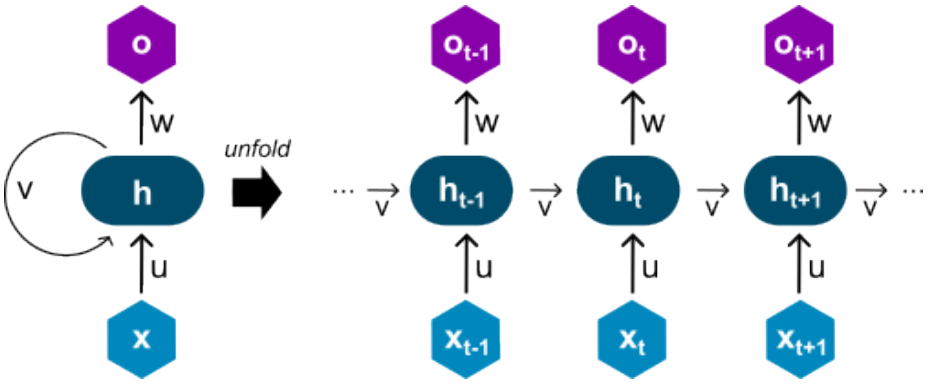


Figure 2.8: Recurrent Neural Network architecture

In general, a RNN is a special type of neural network that focuses on sequential or time series data. These types of networks are often used in temporal problems, such as language translation, speech recognition, or natural language processing [66]. These sequences are

learned by adding connections between the nodes of the network, allowing the network to learn the relationship between each sequence of data, as well as the inter-relationship between the data during the current time step. This approach allows the network to develop a "memory" of the state of the system over time, where the state describes the system or process dynamics in a given instance of time. This is particularly useful in engineering applications, where system dynamics are heavily influenced by time dependent physical phenomena.

Although RNNs are able to create a memory state of the system, they suffer from only having a short-term memory. This means that if a time series is long enough, the network will have problems carrying the impacts from earlier time steps forwards. To solve this problem, two types of networks have been developed, namely the Gated recurrent units (GRU) and the Long-short-term-memory networks (LSTM). Both types of networks introduce memory gates that control the flow of information. These gates allow the network to learn how to remember or forget certain features. Both GRUs and LSTM function in a similar manner, but the LSTM has an additional output gate that allows it to control the information output, as well as the input and feedback. For this reason, we believe it to be a preferable choice for the types of problems of interest.

Long-Short-Term-Memory Artificial Neural Network

The Long-Short-Term-Memory (LSTM) algorithm is a type of artificial neural network that is popular in the fields of artificial intelligence and deep learning. The LSTM network is a form of RNN, which, as its name suggests, is capable of storing both long term and short term memory about the system being modeled.

Although RNNs are effective, the networks are prone to suffering from exploding and vanishing gradient problems. These problem occur when the networks grow too large due to many sequences being added. As a result of the size of the network, the algorithm is unable to properly update the network weights, preventing it from converging to the correct solution. This effectively causes the system to lose the ability to make connections between newer and older sequences, causing the system to lose its longer term memory.

To address the vanishing gradient limitation, Hochreiter and Schmidhuber developed the LSTM network [58], which was then further refined by Gers in [44]. The LSTM network allows the system to retain short term memory in excess of 1000 time steps, while maintaining long term memory of past events. This is of particular interest, since key events may have occurred with large gaps in-between one another [58].

In a LSTM, each unit is composed of a cell, an input gate, an output gate, and a forget gate. Each cell retains the memory for a time interval, with the gates controlling the flow of information in and out of the cell. The input and output gates, as their names imply, control the flow of information in and out of the cell. The forget gate calculates the proportion of information to be kept, thereby avoiding the state from growing indefinitely and breaking down [44]. An example of this could be a state representing a process variable in a system such as temperature. If the network does not have the ability to "forget" the impact of previous sequences, it may inadvertently increase the temperature beyond reasonable or accurate limits, causing the network to no longer effectively be able to model the dynamics, thus breaking down. This cell representation can be seen in Figure 2.9 that shows the flow of information from the past cell c_{t-1}, to the current cell and to the next cell c_t. The figure shows that various gates, represented by σ have been inserted to control the amount of information that is used in each calculation.

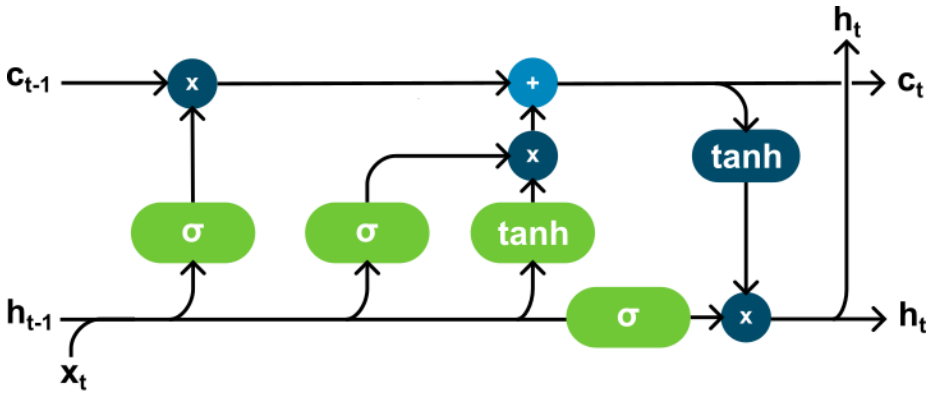


Figure 2.9: LSTM Cell layout

Following the description of the network outlined in [170], and developed by [58] and [44], the key equations of the LSTM network with forget gate are summarized in Equations (2.16) to (2.19). In these equations, the lowercase variables represent vectors, while uppercase variables represents matrices.

As outlined in [58], the LSTM architecture can be defined with matrices $W_q \in \mathbb{R}^{h \times d}$ and $U_q \in \mathbb{R}^{h \times h}$ 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.

$$g_t = \tilde{C}_t = tanh(W^g x_t + U^g h_{t-1} + b^g) \tag{2.16}$$
$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f) \tag{2.17}$$
$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b^i) \tag{2.18}$$
$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b^o) \tag{2.19}$$

where the operator $\odot$ denotes the Hadamard product, otherwise known as element-wise multiplication, σ represents a sigmoid function and the subscript t indexes the time step. The variables used are summarized in Table 2.1

Table 2.1: 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

2.3.2 LSTM Autoencoder

A key structure used in this thesis is the Autoencoder. An Autoencoder is a feed-forward neural network with an encoder-decoder architecture [15]. This type of network is made up of two parts, an encoder and a decoder, with bottleneck layer in between, and is trained to reconstruct a given input. This architecture can be seen in Figure 2.10. In the figure we see that information is flows through the network through layers of decreasing size towards an embedding bottleneck. Afterwards, the data flows out of the network through layers of increasing size.

The encoder is a series of layers that aims to capture a numerical representation of a given input state. This is typically captured in a vector or matrix known as the embedding. A decoder is another series of layers designed to perform the opposite operations from the

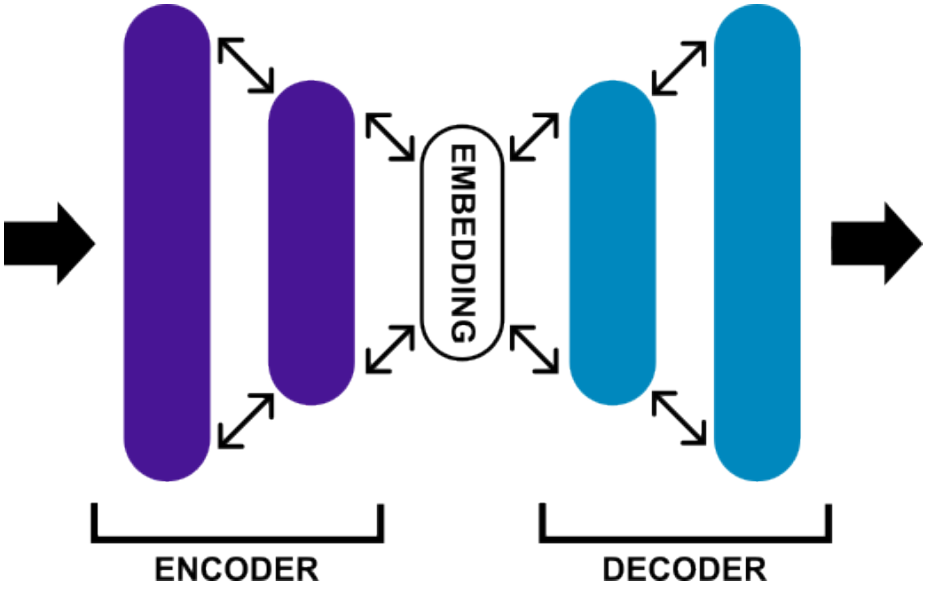


Figure 2.10: LSTM Autoencoder Architecture

encoder, namely to take the embedding and reproduce the original input. This architecture allows for considerable benefits as it is flexible and captures a large depth of information.

We leverage this architecture for anomaly detection by training the Autoencoder using only information from stable cutting conditions. We can then evaluate the difference between the original input and the reconstructed output from the network, this difference is known as the reconstruction error. This reconstruction error is typically defined by evaluating the Mean Absolute Error (MAE), defined as

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}. \tag{2.20}$$

Given that the Autoencoder is trained only using stable cutting conditions, a properly trained network will thus reconstruct a stable cutting signal with little error. However, a signal from an anomaly condition, such as chatter or a broken tool, will reconstruct poorly with a large reconstruction error. The reconstruction error can therefore be used as an indicator of anomaly conditions, if its value crosses a pre-defined threshold.

The threshold for anomaly conditions can be determined by deriving a histogram from the reconstruction error of stable cutting conditions. From this histogram we can determine the mean and variance of the stable cutting data and then set a threshold based on the

three-sigma distance from the mean, which is commonly used to determine a statistical anomaly. An example of this histogram can be seen in Figure 2.11.

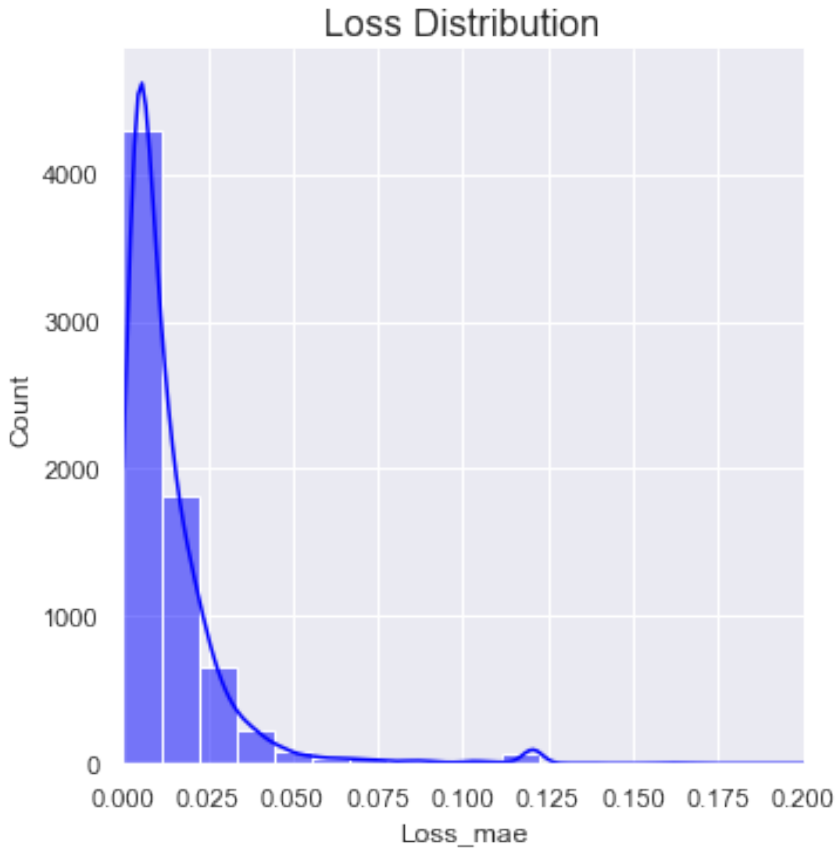


Figure 2.11: Histogram of reconstruction loss from LSTM Autoencoder

2.3.3 Data augmentation of training data for machine learning

A popular technique within the machine learning literature for providing more training data and to reduce overfitting is called data augmentation [15]. In data augmentation, synthetic data is created from original data by applying different transformations to the original data. For an image, this could be accomplished by flipping or rotating the original images to produce more training data.

As first outlined in [32], data augmentation is a statistical technique that generates

maximum likelihood estimates from incomplete data. In the previous example with images, the incomplete data may be a scenario where the image is askew. Data augmentation fills in this gap by applying transformations to mimic these scenarios.

This data augmentation approach has also been extended to time series data to provide similar results. Examples of data augmentation for time series data include adding random noise, time warping, down sampling and flipping the data [207]. These data augmentation techniques provide many powerful tools for improving the fitting of the data, but must be explored carefully to select the optimal technique.

2.3.4 Ensemble learning

Ensemble learning is a technique that can be used to train deep neural networks and potentially improve their performance. Rather than relying on a single system, an ensemble learning approach trains many low accuracy models, known as weak learners, and then combines their outputs [15]. The majority of these ensemble learning techniques can be broken into two methods, boosting and bagging.

In boosting, multiple models are iteratively trained, with each model being different. The aim of boosting is for the models to learn the differences that previous models could not. This contrasts bagging, where copies of the models are made, each with minute differences, that are then combined.

One of the most popular boosting algorithms is an adaptive boosting method known as "AdaBoost" [41]. This algorithm creates an initial strong learner by training a model on a large number of epochs, then trains many weak learners, that are created and trained quickly, and combines the output with a weighted average. This weighted average allows the model to adaptively change the output to favour more accurate weak learners and reduce the impact of any misclassifications. This approach has proven effective and popular, and has spawned many variants and extensions.

In AdaBoost, a boosted classifier is of the form

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \tag{2.21}$$

where f_t represents a weak learner.

Each weak learner contributes a prediction $h(x_i)$ for a given input condition x_i. The weak learners are then combined with a weighting coefficient α_t that is updated at each

time step t. These coefficients are chosen such that the total error E_t is minimized, where E_t is defined as

$$E_t = \sum_t E(F_{t-1}(x_i) + \alpha h(x_i)). \tag{2.22}$$

Chapter 3

Anomaly Detection in CNC Milling Machines using Sensors and Machine Learning

3.1 Motivation for anomaly detection algorithm in CNC milling machines

In the world of CNC machining, an anomaly is anything that causes the milling machine to deviate from its expected cutting operations. This broad definition includes a multitude of situations including tool breakage, machine chatter, spindle failure or severe tool wear to name a few [108]. Given that there are so many potential anomalies to consider, it is prudent to define which anomalies are of focus in this thesis. As such, the main two anomalies considered in this thesis are chatter and tool breakage. These two anomalies were considered because they are well studied and could be induced safely. By choosing these two well studied anomalies, we can build a baseline with which to compare the accuracy and effectiveness of the algorithms studied.

As outlined in [5], vibration due to self excited chatter is one of the most common deformations seen within CNC machining. During this phenomena, the CNC end mill vibrates and eventually jumps out of the cutting zone or breaks due to growing dynamic displacement between the tool and the work piece. The phenomena, if not properly detected and controlled, can lead to unacceptable surface finish, leading to the part being scrapped and the effort wasted. Due to the frequency of this phenomena, and the financial

losses associated with not addressing it, there has been considerable effort invested in both industry and within the research literature.

The second anomaly of interest in this thesis is tool breakage. Extreme cases of tool breakage where the entire end mill is broken are rare in production environments, where operator experience and expertise prevent the end mill from being pushed to these extremes. However, the wear on the tool through extended usage and the damage of cutting flutes is more common and often undetected, as visible marks are not always left. Nevertheless, cutting with damaged tools comes at a cost of increased energy consumption and potential surface defects. To address this, there is a growing body of literature on detecting tool wear and tool breakage.

3.2 Sensors used for anomaly detection in CNC machines

To detect the existence of anomalies and provide feedback to the system, some method of sensing the workspace or machine is required. This is most commonly accomplished by integrating sensors and signal processing systems into the CNC milling machine. From the excerpt from [90], seen in Table 3.1, there are a variety of potential sensors that can be chosen. This multi-sensor approach has produced a wide body of research where different sensors have been implemented for monitoring machining processes. Although these approaches are effective, the level of investment in different sensing hardware and software solutions often discourages their implementation [89]. As such, we wish to choose a limited number of sensors that can provide the capability to detect a wide variety of anomaly conditions. An accelerometer is an ideal choice for this implementation, as it can be used to detect many conditions, including chatter, tool breakage and more[90]. An example of the difference between a stable and unstable signal can be seen in Subsection 2.2.2. In Figure 2.3, we see an accelerometer signal from a stable condition and in Figure 2.4, we see an acceleration signal from an unstable condition. The stable condition has little variance, whereas the unstable condition has many peaks and valleys.

To confirm the feasibility of the various sensors for anomaly detection, and to identify the optimal choice, cutting experiments were completed using microphones, accelerometers, encoders and cameras. These tests were completed on a simple test bench, described in Section 3.4.

During these tests, stable and unstable cutting conditions were encountered and the data was then post-processed using techniques described in the literature review. These

Table 3.1: Typical sensors used in CNC milling operations

Utilized sensors	Main objective
Accelerometer	Flank wear monitoring
Dynamometer	Tool wear prediction
Dynamometer	Tool wear monitoring
Current sensor, vibration sensors	Tool wear monitoring
Power sensors	Abnormal condition detection
Accelerometer, dynamometer, acoustic emission sensor	Tool wear prediction
Acoustic emission sensor, accelerometer	Tool wear monitoring
Acoustic emission sensor, current sensor, accelerometer	Flank wear prediction
Acoustic emission sensor	Sensor location determination
Dynamometer, accelerometer	Observing frequency bandwidths
Dynamometer, accelerometer	Recognition of machining
Accelerometer, dynamometer, acoustic emission sensor	Tool wear prediction
Acoustic emission sensor, current sensor, accelerometer	Tool wear prediction

manual techniques established a baseline and ground truth labels that were used for training machine learning algorithms in the future. This work also confirmed that using accelerometers only, provided sufficient richness of information for the number of anomalies of interest, but additional sensors could be used to detect more information in the future. Nevertheless, accelerometers were chosen as the primary sensor used for this thesis, as they are relatively inexpensive and could theoretically be deployed easily to every CNC machine.

The greatest impact on cutting stability during a CNC milling operation will occur at the most flexible point of the machine [5]. For CNC milling machines this occurs near or at the spindle, thus to obtain the most accurate measurement of the machine's state, the accelerometers are mounted on the spindle housing, as seen in Figure 3.1.

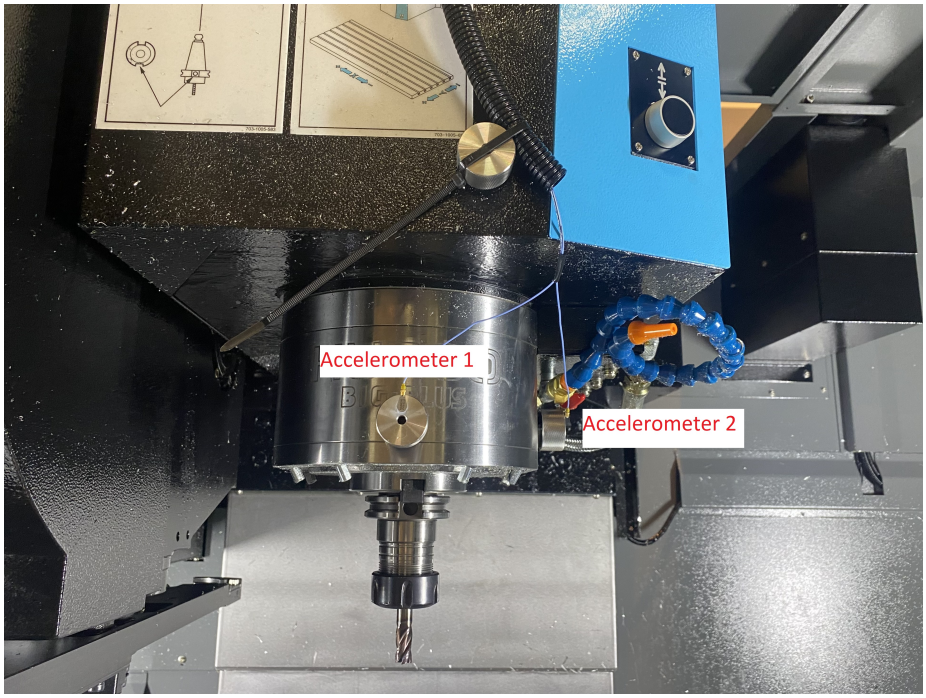


Figure 3.1: Piezoelectric accelerometers attached to a Hurco VMX42Di CNC machine.

By completing these cutting experiments on the simple test bench, we were able to identify the optimal sensors and locations for anomaly detection in CNC machining.

3.3 Machine learning algorithm for anomaly detection in CNC Machines

Many typical CNC milling jobs take place over an extended period of time, with jobs taking several hours to complete. However, anomalies such as chatter can occur in as little as half a second, potentially destroying a part and ruining hours worth of work. In addition, the state of a machine can change from years of prolonged use, with the machine's tool cutting parameters changing due to tool wear.

To capture both these long term and short term changes, we turn our attention to a machine learning algorithm that is inherently designed to capture such dynamics, the Long-Short-Term-Memory (LSTM), described in the preliminaries in Section 2.3.1. One of

the advantages of using a deep network like an LSTM, is that the network can be arranged in such a way to capture the dynamics of a system and attempt to reconstruct the system input using an architecture known as an Autoencoder. The Autoencoder works by taking an input signal and then attempting to reconstruct an estimate of the input signal. If the system reconstructs the input signal properly, then the signal is likely from a stable condition. However, if the signal reconstruction is poor, then the signal is likely from an anomaly. The Autoencoder has proven to be a powerful tool for detecting a variety of anomalies, but also opens up many possibilities that will be explored in this thesis. The LSTM and the Autoencoder configuration are both explained in greater detail in the following sections.

The Autoencoder is made up of two main components, an encoder and a decoder. The encoder is made up of several LSTM layers leading to a bottle neck layer. The decoder then takes the output from the bottle neck layer and reverses the work from the encoder. This general design is used in many different types of Autoencoders, and is not specific to the LSTM-Autoencoder. We define this mathematically as follows:

Given an input x, the encoder of a system is defined as

$$g(x) = f_{LSTMn}(...f_{LSTMl}(f_{LSTM2}(f_{LSTM1}(x)))) = h \tag{3.1}$$

where h is a latent embedding of the input vector, and f_{LSTMl} is a LSTM layer. We define the decoder as

$$f(\cdot) = f_{LSTMn}(...f_{LSTMl}(f_{LSTM2}(f_{LSTM1}(\cdot)))) \tag{3.2}$$

and the Autoencoder is defined as

$$f(g(x)). \tag{3.3}$$

This Autoencoder is trained with a loss function to minimize the reconstruction error $\hat{x} - x$, where $\hat{x}$ is a reconstructed estimate of the input vector.

Given the general nature of an Autoencoder, there are many different design parameters to consider that impact the networks' ability to converge to a viable solution, as well as the network performance. The size and number of layers, as well as the methods for training the system must be considered.

To determine the network structure for the Autoencoder, some simple cutting tests were completed using an experimental platform. These tests were conducted with a limited number of cutting tests, with data collected using an off the shelf, wireless three-axis accelerometer (MPU6050). Fifteen test cuts at stable conditions were completed to obtain

training data. Eight tests were then conducted, with six stable cuts and two from chatter conditions, for testing conditions. Using this data, the size and number of layers were iterated to ensure that the system could properly converge.

This network was trained using data augmentation and mini-batch gradient descent [79]. In mini-batch gradient descent, we pick a small batch of data and feed it to the neural network. We then calculate the mean gradient error between the output and data, and then use this gradient error to update the weights, repeating this process for each batch. This is accomplished effectively by using vectored implementations.

Several tests were conducted to compare various time series data augmentation techniques to select the optimal choice. From the experiments, it was found that dynamic time warping provided the most significant improvement for LSTM-Autoencoders for CNC milling machines because the changes in cutting dynamics over time can still be captured by this augmentation technique, whereas other techniques introduced knowledge that was not pertinent to the situation. In addition, a moving average window was applied to the sensor inputs to provide some filtering of high frequency noise from the accelerometer. Through this design iteration, we were able to determine that the structure seen in Figure 3.2 would provide feasible results. Each rectangle seen in Figure 3.2 indicates a layer of the network, with the size of the layer indicated in the upper corner of the rectangle. This structure has approximately 45 000 training parameters, using fully connected layers with ReLU activation functions and a MAE loss function. The Autoencoder is made up of three encoding layers, a repeat layer for the embedding, and three decoding layers.

Through this experimentation, one important aspect that was discovered was the impact of the bottleneck layer. The bottleneck layer constraints the amount of information and forces the network to learn a "good but compressed" representation of the input data[50]. If the network is constructed with too many layers and a "gradual" decrease in layer size, then the key representations of the data may not be captured. As such, it is key that there is a relatively rapid decrease in layer size towards the bottleneck layer to ensure that the key aspects of the data are captured.

3.4 Experimental implementation of anomaly detection algorithm

To verify the anomaly detection algorithm's ability to capture the anomalies of interest, a test bench was setup to obtain training and testing data. A desktop X-Carve CNC was instrumented with a variety of sensors to capture different dynamics during cutting

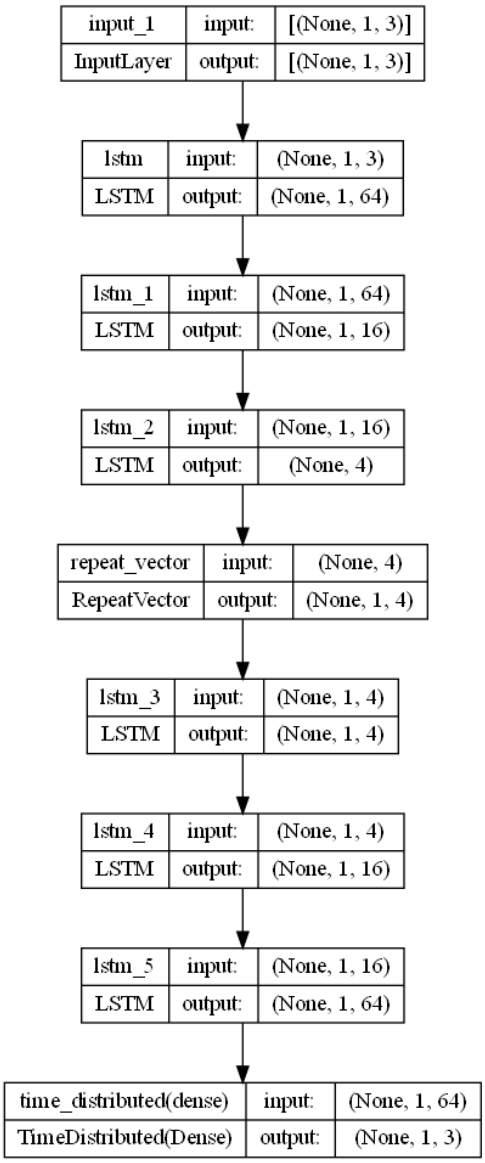


Figure 3.2: Autoencoder structure for LSTM Autoencoder for anomaly detection

operations. These sensors included a three axis accelerometer (MPU 6050), incremental encoders, microphones and a high definition camera. Each of these sensors collected data using an ESP32 microcontroller and then passed their information to a laptop computer. This setup can be seen in Figure 3.3, with the incremental encoders mounted on the machine, the accelerometer mounted on the spindle, and the microphones mounted on the enclosure.

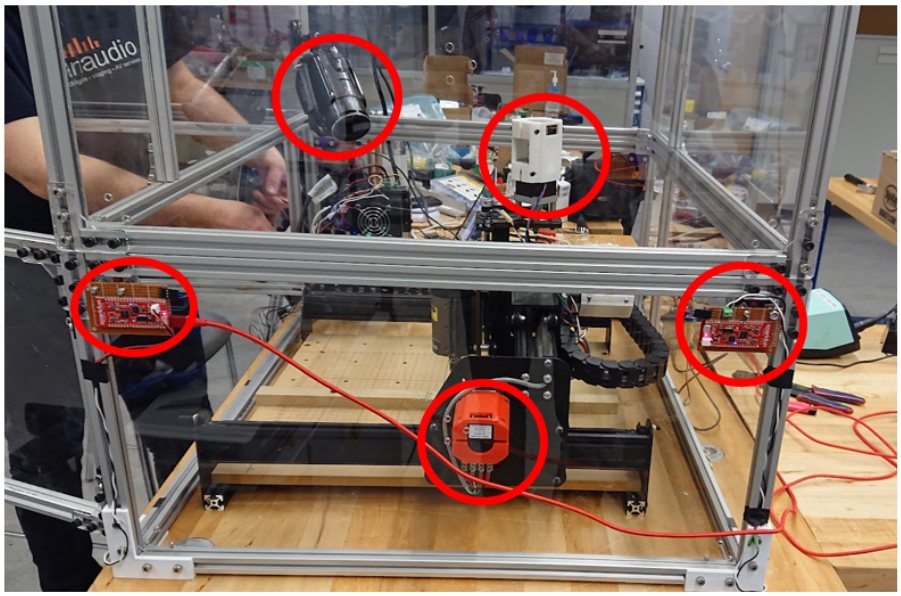


Figure 3.3: Test bench on X-Carve Desktop CNC

This test bench allowed us to compare different combinations of sensors and validate experimental results against those seen in the literature and gain increased understanding of other anomaly detection methods. Furthermore, this test bench allowed us to validate the anomaly detection algorithm proposed in Section 3.3.

Using this test bench, fifteen cutting experiments of stable cutting conditions were conducted. The data from these cutting experiments was then used to train an LSTM-Autoencoder for anomaly detection, using the accelerometer data only. The other sensor data was considered, but did not provide substantially more information that the accelerometer alone could not provide. As such, the data from the other sensors was only used to determine the ground truth stability conditions to compare the anomaly detection system against.

Once the anomaly detection system was trained, eight more cutting experiments were

conducted to test the anomaly detection system. From these eight experiments, six cuts were completed in known stable conditions, and two were completed in known chatter conditions. These eight cuts were used as validation of the system, with the results seen in Figure 3.4. The figure has the reconstruction error on the y-axis and the associated test cut on the x-axis.

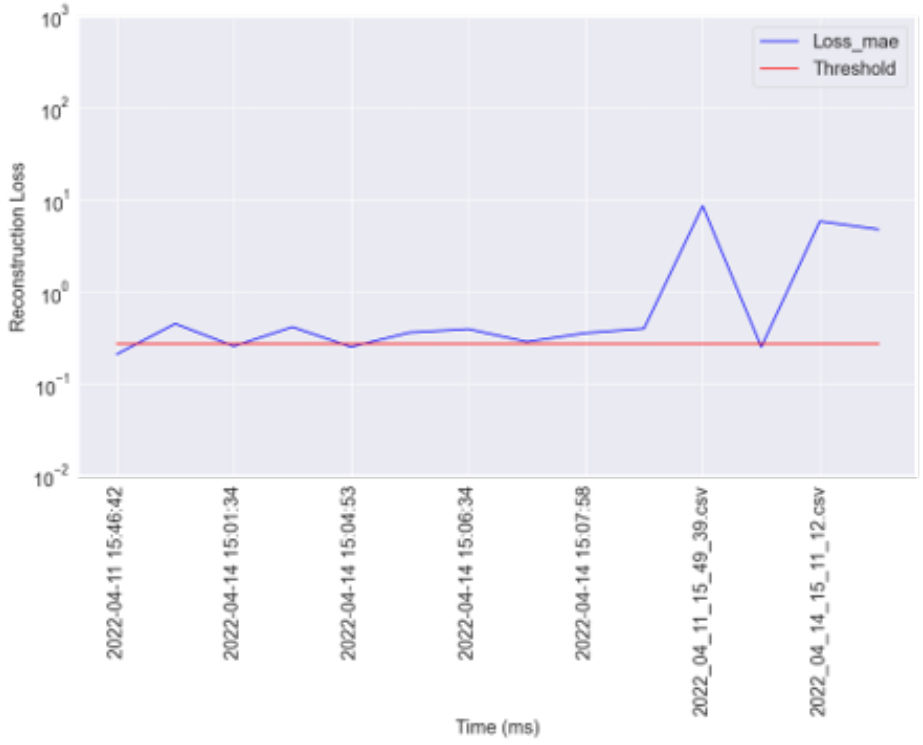


Figure 3.4: Test results for anomaly detection Autoencoder with small sample set

With this limited testing data set of eight data points, the network exhibits some oscillatory behaviour for the stable cuts, but is capable of making a clear distinction for the final two chatter cuts. Although this is not a strong result, it does show that the network is able to converge to a solution that can make a distinction between the data sets with given input data. This indicates that the network structure is stable, and would likely produce acceptable results with a larger data set, providing the motivation to complete future cutting experiments.

With the choice of sensors and locations determined, the test bench was retired. Although the test bench was useful for initial testing, its capability to machine parts was

limited due to its limited power. The remaining experiments in this thesis were conducted on larger more powerful CNC milling machines, capable of machining significantly more complex materials and parts.

3.5 Discussion and summary of contributions

There are numerous anomaly detection methods, each with their own advantages and disadvantages. Based on our experimental results, we believe that many of the dynamics associated with anomalies in a CNC machine can be captured using a series of accelerometers feeding their information into a LSTM-Autoencoder. This structure allows for several different anomalies to be detected, and its deep network structure also provides great flexibility to explore many opportunities. For example, accelerometers can be used to capture several different anomalies, allowing the system to be implemented in many applications. This contrasts with microphones, which would allow for chatter detection systems, but would likely have more trouble detecting broken tools.

This work shows that the proposed structure using accelerometers only can converge to a solution that is capable of separating stable and chatter cuts. In addition, we have been able to identify the impact of bottleneck layers as well as the optimal data augmentation techniques to be used for these problems. Although only a small number of cuts were used to evaluate the system performance, and the network was trained on a small data set, we are able to show that this configuration does converge to a solution. This work provides confidence and a baseline for other work to be developed and compared against.

Chapter 4

Transfer Learning for Anomaly Detection in CNC Machining

In Chapter 3 we showed how an LSTM-Autoencoder could be used with accelerometers to capture the dynamics of a three-axis CNC milling machine for anomaly detection. This method, although successful, requires a large amount of training data to be collected on every machine before it can be generalized and implemented robustly in a production environment. In this chapter, we explore how transfer learning can be used to train neural networks, and potentially reduce the burdens associated with implementing these algorithms in production, by reducing the amount of training data needed to train an anomaly detection system.

4.1 Motivation

First proposed in the 1970s by Bozinovski and Fulgosi [12], transfer learning is a machine learning method where knowledge from a source system is used as a starting point for training a second related system, with similar dynamics and limited new information [185]. 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 [15]. This approach is often used in vision processing tasks where there may be similarity in the images. For example, a network trained to identify dogs can be used as a starting point to train a network to identify cats [51]. Transfer learning provides considerable benefits to users implementing machine learning algorithms, as they

allow them to implement advanced controllers with considerably less training data [96]. This has been applied in other works for chatter detection [195] [217][141] , tool condition monitoring [101] [29], and milling modeling [80][91][204]. These examples demonstrate the feasibility of applying transfer learning in CNC milling, but do so only with vanilla networks, such as SVM, or with CNN or Adaptive Neuro-Fuzzy Inference Systems (ANFIS). These approaches do not explore the full capabilities of transfer learning, and are only used to reduce the training effort, without truly capturing knowledge from one machine and transferring it to another.

Transfer learning provides a promising approach for CNC manufacturers to reduce the amount of effort and cost needed to ship a CNC milling machine with anomaly detection capability. Since CNC manufacturers are continually developing new machines, unless there is no change between a new machine and an old machine, a completely new large set of training data would need to be obtained for every machine that is designed. Given that cutting experiments cost several hundred, to several thousand, dollars to complete for each cut, collecting these data sets costs considerable time and money. Transfer learning allows manufacturers the ability to potentially re-use existing networks and data from old machines as a basis for any new networks as new machines are developed.

In this thesis, we propose transfer learning as a method to train anomaly detection networks in CNC machining using significantly less data than training a system from scratch. The overall approach proposed is summarized in Figure 4.1. In this approach we take a large dataset obtained from stable cutting conditions from one three axis CNC machine, and then transfer its knowledge to a series of other machines.

4.2 Transfer learning algorithm mathematical definition

The mathematical modeling of transfer learning is expressed in detail in [12], and is presented in brief below. Following the conventions defined by Tan[185], we define a domain $\mathcal{D} = \{x, P(X)\}$, containing a feature space x and the probability distribution $P(X)$, where $X = x_1, ..., x_n$. A task can then be presented by $\mathcal{T} = \{\mathcal{Y}, f(x)\}$, 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 $y_i \in \mathcal{Y}$.

Given a learning task $\mathcal{T}_S$ from source domain $\mathcal{D}_S$, and a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$, transfer learning aims to improve the predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$. This is accomplished by using the

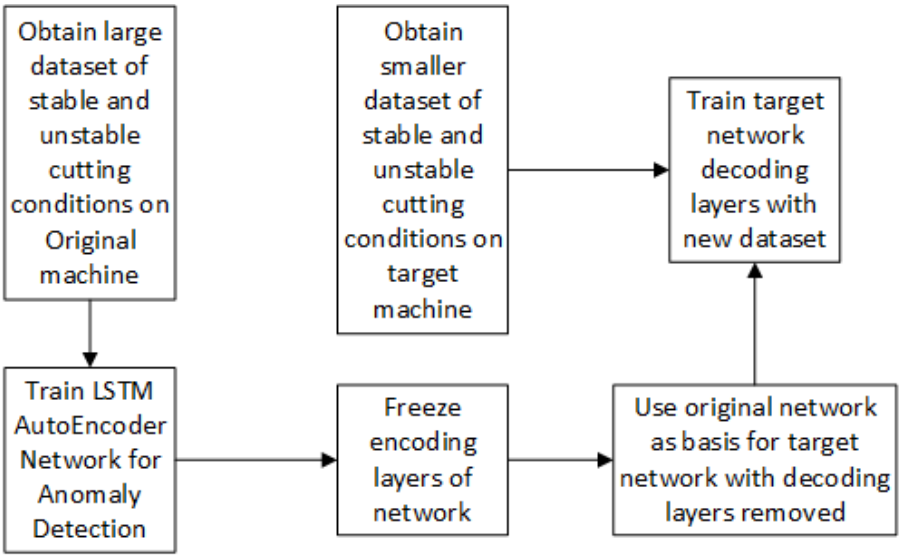


Figure 4.1: Data flow for transfer learning approach

latent knowledge from $\mathcal{D}_S$ and $\mathcal{T}_S$. In most cases the size of $\mathcal{D}_S$ is much larger than $\mathcal{D}_T$ [185].

In practise transfer learning is often accomplished by first developing a deep model based on the source domain data set. Once the deep model has been trained, 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 [15].

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. To evaluate the overlap between the sets, a metric known as the Maximum Mean Discrepancy (MMD) is often used.

4.2.1 Maximum Mean Discrepancy and transfer learning

For transfer learning to be successful, there must be similarity between the source and target sets. Without this similarity, the knowledge from the source set either is not useful, or will lead the network to an incorrect solution.

In early works, this analysis was completed by performing complex statistical tests comparing the similarity between the training data in both the source and target domains [12]. In this approach, a series of inequalities must be solved to show that there is a positive rate of learning from one network to another. Solving these inequalities is done by representing the output of the target and source neural network as a linear combination of the output of the previous iteration of the target and source network, and an input $a_i p_i$, where p_i is the number of appearances of a data pattern x and $a_i = x_i^T x_i$. The user then must solve seven different inequalities to show that the data has a positive rate of learning, and thus transfer learning can occur. These inequalities are complex to solve, and become only more complicated as the dimension of the problem increases.

More recently, a statistical measure known as the Maximum Mean Discrepancy (MMD) has been used, as it is far easier to compute, and provides equally accurate results [47]. The MMD metric was first described by Gretton [47] and has been used to evaluate the feasibility of transfer learning [204][106]. 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 $\phi(*)$ that maps the distributions to a feature space F. The MMD is calculated as

$$MMD^2(X, Y) = \|\mu_X - \mu_Y\|_F^2 . \tag{4.1}$$

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

This test is visualized in two dimensional space in Figure 4.2. From this figure we can see that if the mean is small, then we can conclude that the data is likely from the same, or similar, underlying phenomena, assuming the data from both sets is generated from a similar type of distribution (e.g. both Gaussian distributions versus one Gaussian and one bi-modal).

This expression has a straightforward description but is often difficult to implement, especially in higher dimensional space. Thus, we use an empirical estimation by representing the system with a kernel $k(x_i, x_j)$. There are many choices one can make for the kernel,

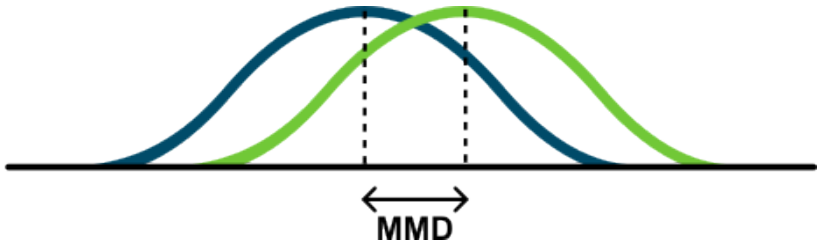


Figure 4.2: Two-Dimensional visualization of MMD metric

but the most common choice is a normally distributed Gaussian kernel. This estimation of the MMD can then be described as

$$MMD^2(X,Y) = \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(x_i, x_j) \tag{4.2}$$

$$-2\frac{1}{m.m} \sum_i \sum_j k(x_i, y_j) + \tag{4.3}$$

$$\frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(y_i, y_j). \tag{4.4}$$

In its practical implementation, the MMD score outlines the similarity in response between two networks to the same input signal; this similarity response is explained in greater detail in Section 4.3.3.

Given its wide acceptance in the literature, and ease of use, we use the MMD score as an indication of whether or not a set of data meets the requirements for transfer learning. From our experiments, we see that if the normalized MMD score is less than 0.1, then the system meets the requirements for transfer learning.

4.3 Transfer learning experimental implementation

To explore the potential of transfer learning for anomaly detection in CNC machining, a series of experiments were conducted. For these experiments, an Autoencoder was developed using the stable cutting conditions from an open source data set[215]. This data was collected from a Dytran-3293A three axis accelerometer sampling at 2048 Hz on a Tongtai TMV-720A three axis CNC milling machining on FDAC steel. The experimental setup used in the paper can be seen in Figure 4.3. After this system is trained on this large

data set, we freeze the encoder portions of the Autoencoder, saving it to be transferred to another CNC machine later. We implement this approach of freezing the encoder and then training the decoder on the target data set as it yields the better results in contrast to freezing the entire Autoencoder and applying additional layers, and training the new layers on the target dataset. We believe that this improvement is because the encoder captures the latent data representation that is not machine specific, whereas the decoder allows us to pick up more machine specific features.

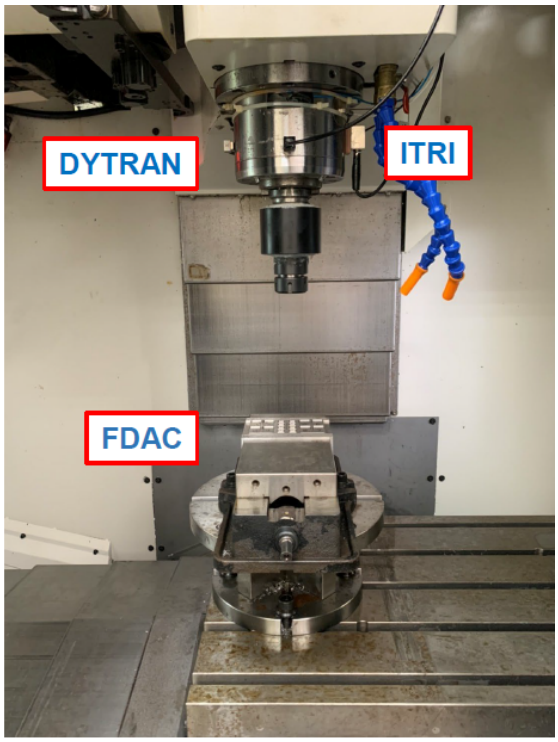


Figure 4.3: Tongtai TMV-720A setup [215]

In parallel to training the Autoencoder on the open source data set from the Tongtai system, we conducted cutting experiments on two different CNC machines on a variety of materials. These experiments were completed on a Makera X-Carve and a Roland MDX-40A using a WitMotion three axis accelerometer sampling at 200 Hz. This experimental setup can be seen in Figure 4.4 Cutting experiments were completed at various depths and spindle speeds on machining wax and wood. These experiments consisted of a mix of stable and chatter cutting conditions. A sample acceleration signal from a cutting experiment completed on the Roland MDX-40A on wood can be seen in Figure 4.5.

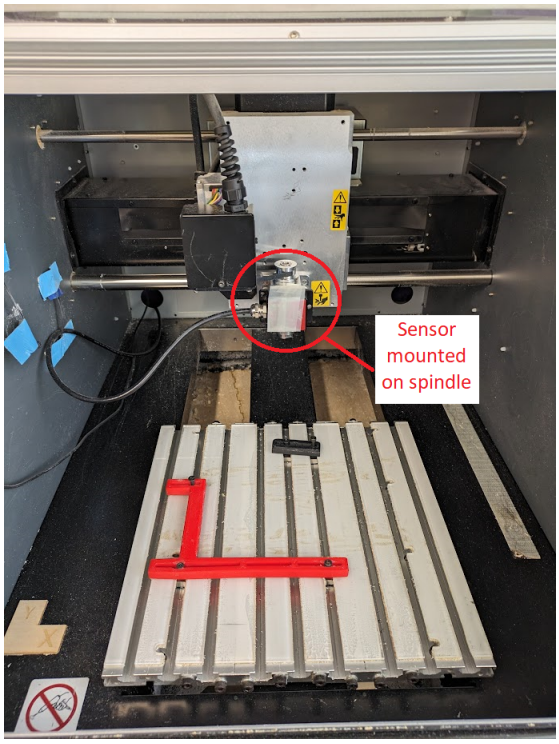


Figure 4.4: Roland MDX-40A experimental setup

The stable and unstable cuts were identified by taking the acceleration signals and performing a Fast Fourier Transformation (FFT) to obtain the frequency domain representations. As outlined in the literature review, stable and unstable cutting conditions can be separated manually by comparing the frequency peaks from the cutting experiments. This process was applied to separate the cutting conditions and create classification labels for training the Autoencoders. These labels were validated by comparing the labeled data points with video recordings of the cutting conditions and visual inspection of the parts and signals.

With the training data, testing data and ground truth information obtained, the transfer learning algorithm was applied. In this proposed approach, we train the Autoencoder with the stable cutting information from the Tongtai 3-axis CNC machine. We then freeze the layers of the encoder and transfer the information to a new network on the target machine, in this case, the Roland MDX-40A. We then re-train the decoder with the data from the target machine. The training was completed using the same configurations developed in Chapter 3, and can be described algebraically as follows: Begin by training an

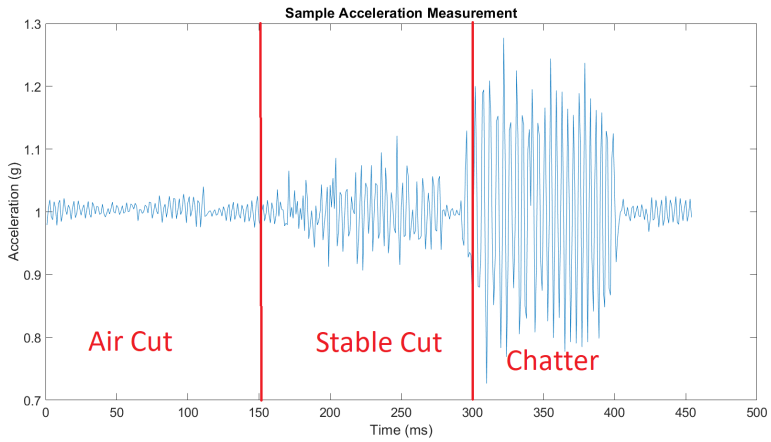


Figure 4.5: Sample Acceleration Signal

Autoencoder on the Tongtai TMV-720A data described as

$$f_{TMV}(g_{TMV}(x)) \tag{4.5}$$

then freeze the encoder $g_{TMV}(\cdot)$ and construct a new decoder and Autoencoder for the target system, the Roland MDX-40A

$$f_{MDX}(g_{TMV}(x)) \tag{4.6}$$

re-train the decoder on data from the Roland MDX-40A, aiming to minimize reconstruction error $\hat{x} - x$.

By implementing this approach, we reduce the total number of training parameters from approximately 45 000 to approximately 22 000, with the original network being trained on 200 000 data points and the transferred network trained with only 4200 data points, an 85% reduction in required data. These data points were collected from tests completed on wood with varying spindle speeds and depths of cut ranging from 3.0mm to 7.0mm.

The system was tested using the data from the input signal seen in Figure 4.5. For completeness, the original network from the Tongtai was provided the same input signal, with the resulting output seen in Figure 4.6. In this figure, the blue line indicates a stable cut, the red line a unstable cut and the green line is the threshold between the two. As this figure shows, the stable and unstable cutting signals are nearly indistinguishable from one another.

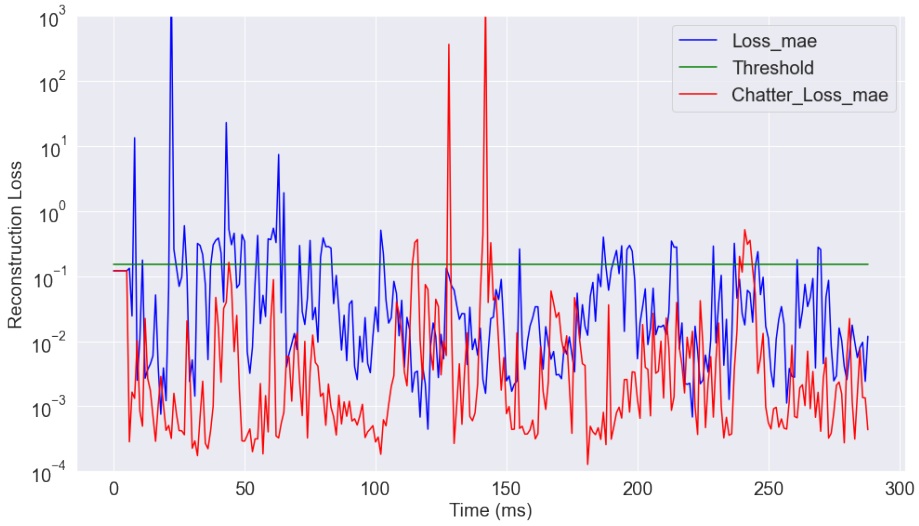


Figure 4.6: Reconstruction Plot generated without Transfer Learning

These same input signals are then applied to the system trained with transfer learning. The output from this test can be seen in Figure 4.7. As this figure shows, there is now a clear distinction between the stable and unstable signals. Furthermore, the network is able to make a clear indication when the signal moves into the unstable region. This is verified when we compare the output from the network, to the ground truth in Figure 4.8.

To obtain a numerical metric for the performance of the system, we calculate the true positives, false negatives, false positives and true negatives of the network. The results of the un-transferred network can be seen in Table 4.1, and the network trained with transfer learning in Table 4.2.

The un-transferred system has a precision of 57.74%, a recall of 37.52% and an accuracy of 72.49%. The system trained with transfer learning has a precision of 69.86%, recall of 74.85 % and an accuracy of 82.43%. These results were comparable to experimental tests found in other works[204], which indicated similar accuracy rates in other systems trained with transfer learning. It is also worth noting that the accuracy and precision are somewhat low compared to other anomaly detection systems, due to the high number of false positives, but this is due to a somewhat aggressive system that classifies many stable conditions as chatter.

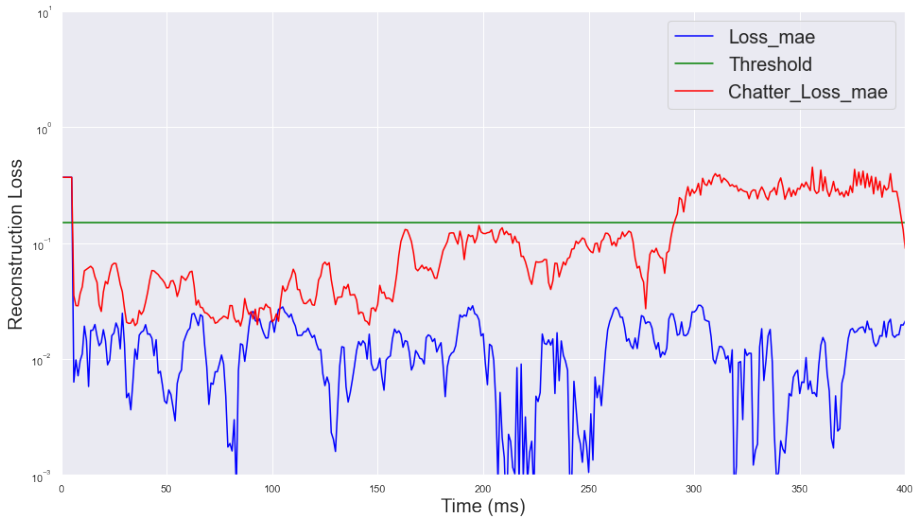


Figure 4.7: Reconstruction Plot generated from Transfer Learning

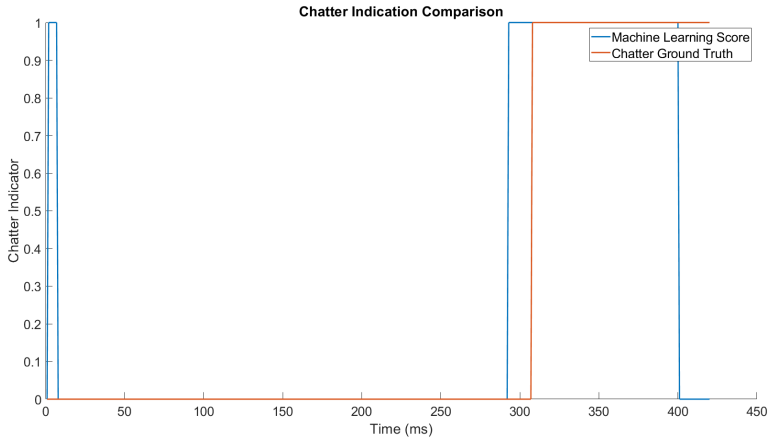


Figure 4.8: Chatter Indicator Comparison

4.3.1 Closed loop implementation of anomaly detection using transfer learning

The experimental implementations shown in Section 4.3 were completed offline, meaning that the cutting experiments were completed while the data was being recorded and then analyzed afterwards to see if chatter could be detected. This approach allowed us to develop

Table 4.1: Confusion Matrix for Anomaly Detection without Transfer Learning

Confusion Matrix	Anomaly (predicted)	No Anomaly (predicted)
Anomaly (actual)	194	323
Not Anomaly (actual)	142	1031

Table 4.2: Confusion Matrix for Anomaly Detection using Transfer Learning

Confusion Matrix	Anomaly (predicted)	No Anomaly (predicted)
Anomaly (actual)	387	130
Not Anomaly (actual)	167	1006

a ground truth to compare the accuracy of the system.

With the verification of the anomaly detection system completed, we proceeded to extend these tests to online verification with closed loop feedback. This task was accomplished by implementing the controller to react to real time data and then increasing the spindle speed by a fixed amount. Although more complex and intelligent control techniques could have been implemented, they were omitted due to being out of the scope of this work.

These closed loop tests were completed on the Roland MDX-40A three-axis CNC milling machine using the WitMotion three-axis accelerometers sampling at 200Hz. Cutting experiments were completed on wood with the end mill at a 1/8“ depth of cut, completing straight cuts. These cutting experiments began at 6000 RPM, which is known to cause chatter from previous experiments.

To implement closed-loop control, the Autoencoder trained on transfer learning was integrated with the CNC controller. This control strategy had the anomaly detection system read sensor data in real-time and then send a signal to the machine controller if an anomaly, in this case chatter, was detected. The CNC controller would then increase the spindle speed by 100 RPM when this signal was received. This choice was made since it is known that the system is stable at 7000 RPM at 1/8" depth of cut on wood for this machine, and more complex control action was not possible due to limitations in the CNC controller. A more aggressive change could have been chosen, but given that this was an arbitrary controller, it was decided to make a conservative choice.

An example of the acceleration signals from a cutting experiment can be seen in Figure 4.9 where no corrective action is taken on the top subplot, and corrective action is taken on the bottom subplot. The figure shows the acceleration in the X, Y and Z axis, in orange and green. As the figure shows, without corrective action the system continually experiences large unstable conditions, but with corrective action the system is able to quickly stabilize the forces observed.

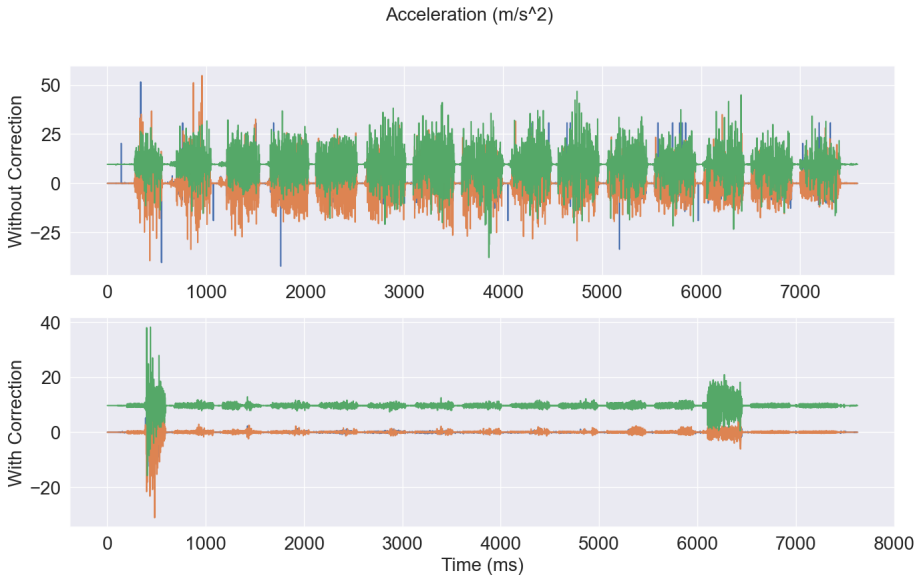


Figure 4.9: Accelerometer signal from closed loop cutting experiment

We can further see the impact of the anomaly detection algorithm in Figure 4.10 where the output of the anomaly detection system, seen in blue, is overlaid on top of the acceleration signals. As the figure shows, the system quickly captures the presence of chatter and is able to send a signal to take corrective action within several milliseconds. We can see

the resulting control action in Figure 4.11 that has the chatter indicators and the spindle RPM plotted.

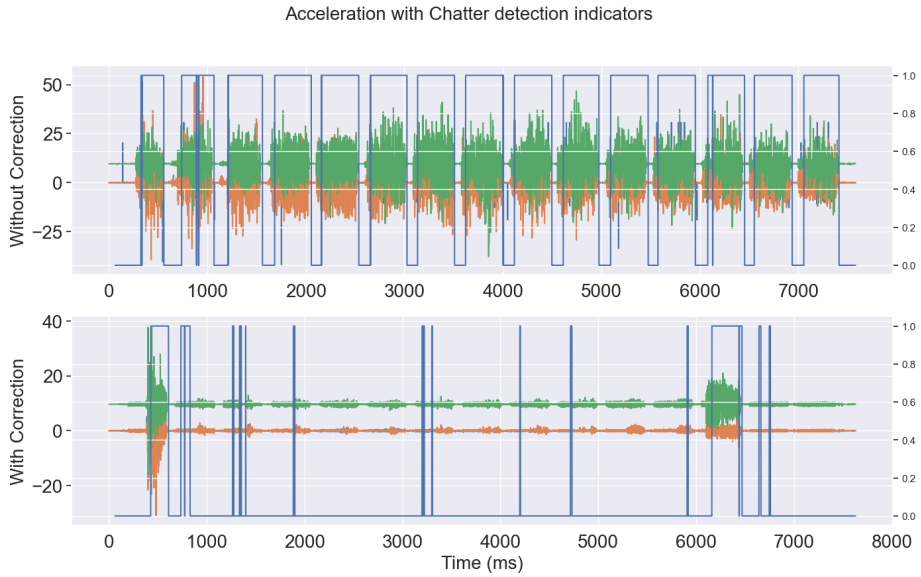


Figure 4.10: Accelerometer signal with anomaly detection indicators

The figures show that the system is able to detect the presence of chatter, and quickly apply corrective action. This allows the CNC milling machine to complete stable cuts for the duration of the cutting experiment, with minor issues detected. This result is promising when we compare the acceleration signals between the uncorrected and corrected systems, showing that the system can be brought to a stable condition in real time, and that the proposed method is not excessively computational expensive or infeasible. It should be noted that the results could likely be improved with a more intelligent control action that is able to make more informed decisions, but is left for future work.

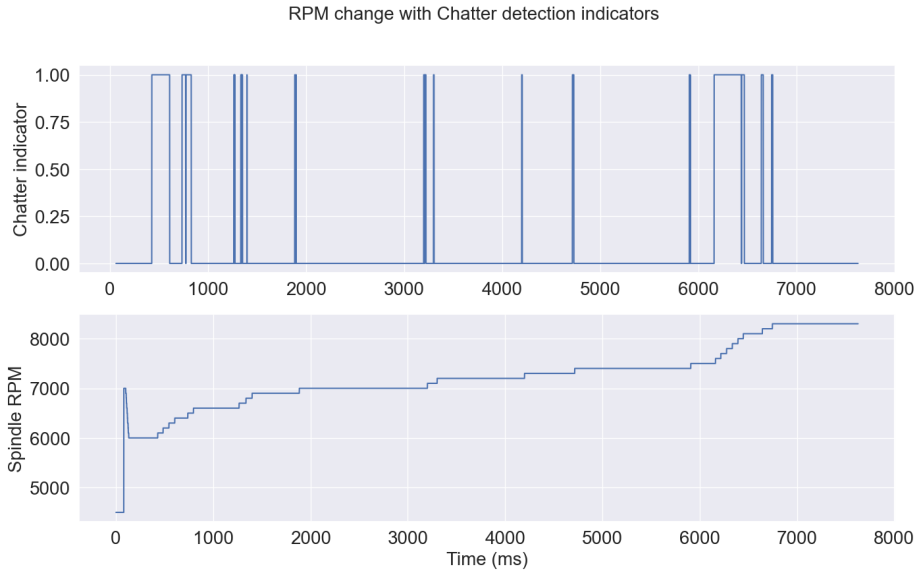


Figure 4.11: Plot of spindle speed (RPM) and anomaly detection indicator

4.3.2 Anomaly detection of broken tool bit using transfer learning

The anomaly detection algorithm proposed in this thesis has demonstrated its potential for detecting chatter conditions. To investigate its ability to detect anomalies beyond just chatter, we also investigated the feasibility of detecting broken tools.

As outlined in [20], broken tools can also be detected using an accelerometer. By using an accelerometer, we are able to implement the anomaly detection system designed previously without having to make any major modifications.

To safely and consistently test for broken tool bits, we begin with two different end mills. We purposely damage one of the cutting flutes on one of the end mills by cutting away a portion of the end mill part way up. This can be seen in Figure 4.12, where a new end mill is seen on the left, and the damaged end mill can be seen on the right. By damaging the end mill in this way we can perform test cuts with the undamaged portion of the damaged end mill and then increase the depth of cut with the damaged end mill to introduce the impacts of the damaged flutes. Although cutting dynamics do change with increased depth of cut, given the small size of the tool, the effects are minimal.

With the broken end mill, we were able to complete cutting experiments at a variety

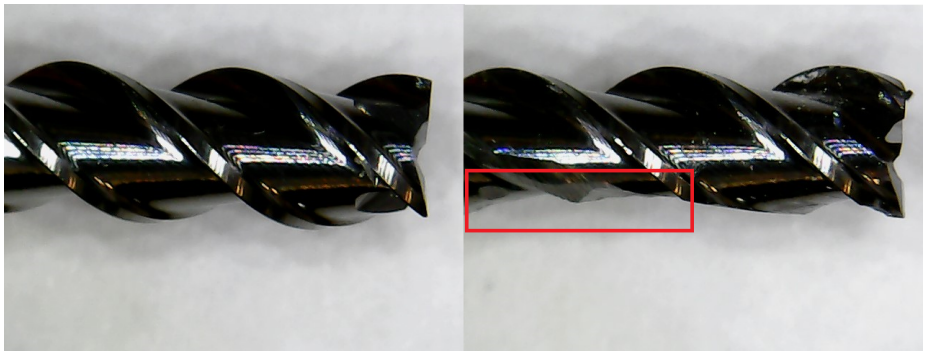


Figure 4.12: Comparison between new end mill and damaged end mill

of cutting speeds. The first of these tests was completed at 7000 RPM at a depth of cut that was known to likely be stable. The results of this cutting experiment with the new end mill can be seen in Figure 4.13. In this figure, the top plot represents the acceleration signal from the accelerator, the middle plot represents a chatter indicator using established methods described in [14], and the bottom plot represents the output from the anomaly detection system described in this thesis.

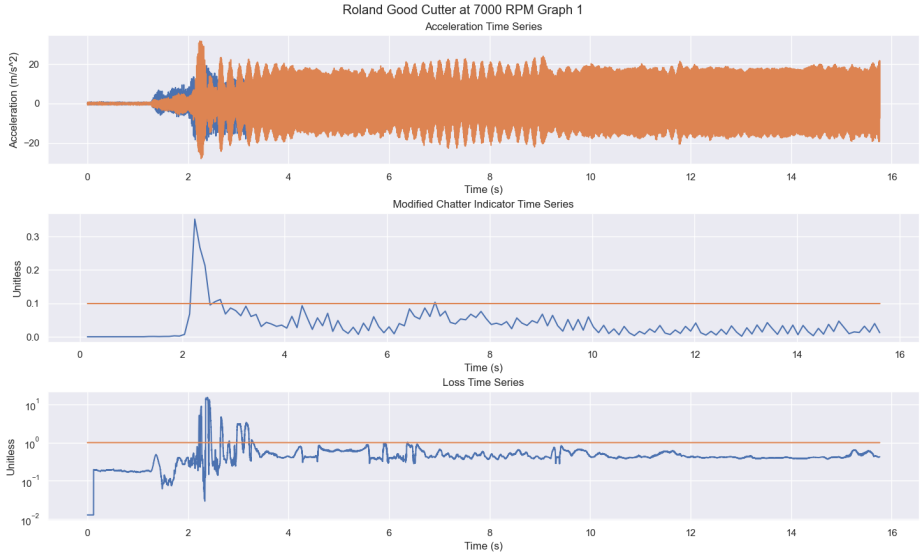


Figure 4.13: Algorithm output with new tool bit at 7000 RPM

As this figure shows, aside from some anomalies due to the initial contact between the end mill and the cutting material, the remainder of the cut is stable. This classification

is consistent between the chatter indicator and the output from the anomaly detection system.

The next cutting experiments were conducted with the same cutting profile, but using the damaged tool bit. The result of this experiment can be seen in Figure 4.14. As this figure shows, the traditional chatter indicator seen in the middle plot, is not able to detect the presence of any issues. This contrasts the output from the anomaly detection algorithm that repeatedly indicates that an anomaly has occurred.

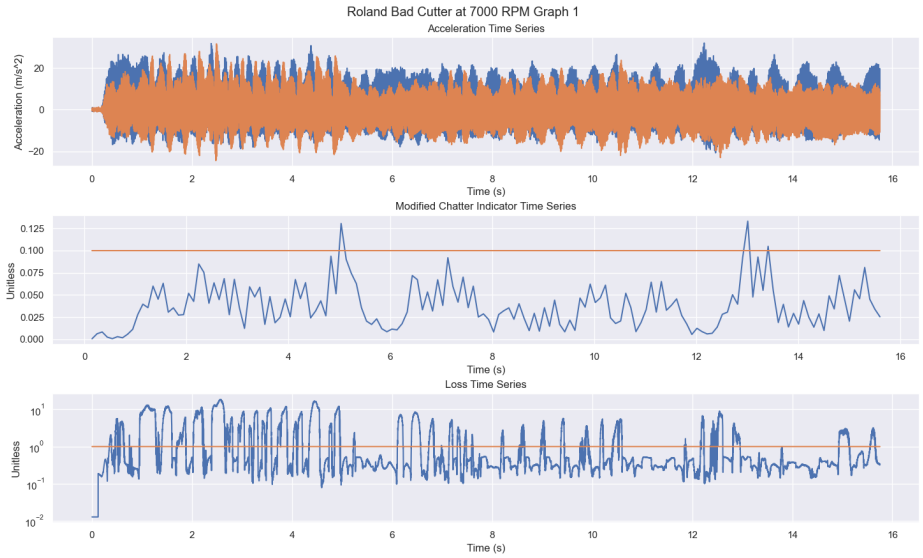


Figure 4.14: Algorithm output with broken tool bit at 7000 RPM

Following these tests, cutting experiments were conducted with the end mills in which chatter conditions would occur. The first of these experiments has the new end mill cutting at 4500 RPM. As Figure 4.15 shows, both the traditional chatter indicator and the anomaly detection algorithm indicate that an anomaly, in this case chatter, occurs at this depth of cut and RPM.

Repeating this experiment with the broken tool bit, seen in Figure 4.16, we can see that both the conventional chatter detection algorithm, and the anomaly detection algorithm are able to detect the chatter that occurs after seven seconds, but the anomaly detection algorithm detects the presence of other issues at the ten second mark.

One limitation of directly applying the anomaly detection algorithm as it is currently designed, is that it can only detect that an anomaly has occurred, but is unable to distinguish between chatter and the broken tool. To address this limitation, a proposed extension

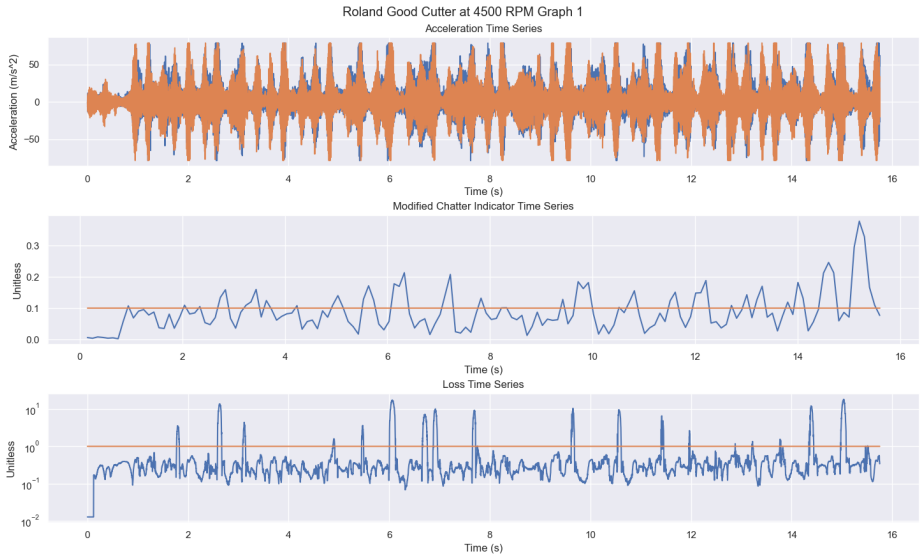


Figure 4.15: Algorithm output with new tool bit at 4500 RPM

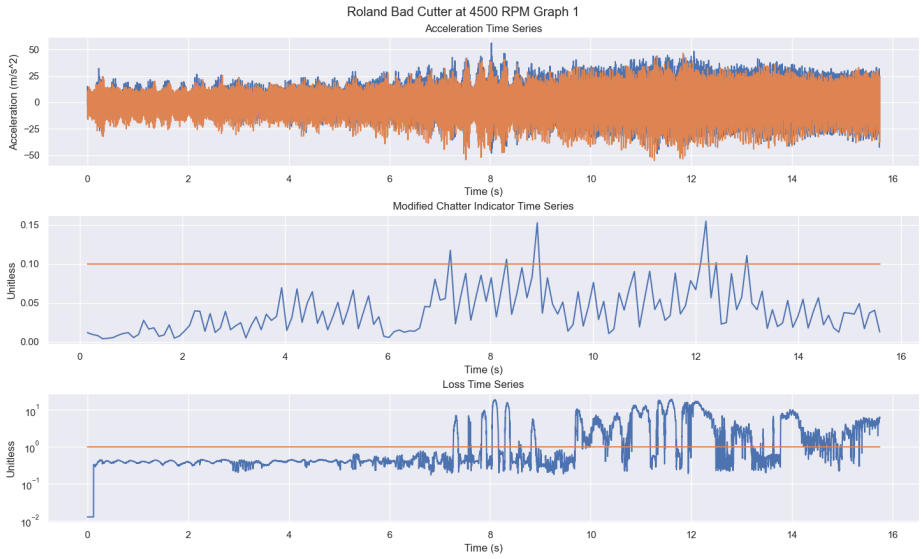


Figure 4.16: Algorithm output with broken tool bit at 4500 RPM

of the algorithm is described in Chapter 7. This extension potentially allows the system to predict the class an anomaly belongs to in addition to detecting it.

4.3.3 Meaning of Maximum Mean Discrepancy in relation CNC Machining

One of the challenges of applying machine learning techniques to CNC machining is the lack of physical intuition from the various metrics and algorithms. Often times, these machine learning techniques have been developed for problems in vision and image processing, so their relationship to a physical phenomena is not as clear. One metric of interest for this thesis is the maximum mean discrepancy (MMD), and the physical intuition that it provides.

As described previously, the maximum mean discrepancy is a characterization of the distance between distributions. Although this provides a clear statistical meaning, the clarity of this metric is lost when we begin to examine systems with a large number of features or complex embeddings.

To gain a deeper understanding of the MMD score, and to determine its relationship to transfer learning in CNC machining, a series of experiments were conducted on a variety of machines with a few test materials. These experiments were conducted with support of our industry partner, Hurco Companies Inc, at one of their beta tests sites. This beta test site is a medium sized job shop that has a variety of different Hurco machines.

The cutting experiments were completed on a series of machines within the same family of Hurco machines, but varied in some structural dynamics, as well as size. The majority of tests were completed on a direct drive Hurco VMX42Di, with additional tests completed on the belt driven Hurco VMX42i, as well as the large bed VMX6030. Several test networks were created from the data from these machines, and the resulting outputs were compared. These test networks are summarized as follows:

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

These test cases capture many of the different dynamics of the machines and answer questions relating to transferring information from one set of conditions to another. Case one studies the impact of transferring networks from two machines that have similar structural design, but are from different manufactures with different drive mechanisms. Case

two evaluates the impact of transferring networks amongst two machines from the same manufacturer with similar geometries and structural dynamics, but a different drive system. In addition, these tests were repeated three times, on three different machines to evaluate the repeatability of the methods. Case three evaluates transferring from the same manufacturer to a similar but larger machine. Lastly, case four evaluates a system trained on one material and implemented on another material but on the same machine. These cases represent the impacts of structural dynamics, drive dynamics, manufacturer and material impacts.

To capture the dynamics of interests in these experiments, uniaxial accelerometers from PCB Piezotronics were used. Theses accelerometers were mounted in the X and Y axes of the CNC spindle and had a ±15% sensitivity of 10 mV/g (1.0 mV/(m/s^2)), measurement range of ±500 g pk (±4900 m/s^2 pk), broadband resolution of 0.004 g rms (0.04 m/s^2 rms), and ±5% frequency range of 1.0 to 10000 Hz. This data was collected using a LabJack T7 DAQ at a sample rate of 100 000 samples per second. This data was streamed to a laptop computer that analyzed the data using a Python script. The data acquisition setup can be seen in Figure 4.17 and the mounting of the system can be seen in Figure 4.18.

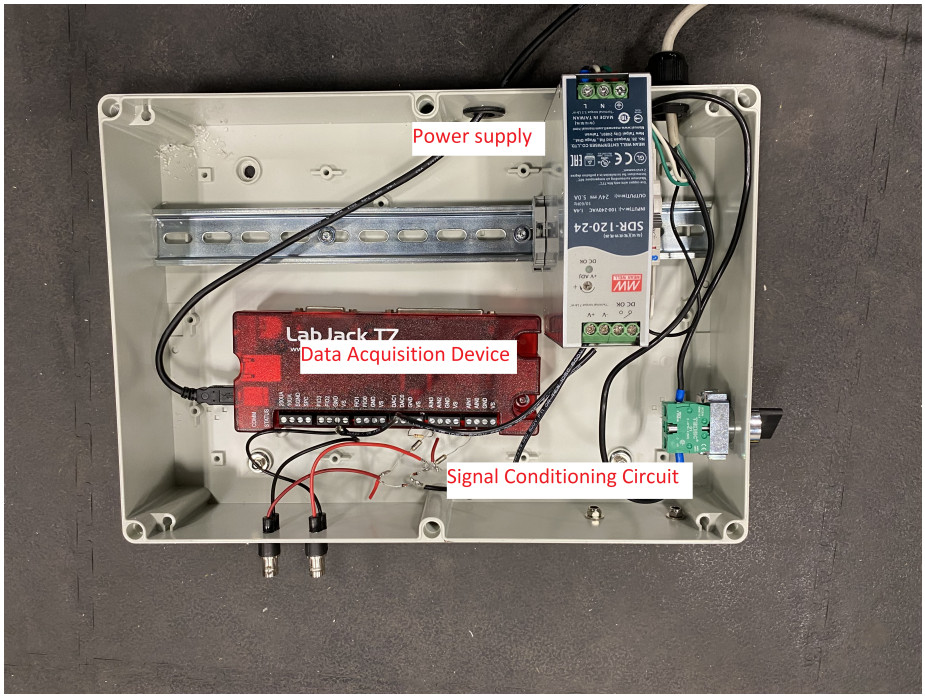


Figure 4.17: The inside of the plastic control box.

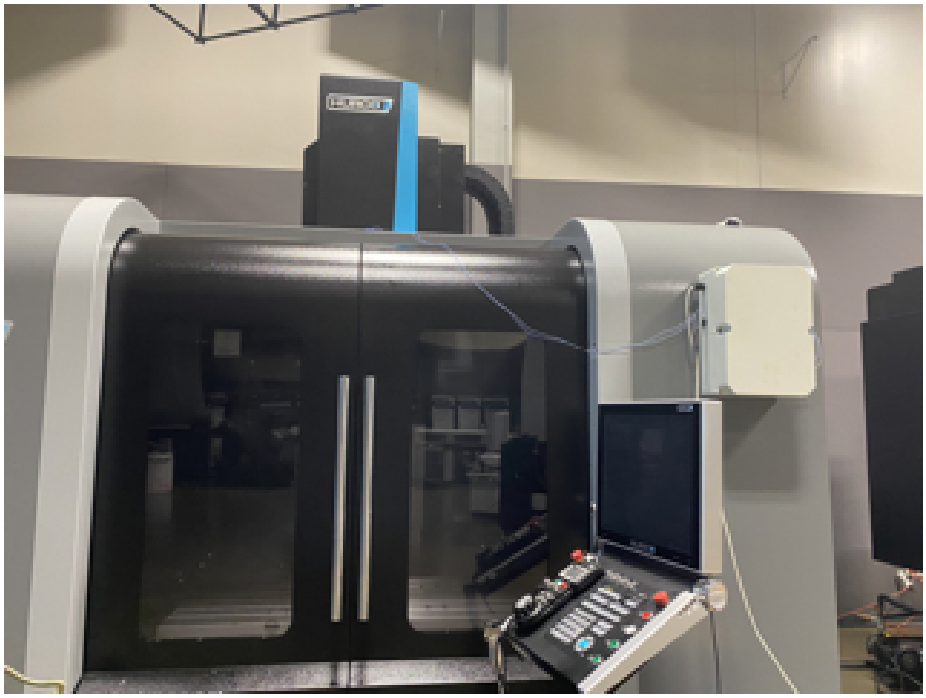


Figure 4.18: The plastic control box mounted onto a Hurco VMX42Di CNC machine.

To establish a ground truth for the chatter indicators, the chatter detection methods developed by Brecher[13] [14] were implemented. This method involved taking the piezoelectric accelerometer values, performing initial filtering to denoise, and then numerically integrating the acceleration signal from the accelerometers. This integration was used to calculate the displacement of the headstock assembly in both X and Y directions. The trajectory of the headstock was plotted using the displacement time series, and then used to calculate the the standard deviation of the headstock's trajectory. This standard deviation is used to estimate the position of the tool in the spindle, which ultimately is used to calculate the chatter indicator.

To determine the impact of the MMD score we begin by developing a series of networks, using the previously described methods, from data obtained on a Hurco VMX42Di. We then transfer this network to each of the different test conditions and use the training data to obtain the MMD score between the source and target training sets.

We then take the acceleration signal from one of the test cuts, seen in Figure 4.19 and apply it to the anomaly detection systems designed on the various networks. The accuracy of these networks is obtained by comparing the output to the ground truth obtained using

Brecher's method. The results of this comparison can be seen in Figure 4.20.

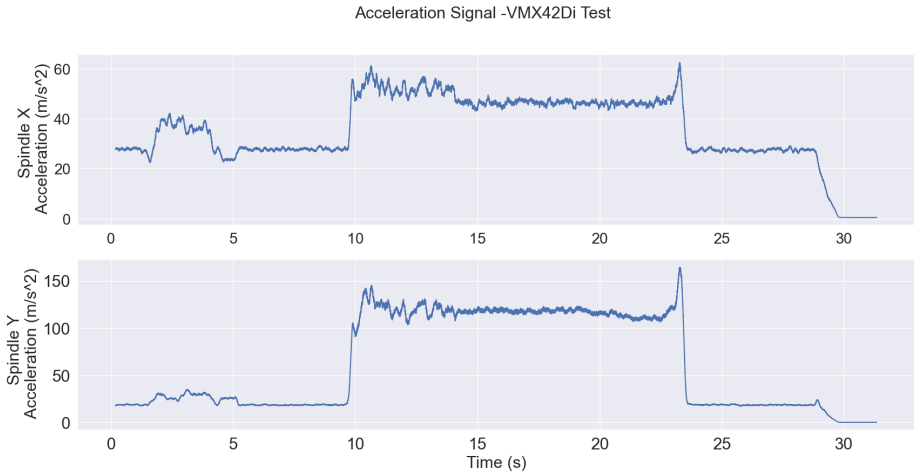


Figure 4.19: Hurco VMX42Di Acceleration Signal

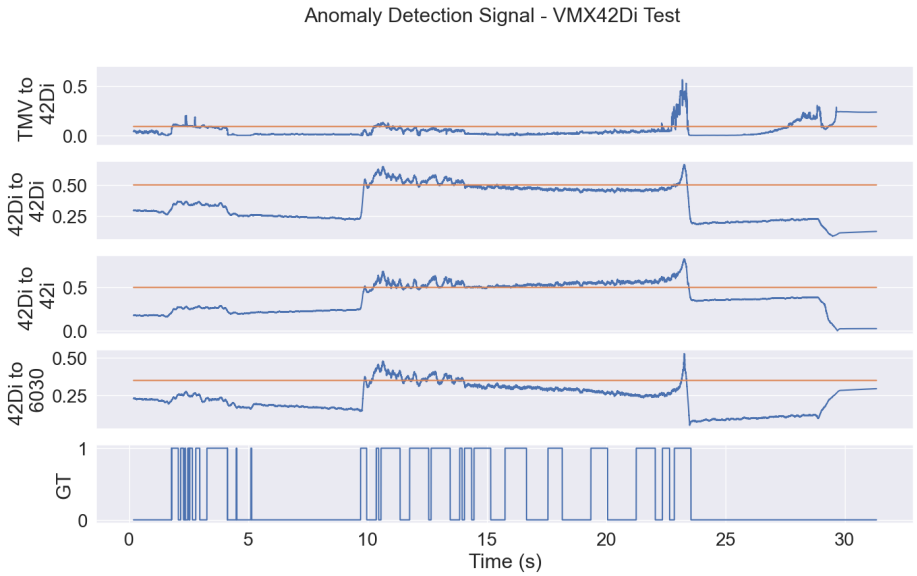


Figure 4.20: Hurco VMX42Di Test comparing transfer learning from different sources

The results of the test completed on the acceleration signal from the Hurco VMX42Di can be seen in Table 4.3. In this table we compare the MMD score, minimum learning loss, the cross entropy and the network accuracy. These metrics were chosen as the learning loss

allows us a numerical metric to evaluate the network's relative ability to learn, the cross entropy is another commonly used metric for comparing distributions and the accuracy is a measure of the performance of the network. It should be noted, that there is no learning loss for the case of the VMX42Di with the aluminum test, as this network was not re-trained.

Table 4.3: 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	55.62×10^{-2}	0.8×10^{-2}	0.14	61.58 %
VMX42Di to VMX42i	15.31×10^{-2}	1.33×10^{-2}	0.06	83.00%
VMX42Di to VMX6030	37.59×10^{-2}	20.96×10^{-2}	0.09	98.03%
VMX42Di on VMX42Di Al	5.67×10^{-2}	N/A	0.04	86.39%

From reviewing the table results alone, it is difficult to draw a conclusion about the 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 see that the MMD score follows the same trend as the cross entropy score, but neither has a strong correlation between the learning loss or 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.

As Figure 4.20, Figure 4.25 and Figure 4.27 shows, 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 Figure 4.21, with the results of these networks seen in Figure 4.22. 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 [160]. This result is confirmed when we evaluate the performance of the various networks seen in Figure 4.22. 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 more reliably, with the VMX42Di tracking the signal the most accurately.

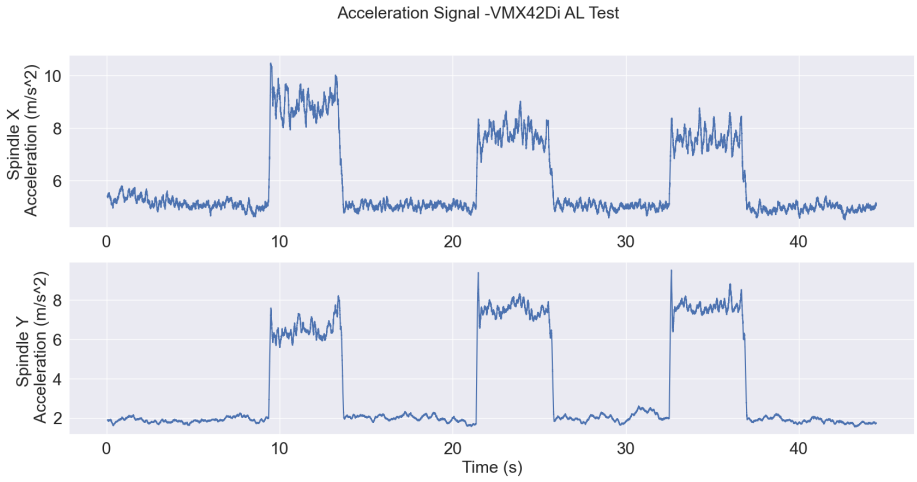


Figure 4.21: Hurco VMX42Di Aluminum Acceleration Signal

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 4.4, and the resulting system outputs can be seen in Figure 4.23. 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.

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 Figure 4.24, Figure 4.25,

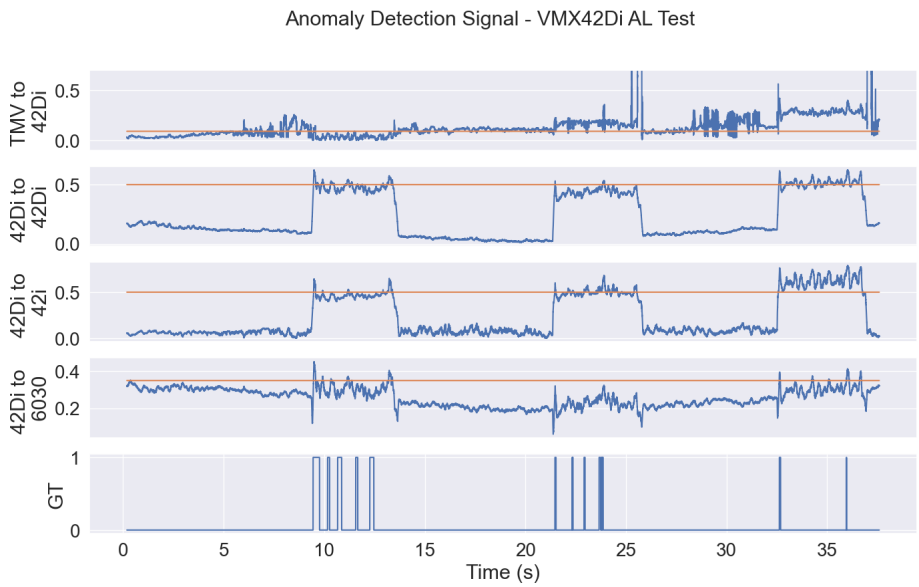


Figure 4.22: Hurco VMX42Di Aluminum Test comparing transfer learning from different sources

Table 4.4: 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	39.58×10^{-2}	4.02×10^{-2}	95.58 %
VMX42Di to VMX42i 2	3.53×10^{-2}	4.04×10^{-2}	88.08%
VMX42Di to VMX42i 3	22.28×10^{-2}	3.04×10^{-2}	97.05%

Figure 4.26 and Figure 4.27. These discrepancies also align with the structure of the mechanistic models, as described in Section 2.2.1. 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.

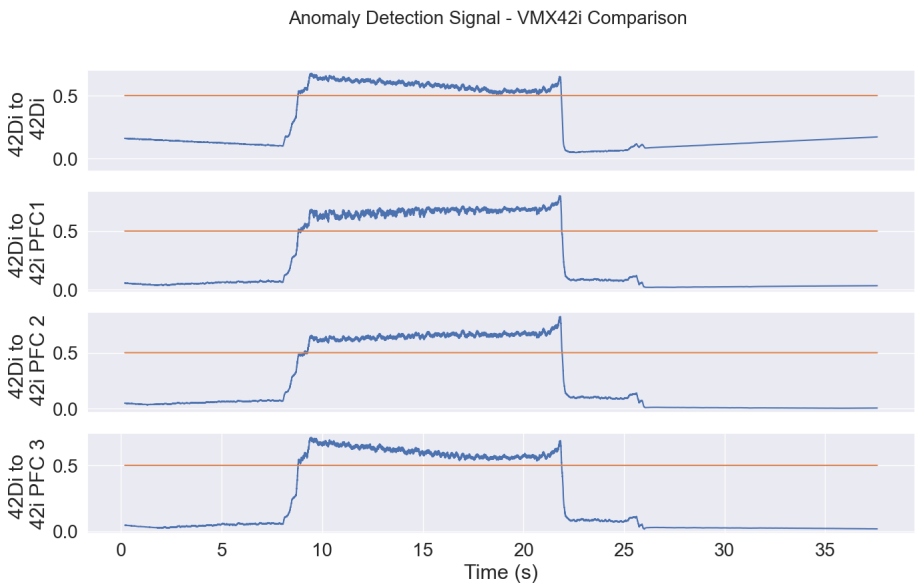


Figure 4.23: Hurco VMX42i Comparison of transfer learning to the same model of machine

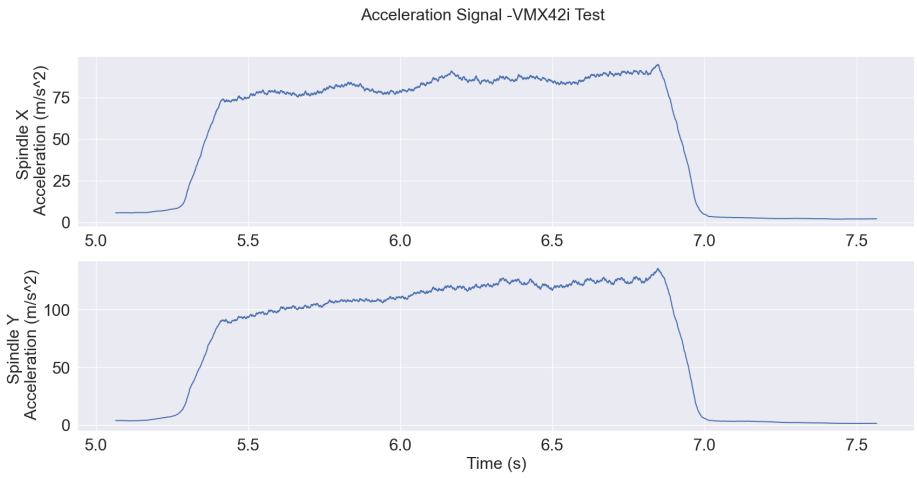


Figure 4.24: Hurco VMX42i Acceleration Signal

These results are also consistent when we evaluate the structural differences in the machines. Following the approach detailed in [63], 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. An example of this analysis can be seen

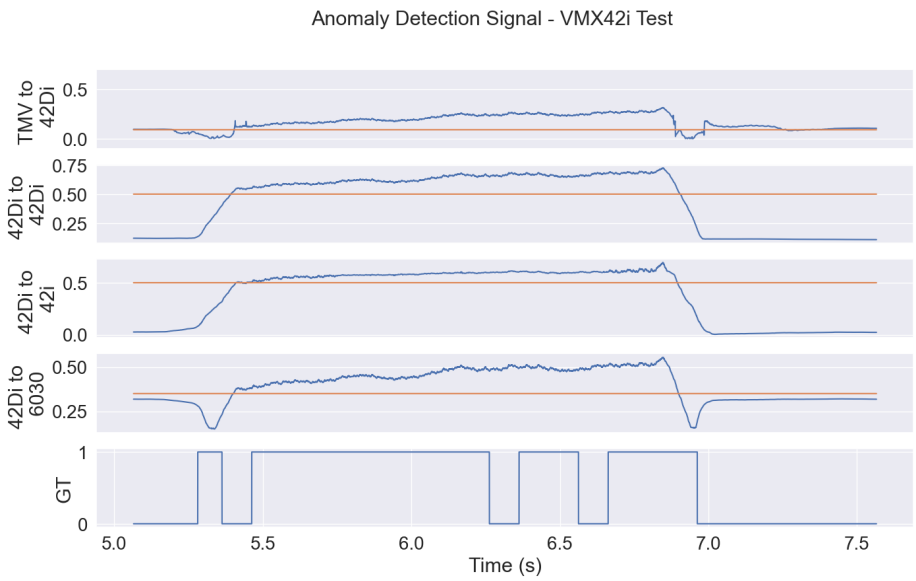


Figure 4.25: Hurco VMX42i Test comparing transfer learning from different sources

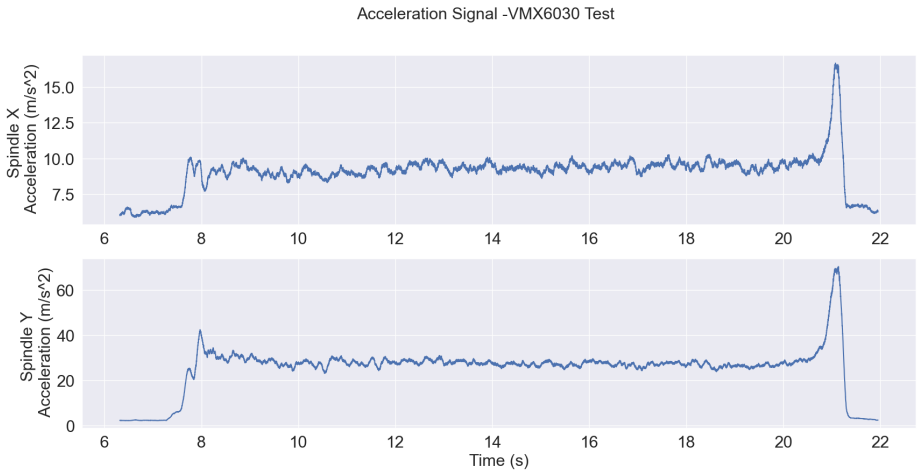


Figure 4.26: Hurco VMX6030 Acceleration Signal

in Figure 4.28 for the VMX42Di. Using finite element analysis, the frequency response of VMX42Di was determined in the X, Y and Z directions to identify the most flexible mode. This flexible mode will influence the structural dynamics of the machine through the bending and twisting of the column. Thus, if there are significant differences in the

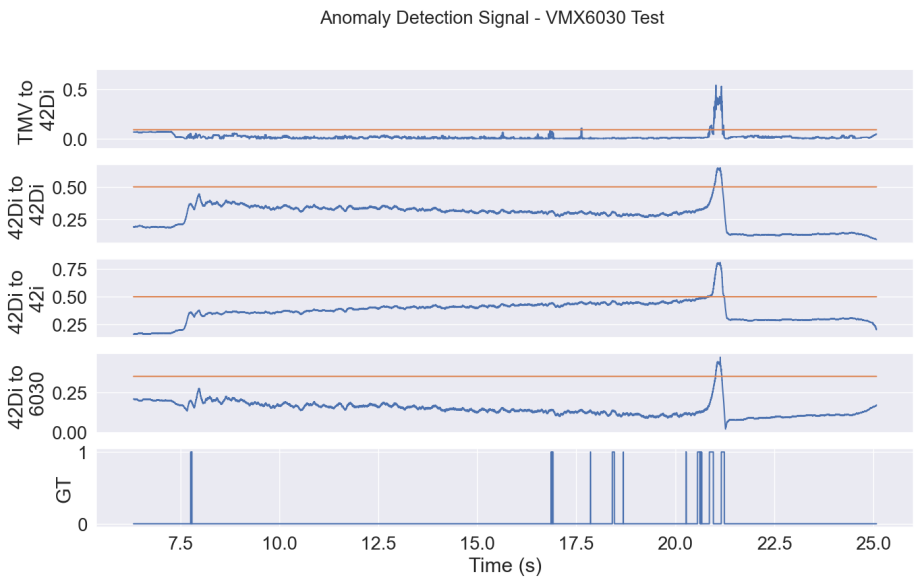


Figure 4.27: Hurco VMX6030 Test comparing transfer learning from different sources

structural components of a machine, we can expect these differences to cause the machine to respond differently. From this analysis we found that the greatest impact on the VMX42Di occurred at 74Hz, with a similar result was seen for the VMX42i at 63 Hz in Figure 4.29 and for the VMX6030 the column bending can be observed at 61Hz in Figure 4.30. This analysis indicates that all three of these systems have similar structural responses, at relatively similar frequencies.

Following this analysis, we 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, and thus has the largest impact on the structural dynamics of the machine. Therefore any differences in the z-column are likely to generate significant differences in the structural 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. This causes the column bending to occur at a different frequency and amplitude compared to the VMX42Di and VMX42i, since the only similarity between the machines is the overall geometry. From this analysis, we conclude that these differences are captured in the data, resulting in the larger MMD scores observed.

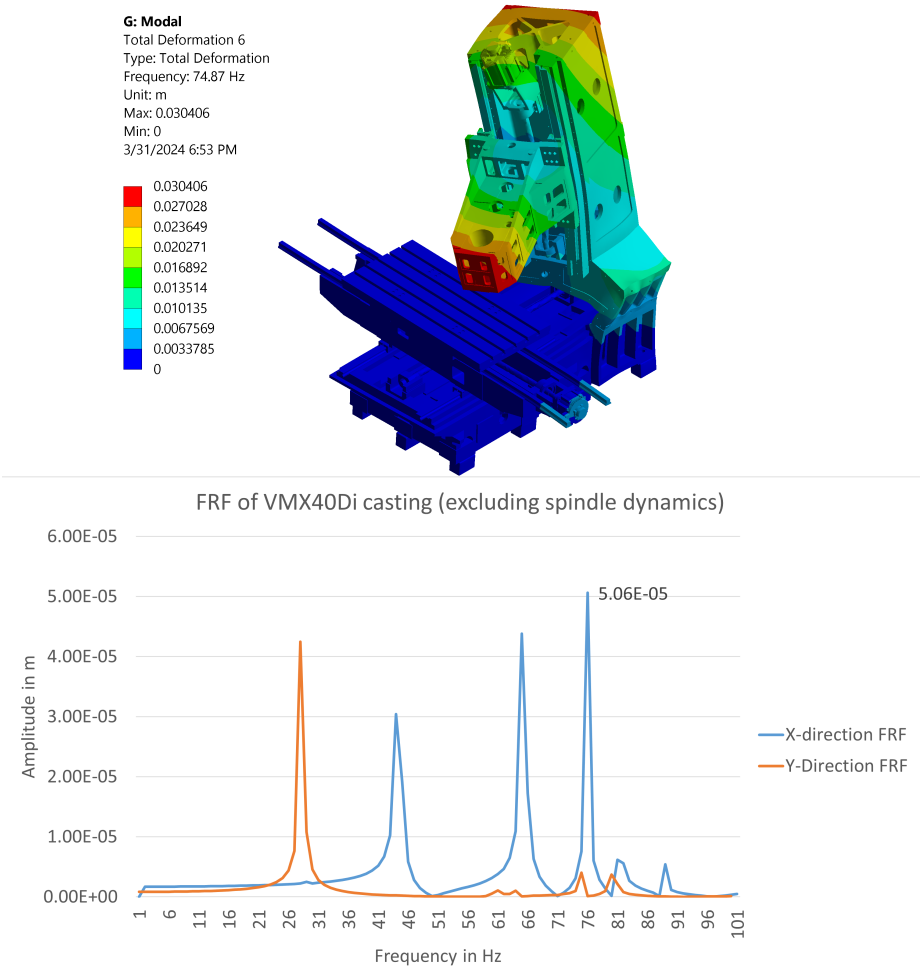


Figure 4.28: FE Analysis of Hurco VMX42Di

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, then the network transferred to the target machine will also produce a signal that outputs similar characteristics. From this, we conclude that 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.

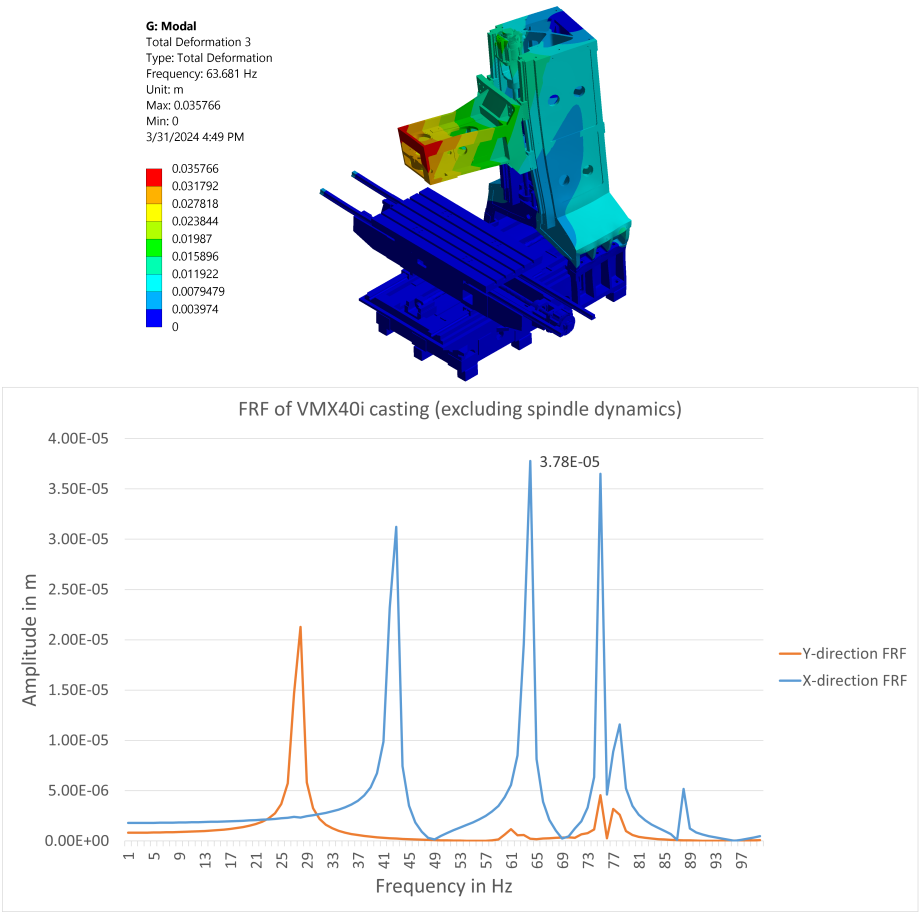


Figure 4.29: FE Analysis of Hurco VMX42i

Furthermore, if the MMD score is sufficiently small, there may not be significant benefit from transferring the network from one source to another. We see this effect when we compare the networks design on steel cutting aluminum, and the networks designed on the VMX42Di being deployed on the VMX42i. The MMD scores between these data sets is small, and the network response is so similar, that the extra effort invested to train a specific network for these cases is not worthwhile.

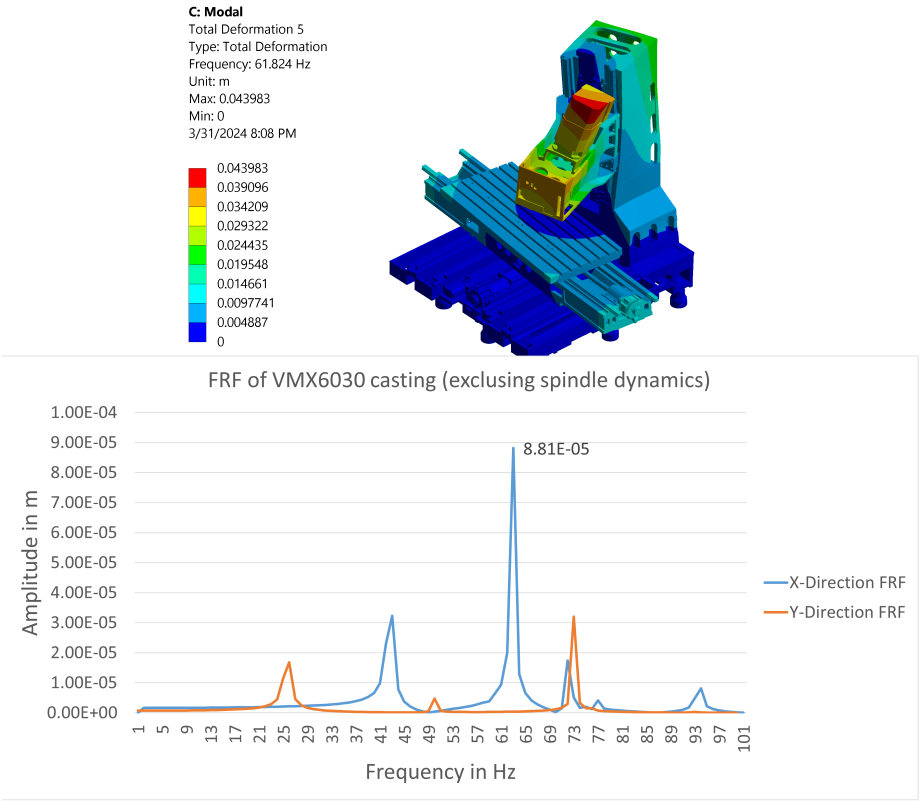


Figure 4.30: FE Analysis of Hurco VMX6030

4.4 Discussion and summary of contributions

In this chapter we propose a method of transfer learning that can be applied for anomaly detection in CNC machining. This proposed methods offers many potential contributions to the literature. These contributions are summarized as follows

- Demonstrated that transfer learning can be applied for anomaly detection in CNC milling machine when using deep learning as such as the LSTM Autoencoder, ultimately reducing the amount of data required and reducing the training time.
- Demonstrated that the anomaly detection algorithms trained with transfer learning can be used for chatter detection, but also extended to detection of broken tools.
- Verified that the MMD score can not only be used to check the viability of transfer

learning, but to provide a metric to show how similar one network will respond to another network when transferred

- Verified that if a MMD score is sufficiently low, then transfer learning may not be required, as the two networks will respond to a given input the same way (e.g. transfer learning is not required for different materials or for similar machines).

One of the challenges in implementing machine learning algorithms is the high cost associated with collecting training data. This cost often adds increased development time before an approach can be implemented. Transfer learning allows us a valuable method for quickly deploying deep networks; in addition, transfer learning allows us to re-use this training data in other situations. For instance, through our work with the MMD score analysis, we know that several of the CNC machines that we developed models for have low MMD scores and are physically quite similar. This means that we can either choose to train networks on the largest data sets and deploy them on the various machines, or we could combine all of the data to create an even larger data set. This proposition means that investments spent for data collection can likely be extended beyond a singular project or machine, reducing the overall finance commitments required to implement these advanced capabilities.

To illustrate the difference in training parameters and data required, the training parameters and data sizes for the VMX42Di, VMX42i and VMX6030 have been tabulated in Table 4.5. To train the VMX42Di, nearly a million data points were required with nearly a week of computation required to train the network. In contrast, the VMX42i and VMX6030 were both trained with nearly a third of the required data, and only a day of computation each. Given that cutting experiments cost several hundred to several thousand dollars to complete, the dramatic reduction in required training data is of substantial savings to the users.

This work and these contributions demonstrate the viability of transfer learning in applications of anomaly detection in CNC machining and are explored further in this thesis.

Table 4.5: Comparison of network training parameters with and without transfer learning

Training Configuration	**Approximate Number of Parameters** ($\times 10^3$)	**Amount of Training Data Required** ($\times 10^3$)
VMX42Di Trained from Scratch	45	1000
VMX42i Trained from Transfer Learning from VMX42Di	22	303
VMX6030 Trained from Transfer Learning from VMX42Di	22	338

Chapter 5

Incremental Learning for Anomaly Detection in CNC Machining

In this thesis we have presented an anomaly detection system based on the LSTM-Autoencoder architecture. We have presented a potential solution to reducing the burden on the user by applying transfer learning to take the knowledge from one system and transfer it to another. However, the systems trained with transfer learning only represent a snapshot in time, and does not encapsulate changes to the machine as more data becomes available. In addition, the network trained from transfer learning alone, has lower accuracy compared to a network trained from scratch.

In this chapter we introduce incremental learning as a potential tool for obtaining high performance of neural networks in anomaly detection, without the need for substantial initial data collection. This approach has the potential to reduce barriers for implementing machine learning in production environments, by presenting a solution that is capable of growing and adapting as a machine is used.

5.1 Motivation

One challenge for many machine learning algorithms is collecting enough high quality data to properly train and test a system. Spending machine time to collect training data often involves paying an operator to perform tests cuts, ultimately using material and time to produce unproductive parts. This process is a costly proposition that is often undesirable for those in production. Another issue with many machine learning algorithms, is the

data only represents a snapshot in time. For example, in tool wear, the cutting forces required can increase over time, causing changes in the observed cutting signals [229]. This limitation has a significant impact on models that may be trained in one domain, but overtime, operate in situations where the conditions have changed. Models designed only to represent new machines from the factory, may end up poorly matching machines that have been used in production for several years.

This type of problem is common for time series systems, where underlying dynamics change overtime. A well known example of this is HVAC controllers adapting to changes in temperature demands due to changes in season, building occupancy and building integrity as time changes [72]. One solution to these problems is to use an incremental learning algorithm [119]. Incremental learning, as its name implies, has the network learn in small batches over time. This approach ensures that the networks are able to incrementally learn from non-stationary streams of data, and avoid losing their previously acquired capabilities [199]. This aspect of machine learning algorithms has garnered significant attention and produced many works such as [36], [43], and [127]. However, this work is often unstructured and lacks a shared framework [199], making advancements fractured and inconsistent.

Incremental learning has been successfully applied in CNC machining for a variety of different applications, demonstrating its feasibility within the domain. However, incremental learning has only been applied for shallow approaches, such as a K-Nearest Neighbors algorithm [25], or Support Vector Machines [212] [220] [226]. The most complex types of networks that incremental learning has been applied to are fuzzy inference networks in [202], [111], [137]. These works show that the framework for CNC milling machines lends itself to incremental learning, but not how, or if, it can be applied to deep networks such as the LSTM.

5.2 Incremental learning algorithm definition

Using the framework in [198], we can classify the aforementioned problems in what is known as domain-incremental learning. In domain-incremental learning, the structure of the problem is the same, but the context or input-distribution changes. This problem definition aligns well with the problems described in Section 5.1, as we are still aiming to detect anomalies in the machining environment, indicating that the structure of the problem is the same, but the inputs of the problem may change, indicating that the context has changed. Thus, by addressing this domain-incremental learning problem, we propose a system that is capable of learning how to adapt to changes in the environment, as the machine's dynamics change with use.

5.2.1 Incremental learning problem formulation

To ensure that our incremental learning algorithms are effective, we follow the paradigm set in the highly cited paper on the Learn++ algorithm [140]. In this work, an incremental learning algorithm is defined as one that meets the following criteria:

- It should be able to learn additional information from new data.
- It should not require access to the original data, used to train the existing classifier.
- It should preserve previously acquired knowledge (that is, it should not suffer from catastrophic forgetting).
- It should be able to accommodate new categories that may be introduced with new data.

A key aspect introduced in these criteria is the concept of catastrophic forgetting. Catastrophic forgetting is described in [75] as the inability of artificial neural networks to retain old information in the presence of new information. This is a major concern in system with memory like the LSTM, as they will lose their ability to generalize beyond the most recent cases.

Another important idea that must be considered is concept drift. Concept drift occurs when networks operate in a non-stationary environment. In these non-stationary environments, the underlying natural phenomena that generates the data may change, causing the underlying data distribution to change as well [36]. These concept drift problems are a subset of the domain shift problems mentioned previously.

In general, concept drift can be described mathematically using the Bayes posterior. If we use the Bayes posterior to describe the probability of a given class as

$$P(w|x) = P(x|w)P(w)/P(x)$$

then concept drift is defined as any scenario where the posterior probability changes over time, namely

$$P_{t+1}(w|x) \neq P_t(w|x)$$

This phenomena is described in greater detail in [36].

In the context of CNC machining, this change in data distribution often occurs due to changes in tool conditions such as tool wear. Tool wear causes significant changes in the cutting forces of the CNC machines, changing by as much as thirty percent [229] [188].

As previously mentioned, these problems fall within the context of domain-incremental learning. In domain incremental learning, the algorithm aims to learn how to solve the same problem, but in different contexts. This is defined mathematically as follows: given an original input vector $x_0 \in \mathbb{X}_0$, a new input vector $x_i \in \mathbb{X}_i$ and an original context label $y \in \mathbb{Y}$, domain-incremental learning is the task of learning a mapping of the form $f : \mathbb{X}_0 \cup \mathbb{X}_i \rightarrow \mathbb{Y}$. In this scenario, the structure of the problem is always the same, but the context or input-distribution changes [123] [76].

5.2.2 Incremental learning algorithm description

The Learn++ incremental learning algorithm proposes a solution to the incremental learning problem by developing an algorithm based on the idea of ensemble learning[140] . This algorithm takes its inspiration from the Adaboost (adaptive boosting) algorithm that combines an ensemble of weak learners trained across a variety of training samples [41], described in the preliminaries in Section 2.3.4. In Adaboost the weak learners are combined by performing a majority-voting scheme to determine the final classification. Learn++ takes this idea and extends it by having the weak learners train themselves on new data, allowing the network to incrementally grow. This approach has shown considerable success and has been extended in works such as [125] and [36] .

Recently, this approach has been extended beyond multi-layer perceptron systems described in the Learn++ algorithm. The authors further developed this algorithm to create an incremental learning method for LSTM models and time series data[203]. This method, named "IncLSTM" has a strong LSTM learner and then an ensemble pool of weak learners. The various learners are fed into a shared layer and selected based on a weighted average, with weak learners being substituted over time. The IncLSTM method has shown effective performance both in simulation and experimental environments, with considerable improvements on accuracy compared to other methods.

Based on the criteria described in the problem formulation in Section 5.2.1, we propose an incremental learning strategy that only uses new data to incrementally train a network, once an initial network has been trained using an initial data set. This approach avoids catastrophic forgetting by freezing the initial network. Following the strategies outlined in [203], we first develop a strong learner using the LSTM-Autoencoder architecture in Chapter 3. Once the strong learner is trained, then a weak learner is developed using the

same architecture. The output of the two learners are then fed into a shared layer. As outlined in [203], and [141], the shared layer achieves better performance than a simple fine-tuning layer. This layout can be seen in Figure 5.1.

As time passes, or as new data becomes available, new weak learners are continually added to the pool of weak learners until the maximum number of learners is reached. After this point, when new learners are to be added to the pool, the oldest learners are removed. By removing the oldest learners from the pool, we remove data from a previous machine state that is no longer as relevant. Furthermore, the duration between the addition and removal of weak learners can be defined by the user and adjusted depending on user knowledge of the system, as well as available memory on the controller. For example, if the machine in question is constantly being used for machining high strength steel, one would expect the tools to wear quickly, thus the weak learners would be updated frequently, or vice versa.

These weak learners are trained on small data sets, normally a few cuts from a limited number of cutting operations, over a limited number of epochs. On their own, the weak learners make poor estimates of the machine state. However, strong learners spend a majority of their time fine-tuning their estimates[140]. By using an ensemble of weak learners, we are able to compensate for their weak estimates, thereby allowing faster training and less over fitting.

5.2.3 Incremental learning algorithm mathematical representation

We define the following: given sensor inputs $X_{SL} = \{x_1, ..., x_n\}$, labels $X_{SL} = \{x_1, ..., x_n\}$, and training pairs $\{x_i, y_i\}$, we train a strong learner $f_{SL}(\cdot)$ of the form described in Chapter 3. New data that is obtained during operation is then collected into the sets $X_{WL_j} = \{x_1, ..., x_n\}$, $X_{WL_j} = \{x_1, ..., x_n\}$, where j represents the $j-th$ operation. A weak learner of the form described in Chapter 3 , is then trained on this data set and defined as $f_{WL_j}(\cdot)$. The pool of weak learners is represented by the set $\mathcal{W}$.

Given these definitions, the Incremental Ensemble LSTM Algorithm (IncEnsemble) is summarized in Algorithm 1.

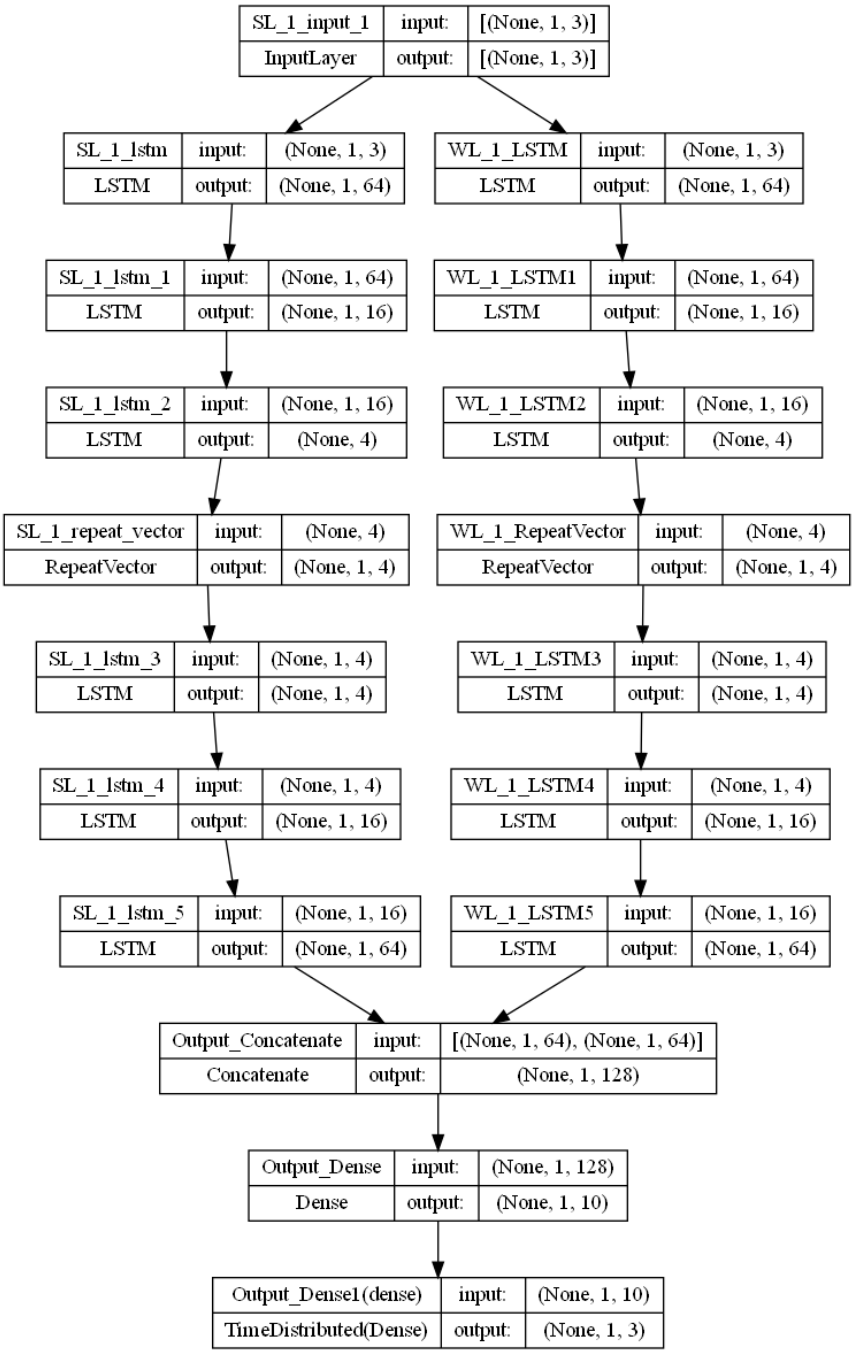


Figure 5.1: Network Structure with Strong and Weak Learner

Algorithm 1 IncEnsemble LSTM Algorithm

Let $N_{learners}$ represent current number of weak learners (WL)
Let N_{max} represent maximum number of learners
Measure IMU data from CNC machine X_{WL_j}
if $N_{learners} < N_{max}$ **then**
    Train $f_{WL_j}(\cdot)$ based on X_{WL_j}
    Add $f_{WL_j}(\cdot)$ to $\mathcal{W}$
else if $N_{learners} \geq N_{max}$ **then**
    Remove oldest WL from $\mathcal{W}$
    Train $f_{WL_j}(\cdot)$ based on X_{WL_j}
    Add $f_{WL_j}(\cdot)$ to $\mathcal{W}$
end if
Concatenate output from $f_{SL}(\cdot)$ and output from all $f_{WL_j}(\cdot) \in \mathcal{W}$
Train shared and output layers based on X_{WL_j}

5.3 Incremental learning experimental implementation

For the incremental learning algorithm, both strong and weak learners were designed using the results of the work described in Chapter 3. These choices for the learners resulted in approximately 45 000 training parameters. For the strong learner, we trained the system with 1 million data points, trained over 150 epochs using mini-batch stochastic gradient descent (SGD) in batches of ten. Once trained, the strong learner was frozen to prevent further training and catastrophic forgetting.

The weak learners were designed to use a similar structure as the strong learner, but trained on a significantly smaller data set of approximately 33 000 data points for only two epochs, while still having approximately 45 000 parameters. It was found that additional epochs did not significantly improve the accuracy of the learners, but did significantly increase the training time. As previously mentioned, the role of the weak learners was only to generate an estimate of the state of the system at that time, with the fine details to be determined with the strong learner.

The output from the strong and weak learners was combined with a concatenation layer, with no training parameters. This concatenation layer was then fed into a fully connected dense layer, with 3000 parameters. These parameters were trained each time and were not frozen. This dense layer represents a weighted average between the learners, and by not

freezing this layer, we are able to dynamically shift the weight to whichever learner most accurately represents the system dynamics at that time.

One of the important factors in the incremental ensemble learning algorithm is the maximum number of learners that are added to the pool of weak learners. To determine an appropriate number of weak learners, experiments were done with the same input data but varying the number of weak learners. Each weak learner was trained using a different data set, representing a different instance of time, but the same training conditions were provided. For simplicity, each set of data was from a different sequential cutting operation, but data from an extended time duration could also have been used. The learning loss for these experiments can be seen in Figure 5.2, with the original system in orange and the ensemble system in blue. As the figure shows, even adding one learner significantly improves the learning loss of the system, while there is no perceptible change from the original system. This indicates that the additional learners allow the network to improve its reconstruction ability, while the original system is no better given the same scenario. However, given our system, we did not see significant performance increases after the addition of four weak learners, but did note an increase in required memory.

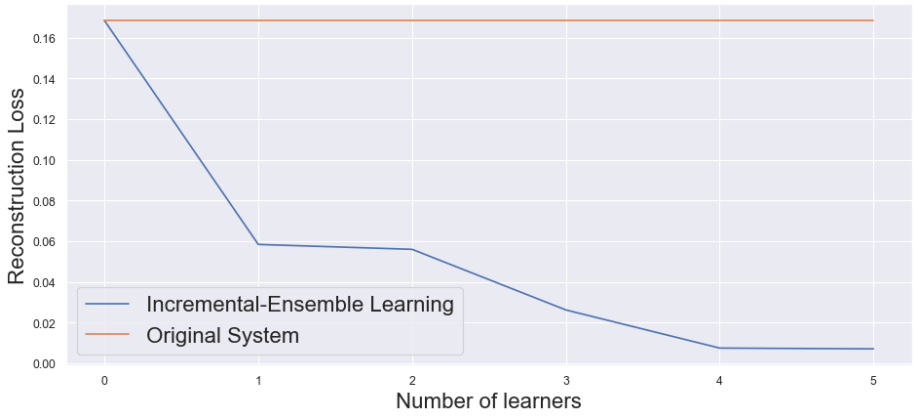


Figure 5.2: Reconstruction Loss with Varying Numbers of Ensemble Learners

To obtain training data for the incremental ensemble learning algorithm, and to validate its efficacy, experiments were conducted using the same configuration seen in Figure 3.1, and described in Section 4.3.3. This testing involved accelerometers mounted on the Hurco VMX42Di three axis CNC mill.

Cutting experiments were completed on 1020 mild steel with variations of spindle RPM, depth of cut and tool engagement. These experiments varied the spindle RPM between

3000 to 6000 RPM, with the depth of cut ranging from 0.25“ to 1.25” and from 25% to 100% tool engagement using a straight tool path. All other parameters were kept constant to ensure that the experiments could be distinguished only by the specific parameter being varied. This large variety of experiments produced a great diversity of cutting conditions for the network to be trained and tested on.

The incremental learning algorithm was tested by evaluating it using a cutting experiment that contained both stable and unstable cutting conditions. The acceleration measurements in the X and Y direction can be seen in Figure 5.3. The outlined sections align with when audible chatter was heard by the operator, with the corresponding ground truth seen in Figure 5.4. The ground truth for these experiments was also developed using the methods outlined by Brecher.

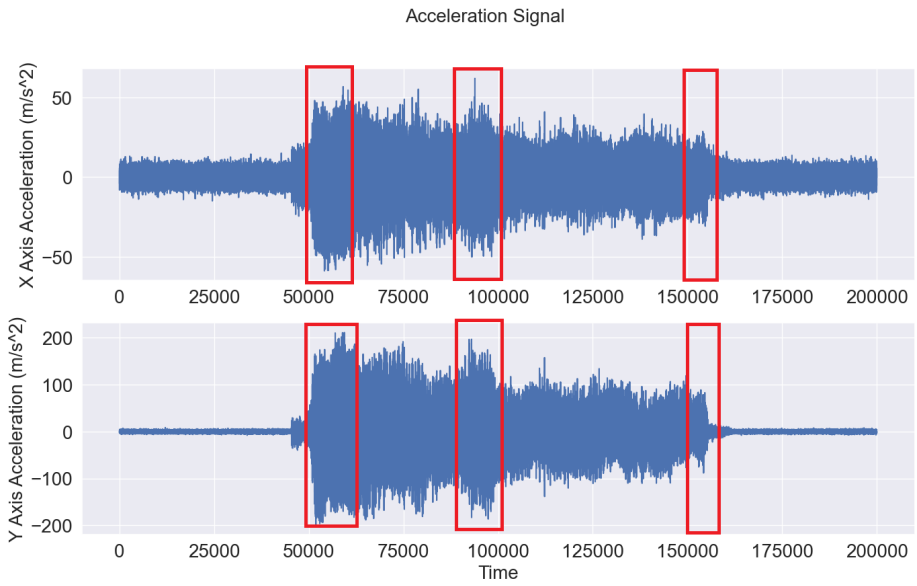


Figure 5.3: Acceleration from test signal used for testing incremental learning algorithm

This experimental data was then tested on an LSTM-Autoencoder network that was directly trained on training data from the machine, an LSTM-Autoencoder network with the incremental ensemble learning algorithm, an LSTM-Autoencoder that was trained using transfer learning from the Tongtai TMV three axis CNC machine, and a LSTM-Autoencoder that was trained using transfer learning and the incremental ensemble learning algorithm. The LSTM-Autoencoder trained using transfer learning had the base Autoencoder trained using the data from the Tongtai[215]. The system was then transferred

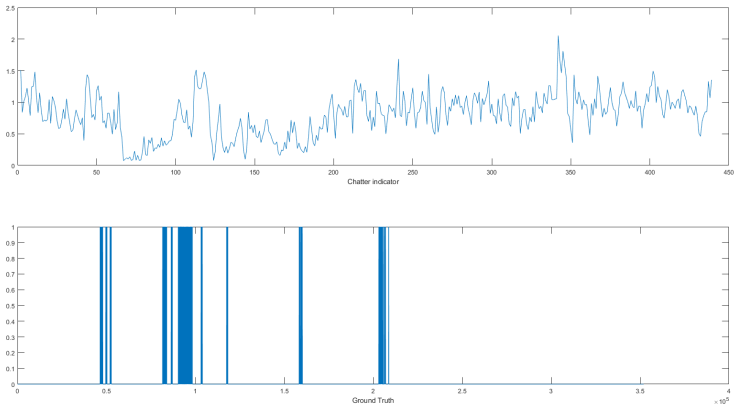


Figure 5.4: Ground truth from test signal used for testing incremental learning algorithm

to the Hurco VMX42Di with the decoding layers being retrained on 30 000 data points, resulting in 22 000 parameters being retrained.

The output of the LSTM-Autoencoder is the signal reconstruction loss. If this reconstruction loss is greater than a pre-determined threshold, then the signal is determined to be an anomaly. This threshold is based on the three sigma rule from the reconstruction losses obtained during training. The reconstruction losses from the directly trained network, and the directly trained network with incremental learning can be seen in Figure 5.5, with the incremental ensemble network seen on the top figure and the directly trained network in the bottom figure.

Both subfigures show the reconstruction signals crossing the threshold limits at timings that align with the chatter portions of the cutting operation, seen in the acceleration signal and the ground truth indicators. There are some minor differences observed between the two networks, but the same general trend can be seen in both. The directly trained system has an accuracy of 95.01%, while the incremental ensemble learning system has an accuracy of 95.62%, indicating little difference between the two system.

The reconstruction losses from the transfer learning trained network and the transfer learning network with incremental ensemble learning can be seen in Figure 5.6. The output of the transfer learning system can be seen in the bottom subplot with the transfer learning network with incremental ensemble learning in the top subplot.

As the figure shows, the transfer learning system has greater difficulty identifying the chatter condition and has significant reconstruction error near the end of the cutting op-

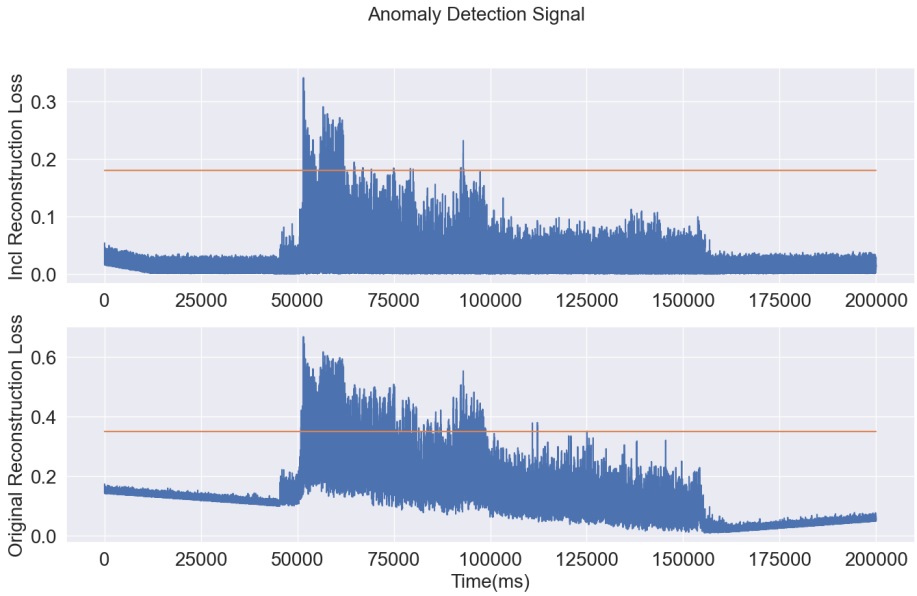


Figure 5.5: Reconstruction Loss from directly trained network and ensemble-inc network

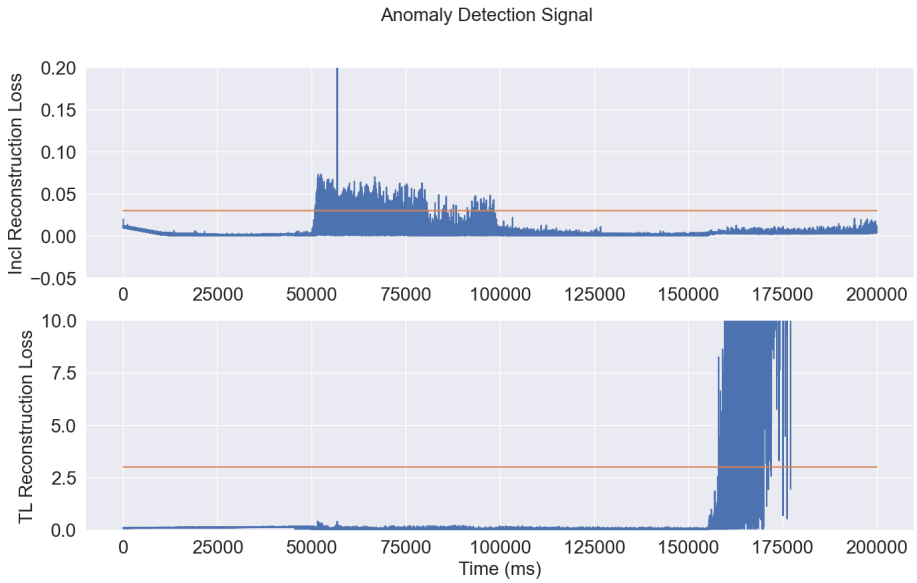


Figure 5.6: Reconstruction Loss from transfer learning network and ensemble-inc network

eration when nothing is occurring. This is in contrast to the transfer learning system with incremental learning, which follows the cutting signals more accurately. Although there are some differences in the reconstruction error, the overall trend matches that seen in Figure 5.5, and matches the ground truth well. This can also be seen when the accuracy is compared. The transfer learning system only has an accuracy of 75.23% but the transfer learning system with incremental learning system has an accuracy of 93.57%. The accuracy results of the four networks is summarized in Table 5.1.

Table 5.1: Accuracy Comparison of Different Machine Learning Architectures

Machine Learning Architecture	Accuracy
Directly Trained Network	95.01%
Directly Trained Network with Incremental Ensemble	95.62%
Transfer Learning Network	75.23%
Transfer Learning Network with Incremental Ensemble	93.57%

In the experiments conducted, the directly trained system had a high accuracy rate of 95%, which is comparable to other experiments seen in related work. We were able to show that the incremental ensemble learning method was able to marginally improve this accuracy, but most importantly, did not make the system worse. Given the already high accuracy of the system, it is challenging to improve the accuracy of said system without overfitting it, thereby reducing its ability to generalize.

Where we did observe considerable increase in accuracy, and thus improvement, is with the transfer learning system. Systems trained with transfer learning often have lower accuracy than those directly trained on a data set, with values ranging between 70-85% [204][96]. This is noticeably lower compared to systems directly trained on a data set; but transfer learning systems require significantly less data to train compared to systems trained from scratch. Requiring less data is of significant interest for those hoping to implement these algorithms in production settings, thus, the incremental ensemble learning algorithm provides the user the "best of both worlds". By using this approach, the user can implement a network quickly using transfer learning and then have it continually improve and reach accuracy levels comparable to networks directly trained on millions of data points.

Although the incremental ensemble learning method is effective, one of its major trade

offs is the amount of memory and computational resources it requires. Training an LSTM-AutoEncoder directly on the data requires training approximately 45 000 parameters on over a million data points. On a desktop computer with 16 GB of memory, this process requires several days of intensive training and computation. However, this training is done offline, and does not impact any of the machining processes. The final network that is deployed requires only several megabytes of memory, and virtually no computational effort. This is a similar result for the system trained with transfer learning, that only requires training 22 000 parameters on 100 000 data points. In comparison, the incremental learning method has 275 000 total parameters, with nearly 50 000 parameters being trained each time a weak learner is added. Although this work can be done in the background, it is done on the CNC machine itself. Depending on the computational load of the other processes on the machine, this may or may not be possible. This problem is further complicated when we compare the cost of an extra computer versus the cost of increased computational load on the CNC machine.

Nevertheless, one benefit of the incremental learning method is the ability to learn and add classifiers as time progresses. In our experiments we focused on increasing accuracy of the system and the ability to accurately detect chatter conditions. However, one of the strengths of incremental learning is the power it gives a network to learn how to deal with new classes. In this case, new anomalies can be learned over time as new data is made available, provided there are sufficient indicators to distinguish these classes. As new anomalies are introduced to the network, the incremental learning method can be used to identify them, using the same framework as the anomaly detection with little modification required. This potential is discussed in Chapter 7 but is ultimately left as a future work

5.4 Discussion and summary of contributions

In this section we propose an incremental learning approach for anomaly detection in CNC milling machines. This approach has several potential contributions to the literature. These contributions are summarized as follows

- Developed and validated incremental learning algorithm for LSTM Autoencoders
- Demonstrated that proposed method produce comparable results for networks trained from scratch to those trained with incremental learning
- Demonstrated that proposed incremental learning method can significantly improve accuracy of LSTM Autoencoders trained using transfer learning approaches

This proposed method has the potential to reduce the amount of data required overall, and more importantly, spreads the burden of data collection over a longer period of time. By applying transfer learning we are able to reduce the amount of data required to train an initial system. As seen in Chapter 4, we are able to train a base anomaly detection system using a data set that is 15-25% the size of one needed to train from scratch. In our experiments conducted in this chapter, we trained each of the week learners with small data sets that were only 2.2 % the size of the original data set. This combined approach allows us to obtain comparable results to the network that is trained from scratch, but is significantly smaller.

Although this approach gives us considerable accuracy, it does require more memory to be used, and results in more computing resources to be used. This approach may be a detriment to some smaller CNC milling machines that are limited on computing power. This issue was not experienced during our experimental testing, and would not likely be an issue with the machines that we used, as the Hurco CNC machines have substantial computing power, but this could theoretically be a limitation of this approach. Nevertheless, these methods show great potential for future applications and for reducing much of the complexity for integrating machine learning methods in production.

Chapter 6

Multi-Axis Anomaly Detection with Transfer and Ensemble Learning

CNC machines are designed and built in a variety of configurations. The most common types of CNC milling machines are three axis milling machines. However, a fourth rotatory axis or five axis CNC milling machines are also commonly used in manufacturing. These extra axes introduce more functionality, but also more complexity. Unfortunately, the majority of work in the literature is only focused on three-axis CNC machining, leaving few systems that can be deployed to multi-axis CNC machines.

In this chapter, we explore how the methods described early can be extended to CNC milling applications beyond the three axis CNC machines described. We focus this work on five axis CNC milling machines, but believe many of the techniques could also be applied to four axis machines.

6.1 Motivation

Five axis CNC milling machines are capable of achieving high machining performance when manufacturing parts with complex geometry, requiring high precision. Although these machines have considerable capabilities, they also cost significantly more than their three axis counterparts, with typical five axis milling machines being three times more expensive. As a result of this increased cost, there are typically fewer five axis machines implemented in production, and less available data as a consequence. This makes training machine learning models challenging as an operator may not have an abundance of old data

to draw from for training a network, ultimately limiting the amount of readily available resources.

Of the many different types of configurations of five axis CNC milling machines, one particular configuration provides a unique opportunity to leverage transfer learning to its advantage. For swivel head and rotary table style CNC machines, such as the Hurco VMX42SRTi, there is significant overlap in the geometry between a three-axis mill and the five-axis mill. For example, the Hurco VMX42SRTi and the Hurco VMX42Di are constructed with the same column casting, base case and y-saddle. This resemblance means that the two machines will respond to an input signal with greater similarity [63]. An example of this swivel and tilt type of machine can be seen in Figure 6.1.

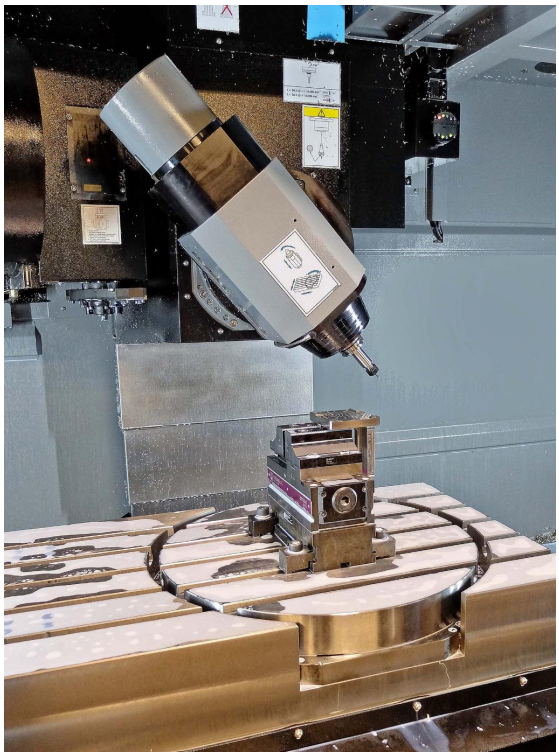


Figure 6.1: Hurco VMXSRTi showing the B-Axis Spindle and C-Axis Rotary Table[138]

We propose a method that leverages the similarity between these machines to create an anomaly detection method using data from the three axis VMX42Di and transferring it to the VMX42SRTi. This algorithm is described in greater detail in the following sections.

6.2 Multi-axis anomaly detection algorithm using transfer and ensemble learning

To leverage the similarity between the Hurco VMX42Di and the VMX42SRti, we propose a transfer and incremental ensemble learning approach. In this proposed approach, we use the encoder from the three-axis network to capture the known dynamics from the spindle. We then implement an ensemble learner to capture the new degrees of freedom from the spindle and the table. The transferred portion and the ensemble learner is then fed into a decoder that is trained on data from the five axis machine. This topology can be seen in Figure 6.2.

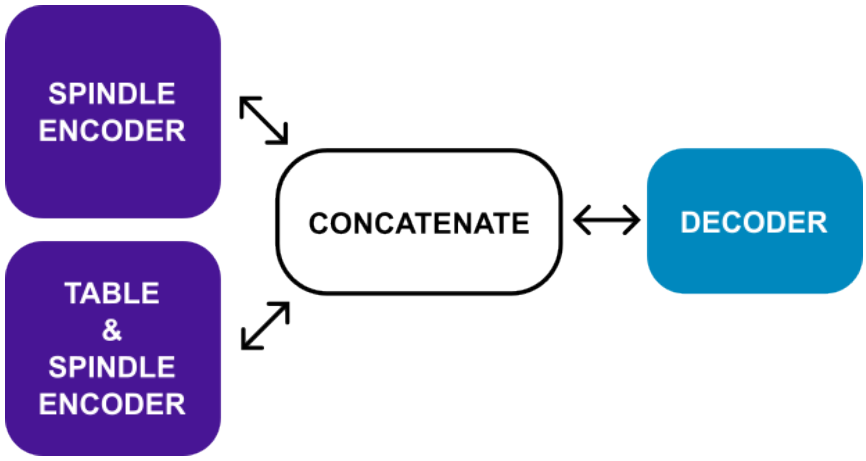


Figure 6.2: Block diagram of five axis ensemble learning approach

As described in the preliminaries in Section 2.3, ensemble learning is a machine learning technique that combines several smaller models to generate a final prediction based on the input of these models. These models work well by promoting a diversity among the models, allowing the combined model a greater span [172]. We use this knowledge to implement what is known as a stacking ensemble learner, also known as stacked generalization. In stacked ensemble learning, we train a series of models on training data and then train a combiner model using new training data. This stacked model produces a final prediction based on acquired knowledge from all of the models [169]. An example of this data flow can be seen in Figure 6.3 that shows the flow of data for a typical stacking process.

Given that one head of the network is taken from the three axis anomaly detection network, its configuration is already known. This head, labeled the spindle side, is trained

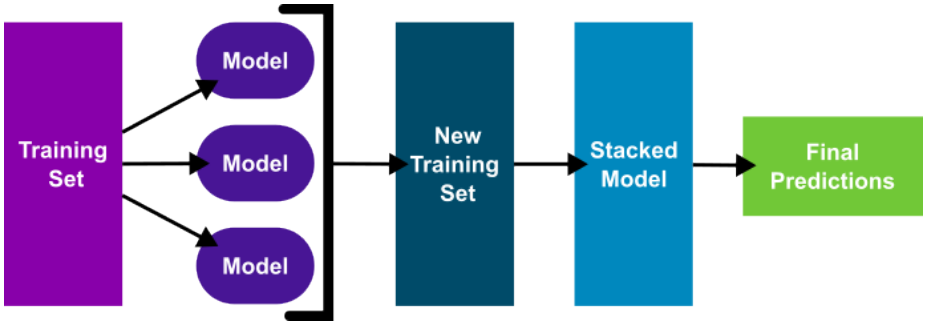


Figure 6.3: Typical stacking ensemble learning process flow

using the data and training parameters described in Chapter 3. We then create another encoder head to capture the table and tilt dynamics, which we have named the table side. This head is designed with the same architecture as the spindle side, but trained on a different data set. Both encoders are then fed into a single decoder with a modified architecture to capture the dynamics from both heads, this is done by having an LSTM layer act as a second bottle neck layer, further capturing the dynamics of the interaction between the two heads. Following this bottleneck layer, the decoder reverses the dynamics of the encoders to reconstruct the original signal. This architecture is seen in Figure A.1 in Appendix A.

This algorithm can be described algebraically as follows. Given input $x_{spindle}$ and x_{table} that represent sensor inputs from the spindle and table of the five axis CNC machine respectively, we can define one encoder head corresponding to the spindle as $g_{spindle}(x)$ and one encoder corresponding to the table as $g_{table}(x)$. We then combine these two encoders with an LSTM layer, where the output from the previous layer is defined as the concatenation of the outputs from the two encoders

$$h_{LSTMEncoder}(\cdot) = [h_{spindle}, h_{table}]^T. \tag{6.1}$$

This combined layer output then feeds into a combined LSTM decoder to reconstruct a combined input signal of the form

$$\hat{x} = [x_{spindle}, x_{table}]^T. \tag{6.2}$$

The overall Autoencoder is thus defined as

$$f_{StackedDecoder}(g_{spindle}(x_{spindle}), g_{table}(x_{table})). \tag{6.3}$$

This approach contrasts a standard LSTM Autoencoder that would be trained from scratch, which is simply defined as

$$f_{decoder}(g_{encoder}(x_{spindle,x_{table}})). \tag{6.4}$$

This standard LSTM Autoencoder requires a large set of training data on the five axis machine to properly converge, whereas the stacked encoder in Equation (6.3) can make use of some trained parameters from the spindle encoder to reduce the overall data required.

6.2.1 Incremental Ensemble Learning for multi-axis anomaly detection

To improve the accuracy and robustness of this network, we propose implementing an incremental learning strategy using the methods described in Chapter 5. This approach allows the network to compensate for limited initial training data, and improve its performance over time, as more five axis machining data is made available.

This proposed method begins by training a stacked ensemble learner described in Section 6.2 to act as the strong learner. This method transfers the encoder spindle from a compatible three axis CNC anomaly detection system, and then trains an encoder for the table and a stacked decoder. This training is completed on an initial dataset generated from the five axis CNC machine.

This initial network can be deployed on the machine, then as more data is made available, weak learners are added to the system. These weak learners copy the same structure as the spindle encoder head, the table encoder head, and the stacked decoder. This means that each weak learner separately has two encoder heads and one decoder head. The output of all of these weak learners and the strong learner are combined using a single dense layer.

The weak learners are added periodically as more data is made available. For testing purposes, each set of cutting data was immediately added, but a buffer of data could also be added in practice. The weak learners were trained for two epochs, in batch sizes of 10 using mini-batch gradient descent. Through experimental testing, it was found a weak learner pool of five learners provided the optimal result. The change in learning loss and number of weak learners can be seen in Table 6.1.

A simplified block diagram outlining the construction of this network can be seen in Figure 6.4, and the full network details seen in Figure A.2 in Appendix A.

Table 6.1: Number of weak learners vs learning loss for multi-axis anomaly detection

Number of Weak Learners	0	1	2	3	4	5
Learning Loss (10^{-2})	17.92	12.54	10.05	5.61	5.30	4.67

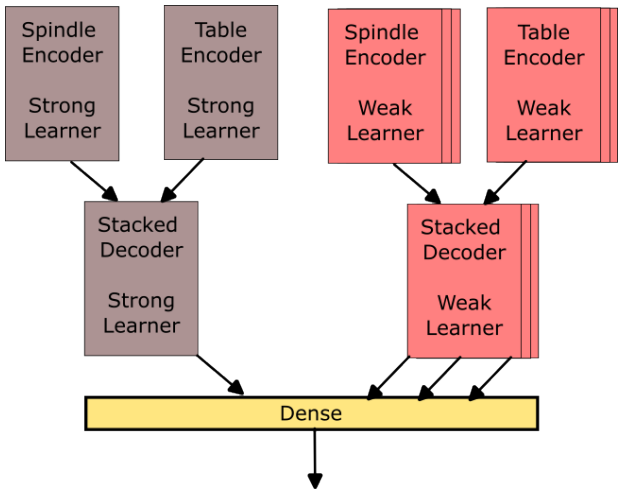


Figure 6.4: Block diagram for five-axis ensemble incremental learning architecture

6.3 Experimental implementation

To experimentally develop and validate the multi-axis anomaly detection algorithm, a Hurco VMX42SRTi five-axis CNC milling machine was instrumented with accelerometers to capture the spindle and table dynamics. To capture this data, a set of Erbessd EPH-V11E wireless tri-axial vibration sensors were modified to capture the necessary cutting information at an acceptable sampling rate. The information was sent via the Erbessd Phantom gateway to a desktop computer, where it was recorded for post processing. The Erbessd sensors were also supplemented with additional single axis accelerometers, produced by PCB Piezotronics. This experimental setup can be seen in Figure 6.5.

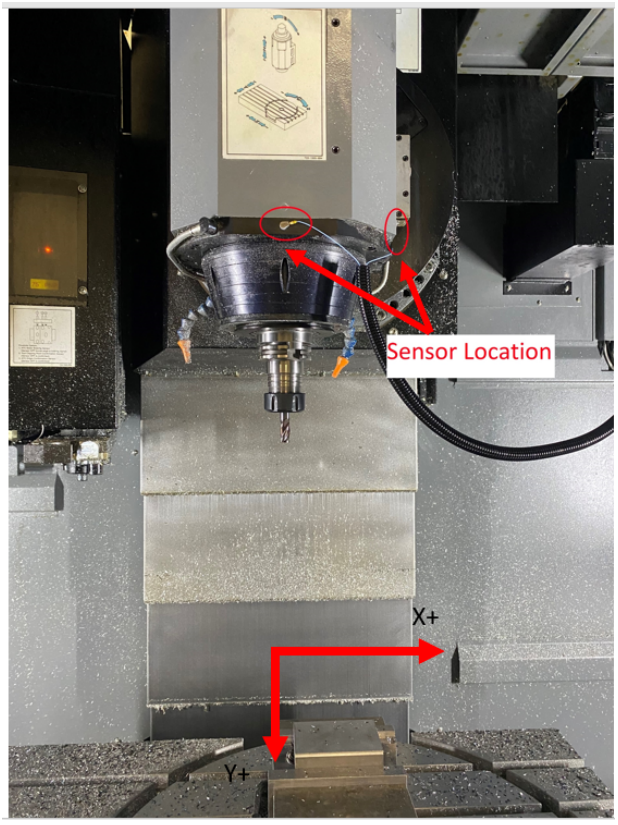


Figure 6.5: Experimental Setup on Hurco VMX42SRTi

Cutting experiments were completed by performing straight cuts in the X and Y direction, and then completing a "ramp" cut in the XYZ direction. This 3D model of this ramp can be seen in Figure 6.6, with the resulting cut part in Figure 6.7.

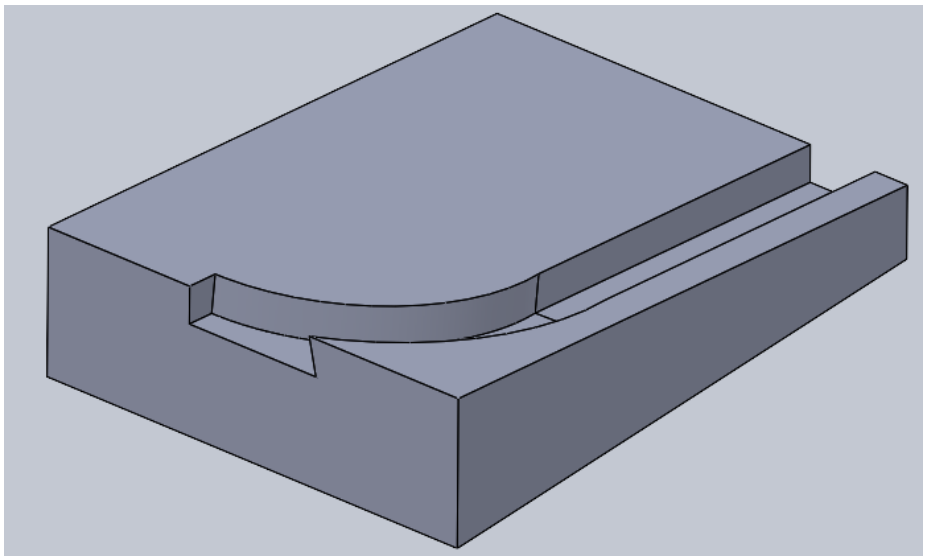


Figure 6.6: Solidworks model of ramp cut completed in XYZ

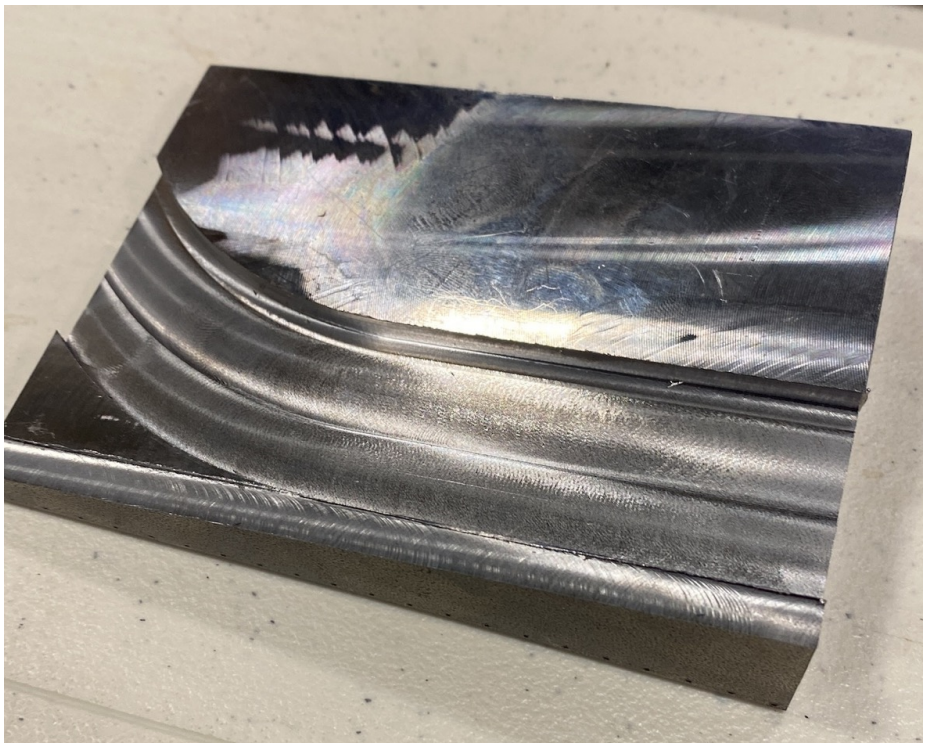


Figure 6.7: Experimental results of ramp cut completed in XYZ

Upon inspection of Figure 6.7, we see that there are surface defects and chatter marks near the end of the cut seen in the bottom right corner of the figure. This result is corroborated when the frequency response of the system is evaluated, seen in Figure 6.8. We note the change in magnitude for the cutting frequencies compared to the stable cuts at the times coinciding with the onset of chatter during the ramp cut. Using these thresholds, we were able to develop a ground truth indicator for when chatter occurred.

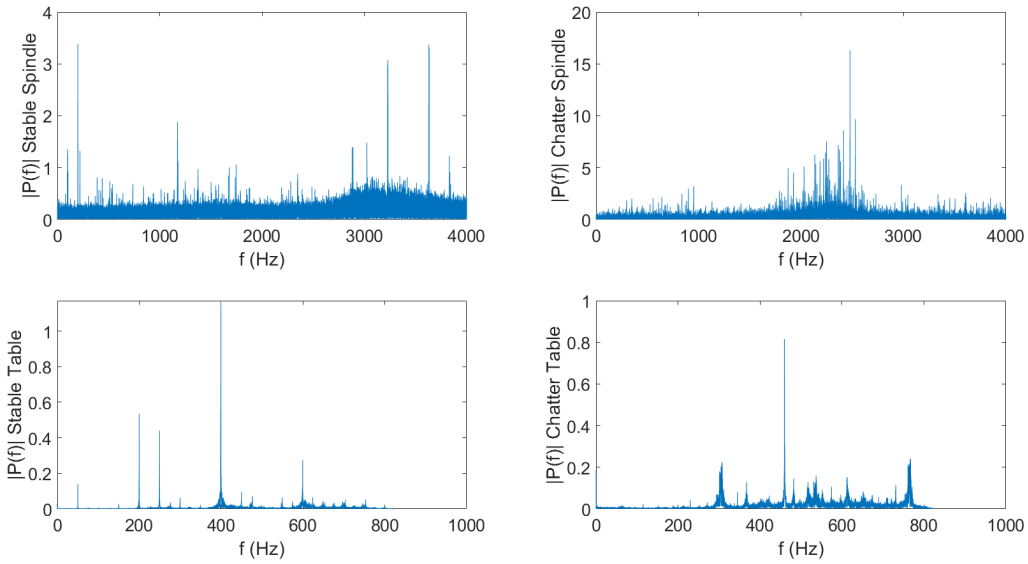


Figure 6.8: Frequency response from ramp cut completed on five axis CNC

Once the cutting experiments were completed, an anomaly detection system was trained from scratch using the available data to develop an Autoencoder based on the structure outlined in Chapter 3. This anomaly detection system was trained using all of the data available for stable cutting conditions, and then was provided cutting signals from both stable and chatter conditions for testing data. This system has approximately 70 000 trainable parameters and was trained on approximately 150 000 data points.

Using the available testing data, the network trained from scratch was tested with the signal seen in Figure 6.9. The result of this network can be seen in Figure 6.10, that has the reconstruction loss in the top plot, the decision from the chatter indicator in the middle plot and the ground truth on the bottom plot. As the figure shows, the system is able to capture many of the chatter dynamics, but has difficulty making a clear distinction at the 60 second mark. This results in an accuracy of 76.62% and a recall of 78.75%, which acts

as a baseline accuracy for the system.

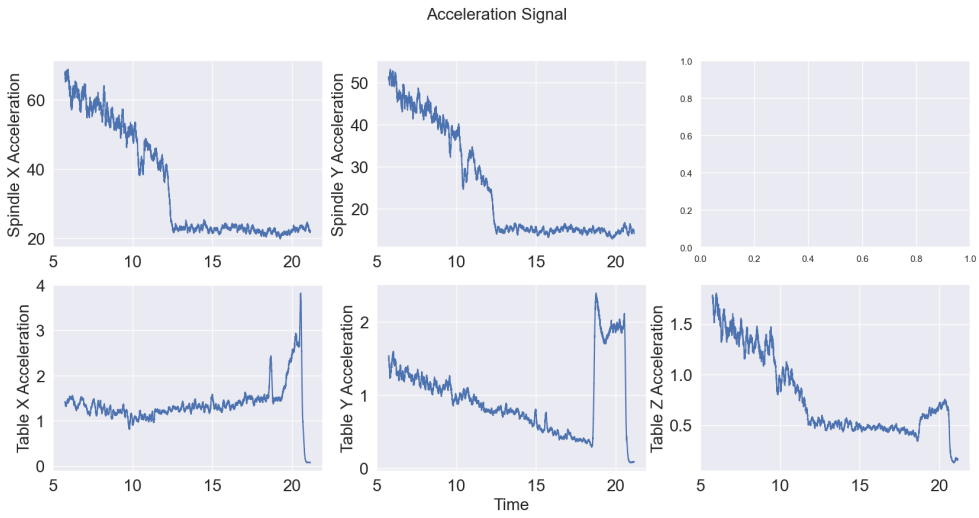


Figure 6.9: Example Accelerometer Signal for Five Axis Machine

Following the approach described in the Section 6.2, an initial network was created using the transfer and ensemble learning approach, without incremental learning. The three axis portion of this network was transferred from a Hurco VMX42Di system that was trained and implemented in Chapter 3 and Chapter 4. This system has approximately 23 000 trainable parameters, and was trained on approximately 25 000 data points. The results of this initial network are quite poor and can be seen in Figure 6.11. This network only has an accuracy of 16.34 % and a recall of 13.28 %, indicating significant misclassifications.

The result from the transfer and ensemble learning approach is in stark contrast to the transfer and incremental ensemble learning approach. In this system, we begin with the network designed with only transfer and ensemble learning. Then as additional data is provided to the system, the incremental learning methodology described in Section 6.2.1 can be implemented. These weak ensemble learners allow the network to gain a more complete understanding of the impact the additional aspects of the five axis mill, namely the swivel and tilt, have on the system. This final network is quite large, with nearly 415 000 total parameters, but only introduces 70 000 parameters during the training phase of each weak learner. Furthermore, each weak learner is only trained on 4500 data points. This means that the entire data set is only trained on 50 000 data points, which contrasts the 150 000 data points required to train the network from scratch. This result is encouraging, since simpler CNN Autoencoder networks for five axis machines require approximately



Figure 6.10: Anomaly Detection signal from directly trained network compared to Ground Truth

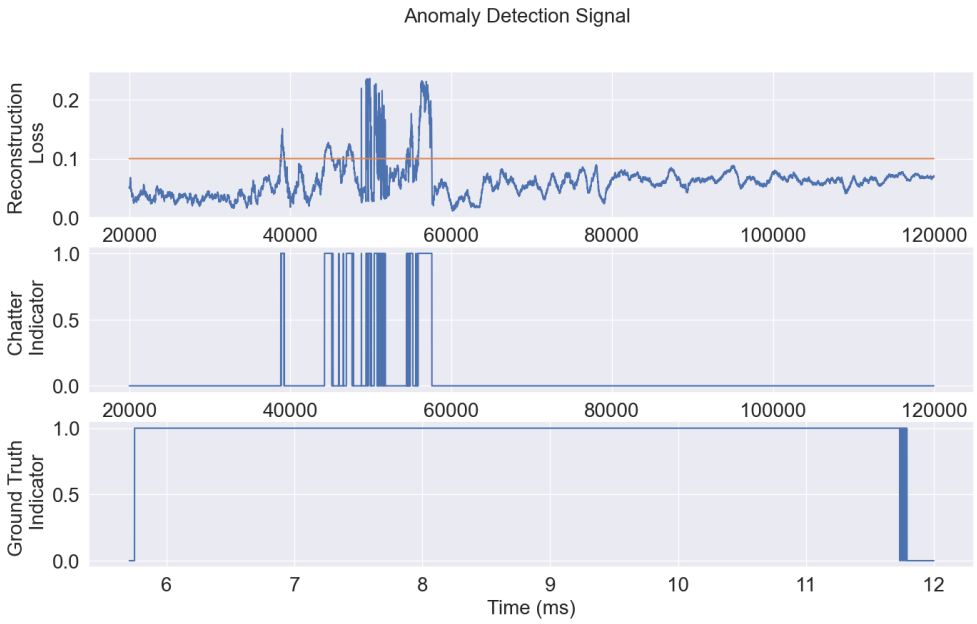


Figure 6.11: Anomaly Detection signal from Transfer Learning compared to Ground Truth

80 000 data points to train [156], and do not have the same capability as LSTM Autoencoder networks. The results of our network trained using transfer and ensemble learning can be seen in Figure 6.12. These experiments show that the network trained with transfer and incremental ensemble learning have an accuracy of 72.87% and a recall of 74.85%, which is comparable to the network trained from scratch. The comparison of these metrics for the various networks can be seen in summary in Table 6.2.



Figure 6.12: Anomaly Detection signal from Transfer Learning and Ensemble Incremental Learning compared to Ground Truth

Table 6.2: Comparison of network performance metrics

Network Configuration	**Accuracy Score (%)**	**Recall Score (%)**
Trained from Scratch	76.62	78.75
Transfer and Ensemble Learning Only	16.34	13.28
Transfer and Ensemble Learning with Incremental Learning	72.87	74.85

These results indicate that we can leverage the data and encoder from the three-axis anomaly detection system to a five-axis anomaly detection system, if we use an incremental

ensemble learning approach. We are able to obtain similar accuracy compared to a traditional Autoencoder that is trained from scratch, but able to leverage the systems already developed.

6.4 Discussion and summary of contributions

In this chapter we proposed a method for how the anomaly detection system designed for a three-axis CNC milling machine can be extended to a five-axis milling machine. This is accomplished by freezing the encoder from the three-axis milling machine and then implementing an extension of the incremental ensemble learning method described in Chapter 5. The results from this approach show that this method is able to produce results comparable to a system trained from scratch on a much larger data set from a five-axis milling machine.

This proposed approach has considerable potential benefit for the swivel-tilt style of 5-axis CNC milling machine, but could potentially be extended to and 3+2 style milling machine. Through our approach we captured the dynamics of a vertical spindle in one encoder and the table in another encoder. This dynamic is common for many 3+2 milling machines, where one part of the machine is responsible for movement in three axes and one part is responsible for another two. Depending on the configuration of the portion of the machine responsible for the movement in three axes, there may be a greater MMD score, but provided it is not excessively large, transfer learning may still be possible.

While this approach is likely to be effective for many 3+2 machines, it is likely not going to be effective for 4+1 style milling machines, or true five axis milling machines. These style of milling machines have four or more axes moving simultaneously, which would likely introduce more complex dynamics, such as rotation, that have more coupled dynamics that cannot be as easily separated. This aspect contrasts the 3+2 style machines that typically make moves in one plane or the other, leaving the other axes fixed. Nevertheless, this does represent a considerable improvement, as there are currently few options for any five axis style milling machines, and the 4+1 style machines can be explored in future work.

Chapter 7

Multi-Class Anomaly Prediction

In this thesis we have presented several methods for anomaly detection in CNC machining. Although we have focused on chatter, these methods are not specific to any one anomaly, and can be applied to many types of anomalies. Although detecting the anomalies is crucial, being able to classify the specific type of anomaly is advantageous for troubleshooting and maintenance operations.

In this chapter we discuss a proposed multi-class anomaly prediction method that can be applied for three axis CNC milling machines. This proposed method is able to not only detect an anomaly, but present a probable class label. This proposed approach builds on the networks described earlier in this thesis and applies many of the techniques previously discussed.

7.1 Motivation

As outlined in Chapter 1, there are many potential anomalies that may occur during the CNC milling process. Through the work shown in Section 4.3.2, it was demonstrated that the proposed anomaly detection approach is capable of detecting both chatter and the presence of a broken tool bit. An example of this can be seen in Figure 7.1 that indicates the portions a cutting signal that are stable, chatter and where a broken tool is engaged. This signal was captured on the Roland MDX-40A using the wireless Erbessd tri-axial accelerometers.

However, a limitation of this approach is the inability to distinguish one type of anomaly from another. Although the presence of any anomaly is undesirable, being able to make a

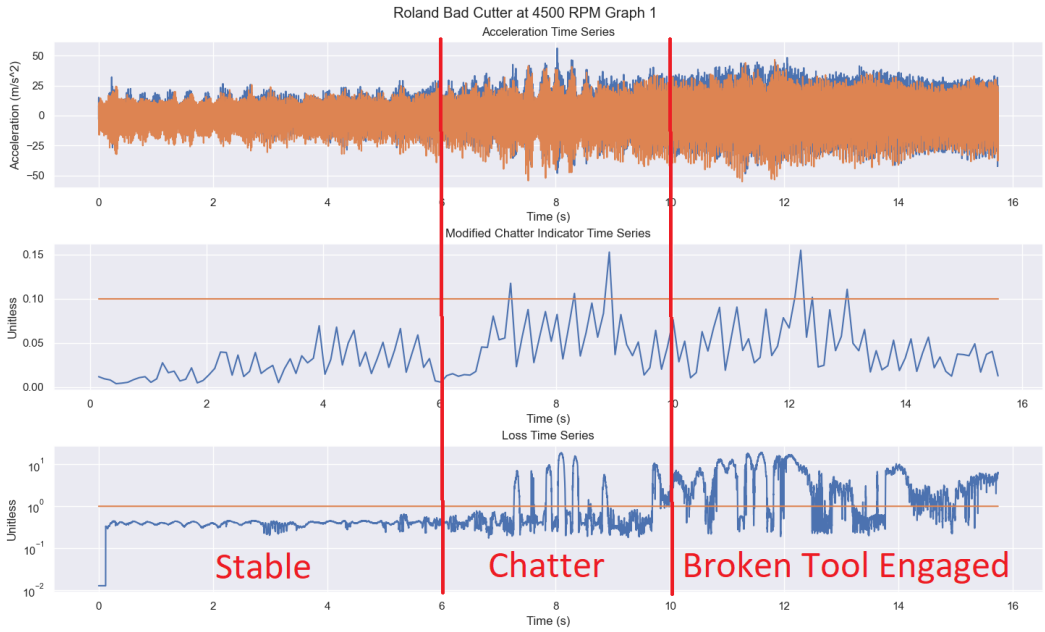


Figure 7.1: Example Cutting Signal Containing Stable, Broken and Chatter Cutting Conditions

prediction as to what the anomaly is provides the user with a useful troubleshooting tool. Furthermore, if this predictive ability can be further refined, estimates can be made as to how likely a tool is to break or how likely it is that chatter may occur.

7.2 Multi-class anomaly prediction algorithm

To be able to predict a label for multiple classes of anomalies during CNC milling, we propose an extension of the work described by Kvinevskiy [94]. Kvinevskiy proposed using a CNN Autoencoder for dimensionality reduction of microphone data and then apply a SVM to separate the data between stable and chatter cutting conditions. We propose to use the LSTM Autoencoder we designed previously to apply a similar dimensionality reduction and then apply a Linear SVC to develop decision boundaries for the stable, broken and chatter cutting data. We believe that this approach is effective since we have been able to detect both broken tool cutting conditions and chatter cutting conditions with this Autoencoder. The overall algorithm can be seen in Algorithm 2.

Algorithm 2 Multi-Class anomaly prediction algorithm

1: Let x represent an input signal to an Autoencoder anomaly detection system
2: Let $f(g(x))$ represent an Autoencoder with $g(x) = h$ begin the encoder portion of the network, $f(\cdot)$ being the entire decoder portion of the network, and h being a latent representation of the features
3: Train $f(g(x))$ aiming to minimize the cost function $x - \hat{x}$, where $\hat{x}$ is the reconstructed input signal
4: Modify the Autoencoder to $f_l(g(x))$ where $f_l(\cdot)$ is the desired layer of the decoder
5: Pass in training data to $f_l(g(x))$ to create dataset X_{multi}
6: Train a One-versus-all Linear SVC of the form $S(x) = w^T(x) + b$ where w is a matrix of weights and b is a bias vector
7: Convert the output from the Linear SVC to percentages using Platt scaling by optimizing parameters A and B in $P(\hat{y}|x) = \frac{1}{(1+exp(A*S(x)+B)}$ where $\hat{y}$ is the predicted class label
8: Monitor the output probabilities for changes above a user defined rate

7.2.1 Dimensionality Reduction using a LSTM-Autoencoder

Dimensionality reduction is often used to convert a network with a large feature set to a lower dimensional output for visualization or interpretation. Of the many techniques available, the Autoencoder is one of the most effective dimensionality reduction methods because the bottleneck layer naturally acts as a reduced-dimensionality vector [15]. This means that we can design and develop the Autoencoder using the already established methods, and then modify its output to easily produce a network for dimensionality reduction.

To accomplish this task, we begin by modifying the Autoencoder architecture from our anomaly detection system to have a layer output at dimension of interest. Once the network layer has been modified, the entire network can be trained using data from our stable cutting conditions. After the entire network has been trained, the network can then be truncated at our desired layer, producing the dimensionality reduction system. The overall network design can be seen in Figure A.3 in Appendix A.

Through our experimentation it was found that the output from the bottleneck layer alone did not provide a high degree of separation between data from different data sets. To compensate for this, instead of taking the output from the bottleneck layer, we take the output from the first layer of the decoder as the output for our dimensionality reduction system. For this particular implementation, we found that the data separation could be further improved by outputting the data in three dimensions and then projecting it onto

a two dimensional plane using the dot product.

The dimensionality reduction system was tested using the data from the cutting experiment seen in Figure 7.1. The data was manually separated in stable, broken and chatter signals and fed into the dimensionality reduction system. The output from this network can be seen in Figure 7.2. As the figure shows, there is a clear distinction between the chatter data set and the broken data set. However, there are some sections of the stable data set that do overlap with sections of the broken and chatter data sets. This is likely due to transition periods where a signal is technically still classified as stable, but begins to transition to chatter, or signals obtained during cutting with the undamaged flutes half of the damaged tool bit. Nevertheless, this result demonstrates that the dimensionality reduction system is capable of projecting the data sets in two dimensions with some degree of separation observed between the classes.

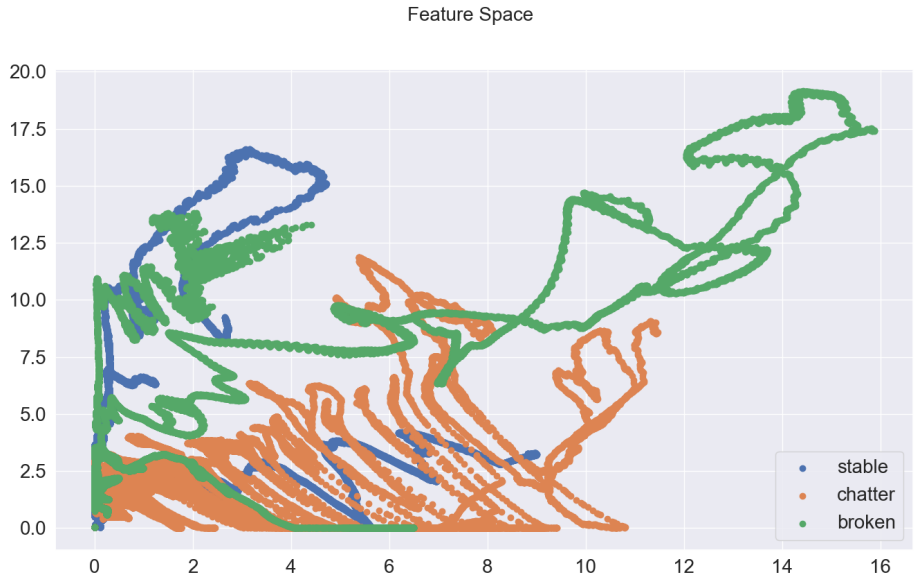


Figure 7.2: Feature Mapping of Cutting Conditions through dimensionality reduction via Autoencoder

7.2.2 Multi-class classification with Linear SVC

Using the dimensionality reduction system described in Subsection 7.2.1, we are able to take the accelerometer sensor data and create a two dimensional representation. With

this representation, we aim to build a multi-class label prediction system using simple classifiers.

Due to its simplicity and effectiveness, a Support Vector Machine (SVM) based approach was chosen. The SVM is a popular lightweight machine learning algorithm that is typically used for binary classification, but can be extended to multi-class classification by using a one-versus-all approach. In a one-versus-all (OvA) approach, the system heuristically iterates through the given classes and performs a binary classification between a given class and the remainder of the options. The classifier then assigns a label based on the result of these different combinations. A simple analogy for this approach would be a Venn diagram where we aim to look for the unique combination of descriptors that most correctly labels a data point.

An important design parameter during the construction of this SVM classifier is the type of kernel that is used. The classical implementation for a SVM is to find a linear boundary condition. However, data is often non-linear in nature and may not be easily separated with a linear boundary. To compensate for this problem, the data can be transformed using different kernels. Two popular kernel options are the radial basis function (RBF) kernel and the polynomial kernel. Both of these options were considered, with the output seen in Figure 7.3.

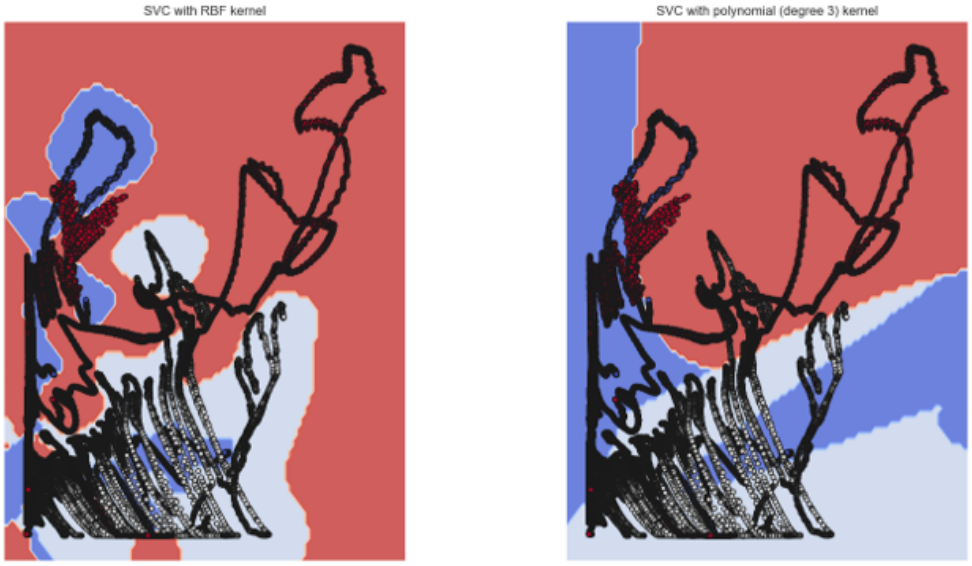


Figure 7.3: SVC with RBF Kernel and SVC with Polynomial

As the figure shows, neither kernel performs particularly well at separating the data.

Moreover, the training of these classifiers grows with the dataset, taking away the simplicity of this approach. After this experimentation, it was decided that the LinearSVC (linear kernel) was the preferred approach, as it was able to generate clear decision boundaries and could be computed quickly. The result of this classifier can be seen in Figure 7.4.

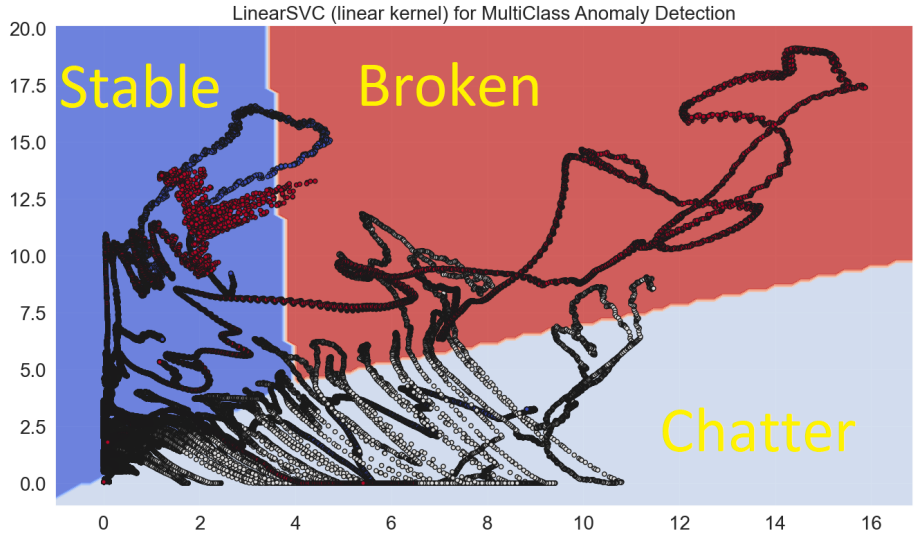


Figure 7.4: Linear SVC classifier with class boundaries for mapped features

7.2.3 Multi-class label prediction with Linear SVC

Using the methods described in previous chapters, we are able to generate a classification for a given data point. However, in multi-class problems, it is often more useful to provide a probability of a given class rather than a specific label. This probability provides specifics benefits for multi-class label prediction, by offering a numerical metric that can be monitored. For example, if we were to monitor the broken bit probability and noticed that the probability grew over a set time interval, we could make a prediction that the scenario is likely leading to the tool breaking. This method is a common approach applied with fuzzy inference, and could be easily integrated with the probability output from this multi-class label prediction.

To transform the multi-class label to a probability, we use the well known Platt scaling

equation, presented in the following

$$P(y = 1|x) = \frac{1}{1 + \exp(A * f(x) + B)}. \tag{7.1}$$

In Platt scaling, the output of the classifier $f(x)$ is used in a logistic transformation, where parameters A and B are found by estimating a maximum likelihood from the training data. This scaling produces a probability estimate of a given input for each of the classes the classifier is trained on, rather than a specific class label.

To integrate the Platt scaling, we modify the output of our multi-class SVM and train the A and B parameters from the training data. This allows us to generate a probability for all three classes from a given input. We test the output of this system using the input data seen in Figure 7.1. The resulting probability graph can be seen in Figure 7.5

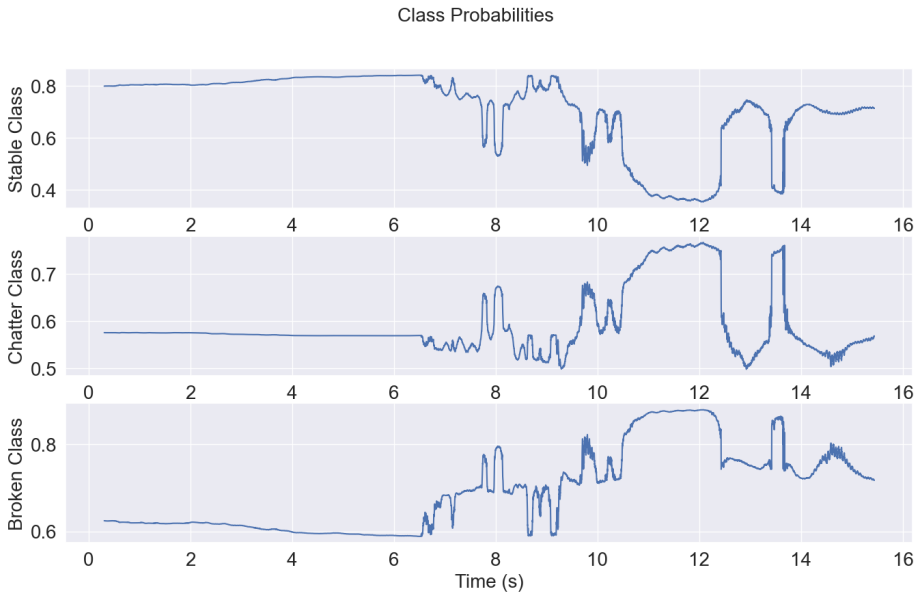


Figure 7.5: Predicted Class Probabilities based on input signal

From the figure, we see that the output of the graph qualitatively tracks the results observed. However, to properly evaluate the effectiveness of this class prediction method, we calculate the negative log likelihood to produce a score that can be used to quantitatively compare the performance of the algorithm. The negative log likelihood is an effective measure that has been used in other multi-label classification problems and has a practical meaning similar to the well known cross-entropy metric.

In the negative log likelihood metric, a score of zero indicates a perfect fit where the classifier is able to predict the correct class with no error, and a score of one is one where the classifier incorrect classifies every instance. This score is calculated with

$$-log(\hat{y}) \tag{7.2}$$

where $\hat{y}$ is the prediction corresponding to the true class label, after the model outputs have been converted to a probability through Platt scaling.

Using the negative log likelihood metric we evaluate the performance of the multi-class label prediction system and obtain a negative log likelihood score of 0.03. This score is extremely close to zero and indicates an effective system which is able to clearly predict the correct classes.

7.3 Multi-Class anomaly prediction with transfer learning

Using the proposed multi-class anomaly prediction system described in Section 7.2 we are able to generate an accurate prediction of a given anomaly. Given that the dimensionality reduction in this system is accomplished using an Autoencoder, we propose an extension that leverages transfer learning to implement the multi-class system while requiring less training data.

In this proposed approach, we train an Autoencoder using the transfer learning techniques described in Chapter 4, with the encoder trained using the source data set and the decoder trained using the target data set. We then follow the dimensionality reduction methods described in Section 7.2.1 and apply the proposed method described in Section 7.2.2 and Section 7.2.3 to generate a multi-class anomaly prediction method.

7.3.1 Experimental implementation of multi-class anomaly prediction

To experimentally implement and validate the proposed approach, an Autoencoder was trained using data from the Tongtai TMV-720A data set for the source network. Data was then collected from the Roland MDX-40A using the Erbessd wireless tri-axial accelerometers for the target dataset. A new Autoencoder was then trained using transfer learning

from these two datasets and then modified for dimensionality reduction. The accelerometer input shown in Figure 7.1 was then fed into this system with the output from the dimensionality reduction seen in Figure 7.6

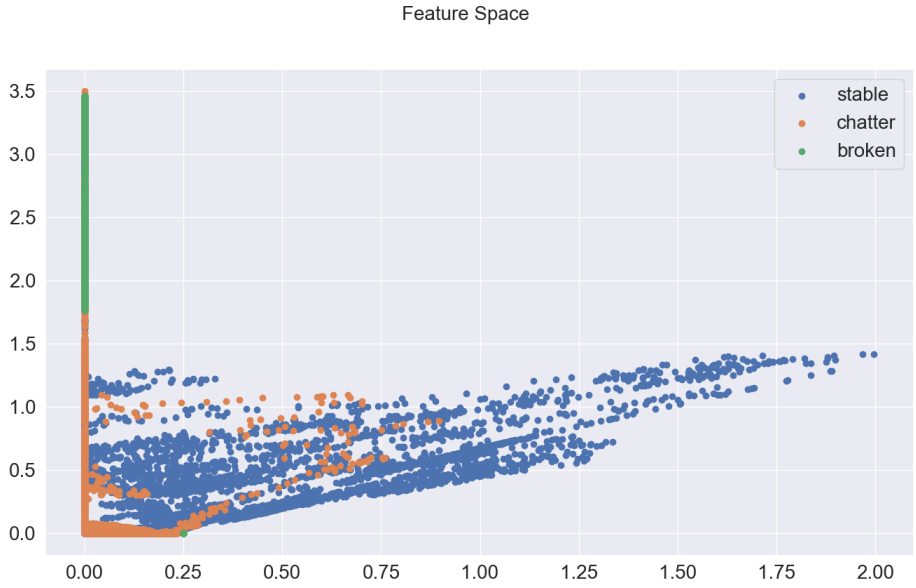


Figure 7.6: Feature Mapping of Cutting Conditions through dimensionality reduction via Autoencoder trained with transfer learning

The multi-class classifier was then trained using this transfer learning trained dimensionality reduction system and the data obtained on the target dataset only. The output from this classifier with the given test input can be seen in Figure 7.7.

As the figures show, given the same input, the transfer learning system does not have as qualitatively clear separation between the datasets as the system trained from scratch seen in Figure 7.2. Nevertheless, the Linear SVC is still able to establish clear decision boundaries that qualitatively align well with those seen in the system that is trained from scratch.

This system is then extended to generate probabilities using the Platt scaling using Equation (7.1). The resulting probabilities can be seen in Figure 7.8. As this figure shows, there is greater uncertainty in the class probabilities. This result is especially apparent between nine and eleven seconds where the potential class labels vary wildly.

To numerically compare the system trained with transfer learning we once again use the negative log likelihood. Using the methods described in Section 7.2.3, we obtain a negative

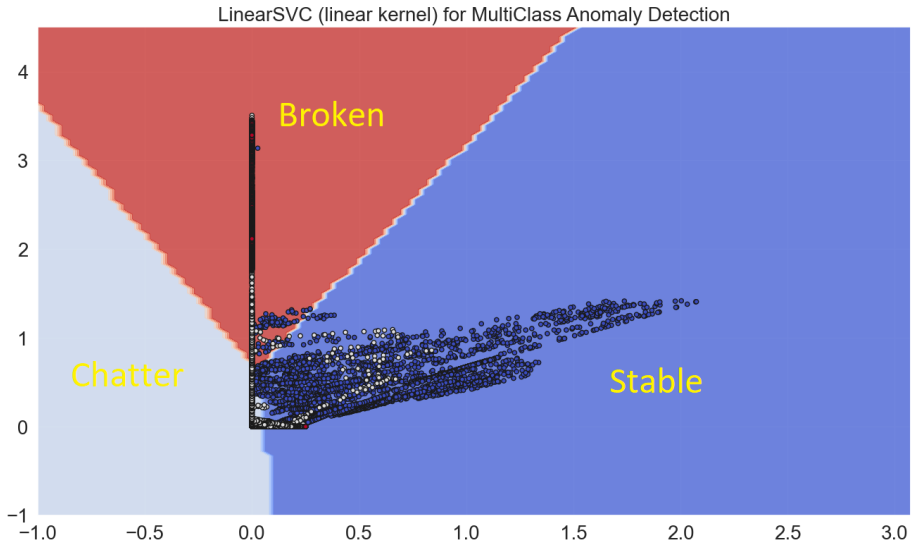


Figure 7.7: Linear SVC classifier with class boundaries for mapped features with Autoencoder trained with transfer learning

log likelihood score of 0.18. In practise, any score below 0.2 is considered to still be very good, but we can see that the system trained with transfer learning is still considerably worse than the system trained from scratch. Although the accuracy could potentially be improved by applying techniques such as incremental learning, this work still indicates that the proposed method is feasible and has merit.

7.4 Discussion and summary of contributions

In this chapter we explore a potential multi-class anomaly prediction method. This proposed method has several potential contributions to the literature. These contributions are summarized as follows:

- Demonstrated that multi-class anomaly detection is possible using an LSTM-Autoencoder for dimensionality reduction and using a LinearSVC for generating decision boundaries
- Demonstrated that multi-class label prediction is possible by modifying the aforementioned system using Platt scaling, which can be integrated into a fuzzy inference system

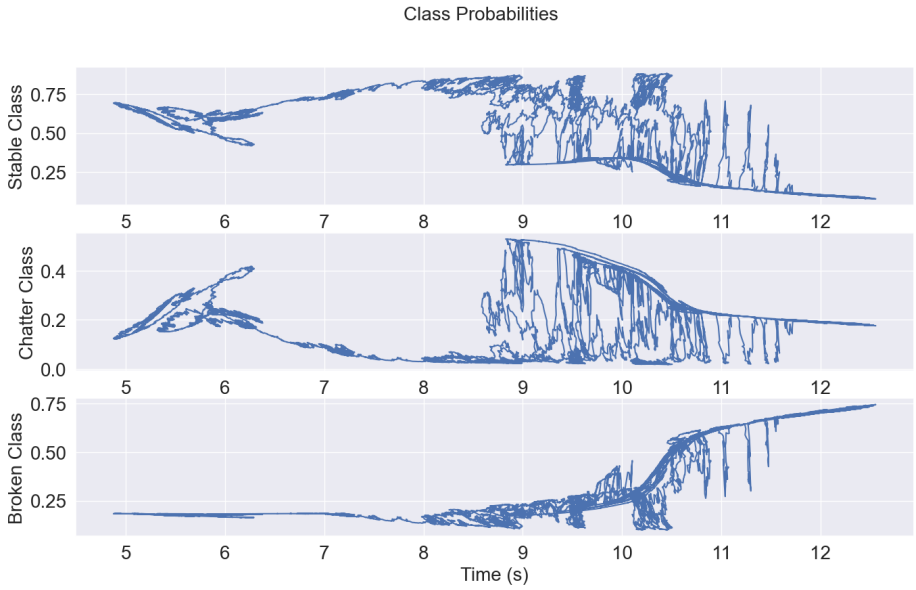


Figure 7.8: Linear SVC classifier with class boundaries for mapped features

- Demonstrated that transfer learning is a viable option for training the LSTM-Autoencoder using in dimensionality reduction

The work for this chapter demonstrates a unique distinction from many other systems seen in the literature, as we are able to not only detect, but identify multiple types of anomalies. This ability to identify multiple types of anomalies is completed with relatively inexpensive sensors and a flexible architecture that can be extended to many different systems.

In this thesis we explored the ability to predict two classes, namely broken tool bits and chatter, however, we can likely extend this approach beyond these two classes. Focusing on only accelerometers, we can also aim to detect tool flank wear, but we can likely detect even more classes if other sensors or inputs are included in the networks. For example, if we were to modify the network to include spindle speed or force measurement, we could likely predict the life of the spindle motor as well. These methods open the door to more general machine health monitoring, and truly demonstrate the potential of these proposed methods.

Chapter 8

Conclusions and Future Work

Motivated by the need to develop technologies to facilitate autonomous manufacturing, the work proposed in this thesis presents a potential framework for anomaly detection in CNC machining. This framework makes uses of deep networks based on the LSTM-Autoencoder and implement various machine learning techniques. These techniques demonstrate the potential of this framework and propose potential methods to overcome common shortcomings of machine learning techniques used in production.

In Chapter 3 we outlined a proposed methodology based on accelerometers and focused on using an LSTM-Autoencoder for anomaly detection in CNC milling machines. This methodology was experimentally tested for chatter detection and compared to established chatter detection methods.

This work was then extended by proposing a transfer learning based method for training the anomaly detection network in Chapter 4. This transfer learning approach allowed for similar performance to the network trained from scratch, but used a tenth of the required machining data and could be trained substantially faster. These methods were experimentally validated on a series of three-axis CNC milling machines and showed considerable promise. Furthermore, statistical measures were investigated in Chapter 4 Section 4.2.1 that investigated the relationships between the physical phenomena of CNC machining and common statistical measures. This work allowed for a deeper understand of where the impact of transfer learning could be greatest, and where there was limited benefit. In summary, it was found that transfer learning between machines with similar structural elements was most likely to be successful, with the source and target networks responding in a similar fashion. However, if the source and target networks were too similar, there was minimal benefit to applying transfer learning. This was most noticeable when net-

works designed on the same model of machine were deployed on one another, or if different materials were used.

The anomaly detection system was then extended by implementing an incremental learning approach in Chapter 5. This incremental learning approach was based on the popular Learn++ algorithm, which in turn, was based on the popular AdaBoost algorithm. This boosting approach proposed a strong learner trained over many epochs on a large data set, to be supplemented by weak learners that continued to learn as data was made available. Through the experiments conducted, it was found that the proposed incremental learning algorithm had little impact on networks trained from scratch, but could substantially improve the accuracy of networks trained with transfer learning. This combination of transfer and incremental learning allowed for the "best of both worlds" to be obtained, by quickly training a new system on a limited data set and then improving its accuracy over time.

This method of using transfer and incremental ensemble learning was then proposed in several extensions for multi-axis CNC milling machines, specifically in five-axis CNC machining in Chapter 6. In this work we propose a solution that transfers the knowledge of the spindle dynamics from a three-axis CNC milling machine to a five-axis CNC milling machine. We then supplement the spindle knowledge with an stacking based ensemble learner to capture the dynamics of the other machine axes, and then train it with incremental ensemble learning. Through experimental validation, we see that this approach is able to produce a network capable of achieving accuracy comparable to a network trained from scratch. This result is accomplished using significantly less target network training data when transfer learning is applied, but is capable of comparable results.

The proposed anomaly detection method is capable of detecting several anomalies at once. To further leverage this deep learning system, we propose a multi-class label prediction system in Chapter 7. In this approach, we use the same LSTM-Autoencoder structure used throughout this thesis. We then modify its output to create a dimensionality reduction system. This is accomplished by taking the output from one of the lower dimensional LSTM layers from the decoder portion of the network, and then projecting it onto a two dimensional plane. This lower projection is then separate using a Linear Support Vector Classifier (SVC). The output from this SVC is then converted to a probability using Platt scaling. This proposed approach allows us to track the changes in the system, and provide a prediction as it trends towards one class or another. This approach is experimentally validated and is able to predict the classes with a high accuracy. Transfer learning is then implemented by training the LSTM-Autoencoder using the methods described previously. This approach has the encoder portion of the network trained using transfer learning, and the decoder trained on the target machine. This transfer learning approach is tested exper-

imentally and still able to obtain good results, albeit lower than the system trained from scratch. This demonstrates the feasibility of this approach, but indicating that further refinement is required.

Through this work, we propose a framework based on the LSTM-Autoencoder that leverages transfer and incremental ensemble learning to detect anomalies in a variety of cutting scenarios using accelerometers, as a potential solution for CNC anomaly detection that is not machine or material specific, and does not require any expensive experimental setup. The experimental results show that this proposed framework has potential benefits to implement accurate algorithms while reducing the heavy burden of data collection on the end user, and can be done with commercially available off the shelf accelerometers.

Although deep learning methods, such as the LSTM Autoencoder, tend to be expensive and time consuming to train, the structure also allows for considerable flexibility. This means that the manufacturer can make a large initial investment to create a general network that can be shipped with each machine. The end user can then improve their own experience with incremental learning on the target machine. Furthermore, the manufacturer can transfer the general network to a variety of machines within their family of products. This flexibility demonstrates that the large initial investment does not need to be repeated for each machine, making it easier to scale these advanced algorithms to end users.

A potential strength of the methods proposed here is the modularity of the approaches. LSTM-Autoencoders have shown to be powerful tools, since deep learning networks can benefit from transfer learning. These systems capture the context of a machine well, and allow us to transfer that context and knowledge to other systems. We can then customize these networks with incremental learning to produce accurate systems that can adapt to different three axis and five axis milling conditions. This flexibility allows machine tool manufacturers to develop a central core that can be customized and deployed to a variety of machines within their product lines, all built upon a single unifying architecture.

8.1 Future Work

Through the work completed for this thesis a number of potential solutions have been presented. Although the results are promising, there is still future work that can be explored.

The Autoencoder structure that has been proposed is capable of capturing many features associated with CNC machining. However, the design of this network is not intuitive

and convergence criteria is unclear. For example, the layer size parameters for the Autoencoder must be sufficiently distinct for the bottleneck effect to be observed, but cannot be arbitrarily chosen, as the system may not converge. Future work could be completed to investigate the convergence criteria necessary for these algorithms. This would allow for different sensors and different features to be explored, if the convergence criteria was better understood.

To demonstrate the feasibility of this system in a closed-loop setting, a simple control strategy was used. Although this controller was able to achieve its goal of stabilizing the cut from a chatter condition to a stable condition, the approach is far from scientific. Further work can be done to better integrate anomaly detection and anomaly correction.

Looking beyond the architecture of the Autoencoder alone, we introduced a multi-class anomaly prediction algorithm in this thesis. Transfer learning was demonstrated to show potential, but the full extent of this system is unclear. It would be of potential interest to see how incremental learning could be use to incrementally learn new classes, and how this could be integrated into the existing architecture.

The work in this thesis has shown that the Autoencoder format has significant power for anomaly detection in CNC machines. Although the work was focused on LSTM-Autoencoders, we have demonstrated the deep networks have significant potential. Future work could be completed to investigate other large Autoencoders, such as transformer based Autoencoders, or different forms of traditional Autoencoder.

Lastly, in this thesis we have focused our work on CNC milling machines, but many of the sensing methods and natural phenomena can be extended beyond just milling. For example, milling, turning, grinding and drilling often experience many of the same issues and can be detected using the same sensors. This work could be extended to these applications, demonstrating the power of the approaches proposed here.

References

[1] Keivan Ahmadi. *Machining Chatter in Flank Milling and Investigation of Process Damping in Surface Generation*. PhD thesis, University of Waterloo, Waterloo, 2011.

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training. *Asian Conference on Computer Vision*, 2018.

[3] Yahya Mohammed Al-Naggar, Norlida Jamil, Mohd Firdaus Hassan, and Ahmad Razlan Yusoff. Condition monitoring based on IoT for predictive maintenance of CNC machines. In *Procedia CIRP*, volume 102, pages 314–318. Elsevier B.V., 2021.

[4] Y Altintas and E Budak. Analytical Prediction of Stability Lobes in Milling. *Annals of the CIRP*, 44(1):357–362, 1995.

[5] Yusuf Altintas. *Manufacturing Automation*. Cambridge University Press, 1 2012.

[6] Qinglong An, Zhengrui Tao, Xingwei Xu, Mohamed El Mansori, and Ming Chen. A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network. *Measurement: Journal of the International Measurement Confederation*, 154, 3 2020.

[7] Oscar Velásquez Arriaza, Zagaa Tumurkhuyagc, and Dong Won Kim. Chatter Identification using Multiple Sensors and Multi-Layer Neural Networks. In *Procedia Manufacturing*, volume 17, pages 150–157. Elsevier B.V., 2018.

[8] Muzaffer Ay, Max Schwenzer, DIrk Abell, and Thomas Bergs. Recurrent Online and Transfer Learning of a CNC-Machining Center with Support Vector Machines. In *IEEE International Symposium on Industrial Electronics*, volume 2021-June. Institute of Electrical and Electronics Engineers Inc., 6 2021.

[9] Davide Azzalini, Luca Bonali, and Francesco Amigoni. A Minimally Supervised Approach Based on Variational Autoencoders for Anomaly Detection in Autonomous Robots. *IEEE Robotics and Automation Letters*, 6(2):2985–2992, 4 2021.

[10] Long Bai, Xin Cheng, Qizhong Yang, and Jianfeng Xu. Predictive model of surface roughness in milling of 7075Al based on chatter stability analysis and back propagation neural network. *International Journal of Advanced Manufacturing Technology*, 126(3):1347–1361, 5 2023.

[11] Soeren Becker, Kevin Styp-Rekowski, Oliver Vincent, Leon Stoll, and Odej Kao. Federated Learning for Autoencoder-based Condition Monitoring in the Industrial Internet of Things. In *IEEE Conference on Big Data*, 11 2022.

[12] Stevo Bozinovski. Reminder of the First Paper on Transfer Learning in Neural Networks, 1976. *Informatica*, 44:291–302, 2020.

[13] Christian Brecher, Prateek Chavan, and Alexander Epple. Efficient determination of stability lobe diagrams by in-process varying of spindle speed and cutting depth. *Advances in Manufacturing*, 6(3):272–279, 9 2018.

[14] Christian Brecher, Ralph Klimaschka, Alexander Steinert, and Stephan Neus. Efficient determination of stability lobe diagrams deploying an automated, data-based online nc program adaption. *MM Science Journal*, 2021-October:4830–4835, 10 2021.

[15] A. Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.

[16] Daniel Carranco. Automation: The Future of Manufacturing Is Here - ProQuest, 2020.

[17] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-End Incremental Learning. In *European Conference on Computer VIsion*, 2018.

[18] Alejandro Cervantes, Christian Gagné, Gagn´ Gagné, Pedro Isasi, and Marc Parizeau. Evaluating and Characterizing Incremental Learning from Non-Stationary Data. *CoRR*, 2018.

[19] C. L.Philip Chen and Zhulin Liu. Broad Learning System: An Effective and Efficient Incremental Learning System Without the Need for Deep Architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):10–24, 1 2018.

[20] Joseph C. Chen and Wei Liang Chen. Tool breakage detection system using an accelerometer sensor. *Journal of Intelligent Manufacturing*, 10(2):187–197, 1999.

[21] Qipeng Chen, Qingsheng Xie, Qingni Yuan, Haisong Huang, and Yiting Li. Research on a real-time monitoring method for the wear state of a tool based on a convolutional bidirectional LSTM model. *Symmetry*, 11(10), 10 2019.

[22] Zeng Chen, Huan Xu, Peng Jiang, Shanen Yu, Guang Lin, Igor Bychkov, Alexey Hmelnov, Gennady Ruzhnikov, Ning Zhu, and Zhen Liu. A transfer Learning-Based LSTM strategy for imputing Large-Scale consecutive missing data and its application in a water quality prediction system. *Journal of Hydrology*, 602:126573, 11 2021.

[23] Bertrand Clarke and Bertrand@stat Ubc Ca. Comparing Bayes Model Averaging and Stacking When Model Approximation Error Cannot be Ignored. *Journal of Machine Learning Research*, 4:683–712, 2003.

[24] Pierre Cleroux. Reshoring in Canada: What is the economic data telling us — BDC.ca, 2023.

[25] Yarens J. Cruz, Marcelino Rivas, Ramón Quiza, Rodolfo E. Haber, Fernando Castaño, and Alberto Villalonga. A two-step machine learning approach for dynamic model selection: A case study on a micro milling process. *Computers in Industry*, 143:103764, 12 2022.

[26] Yuebang Dai, Hongkun Li, Zhaocheng Wei, and Haiyun Zhang. Chatter stability prediction for five-axis ball end milling with precise integration method. *Journal of Manufacturing Processes*, 32:20–31, 4 2018.

[27] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 7 2022.

[28] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18, 1 2000.

[29] B. D. Deebak and Fadi Al-Turjman. Digital-twin assisted: Fault diagnosis using deep transfer learning for machining tool condition. *International Journal of Intelligent Systems*, 2021.

[30] B. D. Deebak and Fadi Al-Turjman. Digital-twin assisted: Fault diagnosis using deep transfer learning for machining tool condition. *International Journal of Intelligent Systems*, 2021.

[31] B. D. Deebak and Fadi Al-Turjman. Digital-twin assisted: Fault diagnosis using deep transfer learning for machining tool condition. *International Journal of Intelligent Systems*, 2021.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm . *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 9 1977.

[33] Jianming Dou, Chuangwen Xu, Shengjie Jiao, Baodong Li, Jilin Zhang, and Xinxin Xu. An unsupervised online monitoring method for tool wear using a sparse auto-encoder. *The International Journal of Advanced Manufacturing Technology*, 106:2493–2507, 2019.

[34] Jonathan Downey, Denis O'Sullivan, Miroslaw Nejmen, Sebastian Bombinski, Paul O'Leary, Ramesh Raghavendra, and Krzysztof Jemielniak. Real Time Monitoring of the CNC Process in a Production Environment- the Data Collection & Analysis Phase. In *Procedia CIRP*, volume 41, pages 920–926. Elsevier B.V., 2016.

[35] João A. Duro, Julian A. Padget, Chris R. Bowen, H. Alicia Kim, and Aydin Nassehi. Multi-sensor data fusion framework for CNC machining monitoring. *Mechanical Systems and Signal Processing*, 66-67:505–520, 2016.

[36] Ryan Elwell and Robi Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 10 2011.

[37] Gizem Erdogan. Land selection criteria for lights out factory districts during the industry 4.0 process. *Journal of Urban Management*, 8(3):377–385, 12 2019.

[38] Cheng Fan, Yongjun Sun, Fu Xiao, Jie Ma, Dasheng Lee, Jiayuan Wang, and Yen Chieh Tseng. Statistical investigations of transfer learning-based methodology for short-term building energy predictions. *Applied Energy*, 262:114499, 3 2020.

[39] FANUC. Machine Tending Robots — FANUC America, 2024.

[40] Oier Franco, Xavier Beudaert, Alex Iglesias, Zoltan Dombovari, Kaan Erkorkmaz, and Jokin Munoa. Optimal cutting condition selection for high quality receptance measurements by sweep milling force excitation. *International Journal of Machine Tools and Manufacture*, 176, 5 2022.

[41] Yoav Freund and Robert E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 8 1997.

[42] Haining Gao, Hongdan Shen, Lei Yu, Wang Yinling, Rongyi Li, and Babar Nazir. Milling Chatter Detection System Based on Multi-Sensor Signal Fusion. *IEEE Sensors Journal*, 21(22):25243–25251, 11 2021.

[43] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. In *European Symposium on Artificial Neural Networks*, bruges, Belgium, 2016.

[44] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 10 2000.

[45] McKinsey Global Institute San Francisco. Offshoring: Is It a Win-Win Game?, 2003.

[46] P. Gray, G. Poon, G. Israeli, S. Bedi, S. Mann, and D. Miller. Model-To-Part: A road map for the CNC machine of the future. *CISM International Centre for Mechanical Sciences, Courses and Lectures*, 486:247–256, 2005.

[47] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander J Smola. A Kernel Method for the Two-Sample-Problem. *Neural Information Processing Systems*, 2008.

[48] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Alexander Smola, Bernhard Schölkopf, and Alexander Smola GRETTON. A Kernel Two-Sample Test Bernhard Schölkopf. *Journal of Machine Learning Research*, 13:723–773, 2012.

[49] Tian Guo, Zhao Xu, Xin Yao, Haifeng Chen, Karl Aberer, and Koichi Funaya. Robust online time series prediction with recurrent neural networks. *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pages 816–825, 12 2016.

[50] Parth Gupta, Rafael E. Banchs, and Paolo Rosso. Squeezing bottlenecks: Exploring the limits of autoencoder semantic representation capabilities. *Neurocomputing*, 175:1001–1008, 1 2016.

[51] Xiaozhen Han and Ran Jin. A Small Sample Image Recognition Method Based on ResNet and Transfer Learning. *Proceedings - 2020 5th International Conference on Computational Intelligence and Applications, ICCIA 2020*, pages 76–81, 6 2020.

[52] Guo Hao and Zhu Kunpeng. Pyramid LSTM auto-encoder for tool wear monitoring. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE).*, Online Zoom Meeting, 8 2020. IEEE.

[53] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu. Efficient and Privacy-Enhanced Federated Learning for Industrial Artificial Intelligence. *IEEE Transactions on Industrial Informatics*, 16(10):6532–6542, 10 2020.

[54] Hart Sean. Necessity is the Mother of Invention: U.S. Manufacturing Onshoring to Overcome Pandemic-Induced Supply Chain Challenges and Demand Shock — POWERS™ Management Consulting, 2022.

[55] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2454 LNCS:170–180, 2002.

[56] Dan He, Zexing Ni, and Xiufeng Wang. Online grinding chatter detection based on minimum entropy deconvolution and autocorrelation function. *International Journal of Advanced Manufacturing Technology*, 120(9-10):6175–6185, 6 2022.

[57] Vinit Hegiste, Tatjana Legler, and Martin Ruskowski. Application of federated learning in manufacturing. In *Proceedings of the International Conference on Industry 4.0 Technology*, 8 2022.

[58] Sepp Hochreiter and J Urgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[59] Ruei Jie Hsieh, Jerry Chou, and Chih Hsiang Ho. Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing. In *Proceedings - 2019 IEEE 12th Conference on Service-Oriented Computing and Applications, SOCA 2019*, pages 90–97. Institute of Electrical and Electronics Engineers Inc., 11 2019.

[60] Yao Wen Hsueh and Chan Yun Yang. Prediction of tool breakage in face milling using support vector machine. *International Journal of Advanced Manufacturing Technology*, 37(9-10):872–880, 6 2008.

[61] Yao Wen Hsueh and Chan Yun Yang. Tool breakage diagnosis in face milling by support vector machine. *Journal of Materials Processing Technology*, 209(1):145–152, 1 2009.

[62] Qingqing Huang, Zhen Kang, Yan Zhang, and Dong Yan. Tool remaining useful life prediction based on edge data processing and LSTM recurrent neural network. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, volume 2020-June. Prognostics and Health Management Society, 6 2020.

[63] Jui Pin Hung, Yuan Lung Lai, Ching Yuan Lin, and Tzu Liang Lo. Modeling the machining stability of a vertical milling machine under the influence of the preloaded linear guide. *International Journal of Machine Tools and Manufacture*, 51(9):731–739, 9 2011.

[64] Truong Thu Huong, Ta Phuong Bac, Dao Minh Long, Tran Duc Luong, Nguyen Minh Dan, Le Anh Quang, Le Thanh Cong, Bui Doan Thang, and Kim Phuc Tran. Detecting cyberattacks using anomaly detection in industrial control systems: A Federated Learning approach. *Computers in Industry*, 132:103509, 11 2021.

[65] IBM. What is Industry 4.0 and how does it work?

[66] IBM. What are Recurrent Neural Networks?, 2020.

[67] Integrity Data Services Pvt. Ltd. Offshoring vs. Onshoring: Choosing the Right Approach for Your Financial Services, 2023.

[68] Inventables. X-Carve Pro, 2022.

[69] F. Ismail and R. Ziaei. Chatter suppression in five-axis machining of flexible parts. *International Journal of Machine Tools and Manufacture*, 42(1):115–122, 1 2002.

[70] Keshav H. Jatakar, Gopal Mulgund, Abhishek D. Patange, B. B. Deshmukh, and Kishor S. Rambhad. Multi-Point Face Milling Tool Condition Monitoring Through Vibration Spectrogram and LSTM-Autoencoder. *International Journal of Performability Engineering*, 18(8):570, 8 2022.

[71] Yakun Jiang, Jihong Chen, Huicheng Zhou, Jianzhong Yang, Pengcheng Hu, and Junxiang Wang. Contour error modeling and compensation of CNC machining based on deep learning and reinforcement learning. *The International Journal of Advanced Manfacturing Technology*, 2022.

[72] Johnson Controls. Pattern Recognition Adaptive Control (PRAC+) overview • EasyIO PID/PRAC+ Commissioning Guide • Reader • OpenBlue, Building Automation and Controls Knowledge Exchange, 2024.

[73] Mohan Krishna K and Mangala M. Patil. CNC Machine Condition Monitoring System Using LSTM Based Deep Learning Techniques. In *OITS International Conference on Information Technology*, pages 42–47. Institute of Electrical and Electronics Engineers (IEEE), 3 2022.

[74] Tobias Kalb, Masoud Roschani, Miriam Ruf, and Jürgen Beyerer. Continual Learning for Class-and Domain-Incremental Semantic Segmentation. In *IEEE Intelligent Vehicles Symposium*, Nagoya, Japan, 7 2021. IEEE.

[75] Prakhar Kaushik, Alex Gain, Adam Kortylewski, and Alan Yuille. Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping. In *5th Workshop on Meta-Learning at NeurIPS*, 2021.

[76] Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. CLASSIC: Continual and Contrastive Learning of Aspect Sentiment Classification Tasks. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 6871–6883, 12 2021.

[77] Asif Khan, Hyunho Hwang, and Heung Soo Kim. Synthetic data augmentation and deep learning for the fault diagnosis of rotating machines. *Mathematics*, 9(18), 9 2021.

[78] Latif U. Khan, Madyan Alsenwi, Ibrar Yaqoob, Muhammad Imran, Zhu Han, and Choong Seon Hong. Resource optimized federated learning-enabled cognitive internet of things for smart industries. *IEEE Access*, 8:168854–168864, 2020.

[79] Sarit Khirirat, Hamid Reza Feyzmahdavian, and Mikael Johansson. Mini-batch gradient descent faster convergence under data sparsity. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Melbourne, Australia, 12 2017. IEEE.

[80] Young Min Kim, Seung Jun Shin, and Hae Won Cho. Predictive Modeling for Machining Power Based on Multi-source Transfer Learning in Metal Cutting. *International Journal of Precision Engineering and Manufacturing - Green Technology*, 9(1):107–125, 1 2022.

[81] Young Min Kim, Seung Jun Shin, and Hae Won Cho. Predictive Modeling for Machining Power Based on Multi-source Transfer Learning in Metal Cutting. *International Journal of Precision Engineering and Manufacturing - Green Technology*, 9(1):107–125, 1 2022.

[82] Young Min Kim, Seung Jun Shin, and Hae Won Cho. Predictive Modeling for Machining Power Based on Multi-source Transfer Learning in Metal Cutting. *International Journal of Precision Engineering and Manufacturing - Green Technology*, 9(1):107–125, 1 2022.

[83] Paweł Knap, Urszula Jachymczyk, and Patryk Bałazy. Tool Wear Detection in Milling Process Using Long Short-Term Memory Networks: An Industry 4.0 Approach. *International Scientific Journals of Scientific Technical Union of Mechanical Engineering*, 17(4):148–151, 2023.

[84] Mehmet Erdi Korkmaz, Munish Kumar Gupta, Zhixiong Li, Grzegorz M. Krolczyk, Mustafa Kuntoglu, Rüstem Binali, Nafiz Yasar, and Danil Yu Pimenov. Indirect monitoring of machining characteristics via advanced sensor systems: a critical review, 6 2022.

[85] Cheick Abdoul Kadir A. Kounta, Lionel Arnaud, Bernard Kamsu-Foguem, and Fana Tangara. Review of AI-based methods for chatter detection in machining based on bibliometric analysis, 9 2022.

[86] Bartosz Krawczyk and Michał Woźniak. One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing*, 19(12):3387–3400, 12 2015.

[87] E. Kuljanic, M. Sortino, and G. Totis. Multisensor approaches for chatter detection in milling. *Journal of Sound and Vibration*, 312(4-5):672–693, 5 2008.

[88] E. Kuljanic, G. Totis, and M. Sortino. Development of an intelligent multisensor chatter detection system in milling. *Mechanical Systems and Signal Processing*, 23(5):1704–1718, 7 2009.

[89] Mustafa Kuntoglu, Abdullah Aslan, Danil Yurievich Pimenov, Usame Ali Usca, Emin Salur, Munish Kumar Gupta, Tadeusz Mikolajczyk, Khaled Giasin, Wojciech Kapłonek, and Shubham Sharma. A review of indirect tool condition monitoring systems and decision-making methods in turning: Critical analysis and trends, 1 2021.

[90] Mustafa Kuntoglu, Emin Salur, Munish Kumar Gupta, Murat Sarıkaya, Danil, and Yu Pimenov. A state-of-the-art review on sensors and signal processing systems in mechanical machining processes. *The International Jounral of Advanced Manufacturing Technology*, 2021.

[91] We Feng Kuo, Bo Min Huang, and Ching Hung Lee. Development of Virtual Milling System Using Data Fusion and Transfer Learning. In *Proceedings - 2020 International Conference on Pervasive Artificial Intelligence, ICPAI 2020*, pages 253–257. Institute of Electrical and Electronics Engineers Inc., 12 2020.

[92] We Feng Kuo, Bo Min Huang, and Ching Hung Lee. Development of Virtual Milling System Using Data Fusion and Transfer Learning. In *Proceedings - 2020 International Conference on Pervasive Artificial Intelligence, ICPAI 2020*, pages 253–257. Institute of Electrical and Electronics Engineers Inc., 12 2020.

[93] We Feng Kuo, Bo Min Huang, and Ching Hung Lee. Development of Virtual Milling System Using Data Fusion and Transfer Learning. In *Proceedings - 2020 International Conference on Pervasive Artificial Intelligence, ICPAI 2020*, pages 253–257. Institute of Electrical and Electronics Engineers Inc., 12 2020.

[94] Ilarion Kvinevskiy, Sanjeev Bedi, and Stephen Mann. Detecting machine chatter using audio data and machine learning. *International Journal of Advanced Manufacturing Technology*, 108(11-12):3707–3716, 6 2020.

[95] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series Extreme Event Forecasting with Neural Networks at Uber. Technical report, Uber, 2017.

[96] Eugene Li, Sanjeev Bedi, and William Melek. Anomaly detection in three-axis CNC machines using LSTM networks and transfer learning. *The International Journal of Advanced Manufacturing Technology*, 127:5185–5198, 2023.

[97] Eugene Li, Yang Li, Sanjeev Bedi, William Melek, and Paul Gray. Incremental learning of LSTM-autoencoder anomaly detection in three-axis CNC machines. *International Journal of Advanced Manufacturing Technology*, 130(3):1265–1277, 12 2023.

[98] Gen Li and Jason J. Jung. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, 91:93–102, 3 2023.

[99] Guang Li, Yan Fu, Duanbing Chen, Lulu Shi, and Junlin Zhou. Deep anomaly detection for CNC machine cutting tool using spindle current signals. *Sensors (Switzerland)*, 20(17):1–18, 9 2020.

[100] Guang Li, Xin Yang, Duanbing Chen, Anxing Song, Yuke Fang, and Junlin Zhou. Tool breakage detection using deep learning. In *Proceedings - 2018 IEEE/ACIS 3rd*

International Conference on Big Data, Cloud Computing, Data Science and Engineering, BCD 2018, pages 37–42. Institute of Electrical and Electronics Engineers Inc., 11 2018.

[101] Jianbo Li, Juan Lu, Chaoyi Chen, Junyan Ma, and Xiaoping Liao. Tool wear state prediction based on feature-based transfer learning. *The International Journal of Advanced Manufacturing Technology*, 2021.

[102] Jianbo Li, Juan Lu, Chaoyi Chen, Junyan Ma, and Xiaoping Liao. Tool wear state prediction based on feature-based transfer learning. *The International Journal of Advanced Manufacturing Technology*, 2021.

[103] Jianbo Li, Juan Lu, Chaoyi Chen, Junyan Ma, and Xiaoping Liao. Tool wear state prediction based on feature-based transfer learning. *The International Journal of Advanced Manufacturing Technology*, 2021.

[104] Jiangang Li, Changgui Qi, Yanan Li, and Zenghao Wu. Prediction and Compensation of Contour Error of CNC Systems Based on LSTM Neural-Network. *IEEE/ASME Transactions on Mechatronics*, 27(1):572–581, 2 2022.

[105] Jianhui Li, Z. Murat Kilic, and Yusuf Altintas. General Cutting Dynamics Model for Five-Axis Ball-End Milling Operations. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 142(12), 12 2020.

[106] W. D. Li and Y. C. Liang. Deep transfer learning based diagnosis for machining process lifecycle. *Procedia CIRP*, 90:642–647, 1 2020.

[107] Xuebing Li, Xianli Liu, Caixu Yue, Steven Y. Liang, and Lihui Wang. Systematic review on tool breakage monitoring techniques in machining operations, 5 2022.

[108] Y. C. Liang, S. Wang, W. D. Li, and X. Lu. Data-Driven Anomaly Diagnosis for Machining Processes. *Engineering*, 5(4):646–652, 8 2019.

[109] Benjamin Lindemann, Fabian Fesenmayr, Nasser Jazdi, and Michael Weyrich. Anomaly detection in discrete manufacturing using self-learning approaches. *Procedia CIRP*, 79:313–318, 2019.

[110] Benjamin Lindemann, Benjamin Maschler, Nada Sahlab, and Michael Weyrich. A survey on anomaly detection for technical systems using LSTM networks, 10 2021.

[111] C. Liu, G. F. Wang, and Z. M. Li. Incremental learning for online tool condition monitoring using Ellipsoid ARTMAP network model. *Applied Soft Computing*, 35:186–198, 10 2015.

[112] Xiaobo Liu, Zhentao Liu, Guangjun Wang, Zhihua Cai, and Harry Zhang. Ensemble Transfer Learning Algorithm. *IEEE Access*, 6:2389–2396, 12 2017.

[113] Yu Chi Liu, Kun Ying Li, and Yao Cheng Tsai. Spindle thermal error prediction based on lstm deep learning for a cnc machine tool. *Applied Sciences (Switzerland)*, 11(12), 6 2021.

[114] Yong Luo, Liancheng Yin, Wenchao Bai, and Keming Mao. An Appraisal of Incremental Learning Methods. *Entropy 2020, Vol. 22, Page 1190*, 22(11):1190, 10 2020.

[115] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. In *2016 Anomaly Detection Workshop*, 7 2016.

[116] Mohamed Marei and Weidong Li. Cutting tool prognostics enabled by hybrid CNN-LSTM with transfer learning. *The International Journal of Advanced Manufacturing Technology*, 2022.

[117] Kamil Adipovich Masalimov. A Machine Learning based Approach to Autogenerate Diagnostic Models for CNC machines. In *Proceedings - 2020 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020*, pages 1358–1360. Institute of Electrical and Electronics Engineers Inc., 9 2020.

[118] Benjamin Maschler and Michael Weyrich. Deep Transfer Learning for Industrial Automation: A Review and Discussion of New Techniques for Data-Driven Machine Learning, 6 2021.

[119] Mathworks. Incremental Anomaly Detection Overview - MATLAB & Simulink, 2024.

[120] Seifeddine Mejri, Vincent Gagnol, Thien Phu Le, Laurent Sabourin, Pascal Ray, and Patrick Paultre. Dynamic characterization of machining robot and stability analysis. *The International Journal of Advanced Manufacturing Technology 2015 82:1*, 82(1):351–359, 6 2015.

[121] Laura Melgar-García, David Gutiérrez-Avilés, Cristina Rubio-Escudero, and Alicia Troncoso. Identifying novelties and anomalies for incremental learning in streaming time series forecasting. *Engineering Applications of Artificial Intelligence*, 123:106326, 8 2023.

[122] Xianghu Meng, Jun Li, Mengchu Zhou, and Xianzhong Dai. Improved population-based incremental learning algorithm for scheduling multi-bridge waterjet cutting processes. *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control, ICNSC 2014*, pages 496–500, 2014.

[123] M Jehanzeb Mirza, Marc Masana, Horst Possegger, and Horst Bischof. An Efficient Domain-Incremental Learning Approach to Drive in All Weather Conditions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 3000–3010, 2022.

[124] K. V.V.N.R.Chandra Mouli, Balla Srinivasa Prasad, A. V. Sridhar, and Sandeep Alanka. A review on multi sensor data fusion technique in CNC machining of tailor-made nanocomposites, 5 2020.

[125] Michael D. Muhlbaier, Apostolos Topalis, and Robi Polikar. Learn++.NC. *IEEE Transactions on Neural Networks*, 20(1):152–168, 1 2009.

[126] Jun Sik Nam and Won Tae Kwon. A Study on Tool Breakage Detection During Milling Process Using LSTM-Autoencoder and Gaussian Mixture Model. *International Journal of Precision Engineering and Manufacturing*, 23(6):667–675, 6 2022.

[127] Parag Narkhede, Rahee Walambe, Shashi Poddar, and Ketan Kotecha. Incremental learning of LSTM framework for sensor fusion in attitude estimation. *PeerJ Computer Science*, 7:1–18, 2021.

[128] Parag Narkhede, Rahee Walambe, Shashi Poddar, and Ketan Kotecha. Incremental learning of LSTM framework for sensor fusion in attitude estimation. *PeerJ Computer Science*, 7:1–18, 2021.

[129] John Henry Navarro-Devia, Yun Chen, Dzung Viet Dao, and Huaizhong Li. Chatter detection in milling processes—a review on signal processing and condition classification. *The International Journal of Advanced Manufacturing Technology 2023 125:9*, 125(9):3943–3980, 2 2023.

[130] A. P. Nemirko and J. H. Dulá. Machine learning algorithm based on convex hull analysis. *Procedia Computer Science*, 186:381–386, 1 2021.

[131] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, Jun Li, Dusit Niyato, and H. Vincent Poor. Federated Learning for Industrial Internet of Things in Future Industries. *IEEE Wireless Communications*, 28(6):192–199, 12 2021.

[132] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad. Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, 57:102282, 4 2021.

[133] Huu Du Nguyen and Kim Phuc Tran. Artificial Intelligence for Smart Manufacturing in Industry 5.0: Methods, Applications, and Challenges. *Springer Series in Reliability Engineering*, Part F4:5–33, 2023.

[134] Jinqiu Pan, Zhibing Liu, Xibin Wang, Che Chen, and Xiaoyu Pan. Boring chatter identification by multi-sensor feature fusion and manifold learning. *International Journal of Advanced Manufacturing Technology*, 2020.

[135] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning, 2010.

[136] Daehyung Park, Yuuna Hoshi, and Charles C. Kemp. A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 7 2018.

[137] Myon Woong Park, Byoung Tae Park, Hyung Min Rho, and Sheung Kown Kim. Incremental Supervised Learning of Cutting Conditons Using the Fuzzy ARTMAP Neural Network. *CIRP Annals*, 49(1):375–378, 1 2000.

[138] PES Media. Adding multi-axis capacity - PES Media, 2020.

[139] Kemal Polat. The Fault Diagnosis based on Deep Long Short-Term Memory Model from the Vibration Signals in the Computer Numerical Control Machines. *Journal of the Institute of Electronics and Computer*, 2(1):72–92, 2020.

[140] Robi Polikar, Lalita Udpa, Satish S. Udpa, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 31(4):497–508, 11 2001.

[141] M. Postel, B. Bugdayci, and K. Wegener. Ensemble transfer learning for refining stability predictions in milling using experimental stability states. *International Journal of Advanced Manufacturing Technology*, 107(9-10):4123–4139, 4 2020.

[142] Mahardhika Pratama, Jie Lu, Edwin Lughofer, Guangquan Zhang, and Meng Joo Er. An Incremental Learning of Concept Drifts Using Evolving Type-2 Recurrent Fuzzy Neural Networks. *IEEE Transactions on Fuzzy Systems*, 25(5):1175–1192, 10 2017.

[143] Quan Qian, Yi Wang, Taisheng Zhang, and Yi Qin. Maximum mean square discrepancy: A new discrepancy representation metric for mechanical fault transfer diagnosis. *Knowledge-Based Systems*, 276:110748, 9 2023.

[144] Yi Qin, Quan Qian, Jun Luo, and Huayan Pu. Deep Joint Distribution Alignment: A Novel Enhanced-Domain Adaptation Mechanism for Fault Transfer Diagnosis. *IEEE Transactions on Cybernetics*, 53(5):3128–3138, 5 2023.

[145] Youyang Qu, Shiva Raj Pokhrel, Sahil Garg, Longxiang Gao, and Yong Xiang. A Blockchained Federated Learning Framework for Cognitive Computing in Industry 4.0 Networks. *IEEE Transactions on Industrial Informatics*, 17(4):2964–2973, 4 2021.

[146] Kirthanaa Raghuraman, Monisha Senthurpandian, Monisha Shanmugasundaram, Bhargavi, and V. Vaidehi. Online Incremental Learning Algorithm for anomaly detection and prediction in health care. *2014 International Conference on Recent Trends in Information Technology, ICRTIT 2014*, 12 2014.

[147] M. Hossein Rahimi, Hoai Nam Huynh, and Yusuf Altintas. On-line chatter detection in milling with hybrid machine learning and physics-based model. *CIRP Journal of Manufacturing Science and Technology*, 35:25–40, 11 2021.

[148] Khandakar M. Rashid and Joseph Louis. Times-series data augmentation and deep learning for construction equipment activity recognition. *Advanced Engineering Informatics*, 42, 10 2019.

[149] P. Rawal, C. Brillinger, C. Böhlmann, and W. Hintze. Sensor based online quality monitoring system for detection of milling defects on CFRP structures. *CEAS Aeronautical Journal*, 11(2):565–574, 6 2020.

[150] Muhammad Rizal, Jaharah A. Ghani, Mohd Zaki Nuawi, and Che Hassan Che Haron. A review of sensor system and application in milling process for tool condition monitoring. *Research Journal of Applied Sciences, Engineering and Technology*, 7(10):2083–2097, 2014.

[151] Robotiq. Machine tending, 2024.

[152] Luis Romeral Martínez, Roque A Osornio Rios, and Miguel Delgado Prieto. *New Trends in the Use of Artificial Intelligence for the Industry 4.0*. IntechOpen, London, United Kingdom, 2020.

[153] David J Roth. *Mechanistic Modelling of 5-Axis Machining*. PhD thesis, University of Waterloo, Waterloo, 2004.

[154] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A Unifying Review of Deep and Shallow Anomaly Detection. In *Proceedings of the IEEE*, pages 756–795. IEEE, 5 2020.

[155] Mahmoud Said Elsayed, Nhien An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. Network Anomaly Detection Using LSTM Based Autoencoder. *Q2SWinet 2020 - Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 37–45, 11 2020.

[156] Thompson Sarkodie-Gyan, Wahyu Caesarendra, Tobias Schlagenhauf, Jan Wolf, and Alexander Puchta. Convolutional-Based Encoder–Decoder Network for Time Series Anomaly Detection during the Milling of 16MnCr5. *Data 2022, Vol. 7, Page 175*, 7(12):175, 12 2022.

[157] Sameer Sayyad, Satish Kumar, Arunkumar Bongale, Ketan Kotecha, Ganeshsree Selvachandran, and Ponnuthurai Nagaratnam Suganthan. Tool wear prediction using long short-term memory variants and hybrid feature selection techniques. *International Journal of Advanced Manufacturing Technology*, 121(9-10):6611–6633, 8 2022.

[158] Jeffrey C. Schlimmer and Richard H. Granger. Incremental learning from noisy data. *Machine Learning 1986 1:3*, 1(3):317–354, 9 1986.

[159] Steven R. Schmid and Shreyes N. Melkote. Manufacturing and the Great Resignation. *Mechanical Engineering*, 144(3):38–43, 5 2022.

[160] Tony L. Schmitz and K. Scott Smith. Machining Dynamics. *Machining Dynamics*, 2019.

[161] G Serin, B Sener, A M Ozbayoglu, and H O Unver. Review of tool condition monitoring in machining and opportunities for deep learning. *The International Journal of Advanced Manufacturing Technology*, 2020.

[162] Saeed Shafieian and Mohammad Zulkernine. Multi-layer stacking ensemble learners for low footprint network intrusion detection. *Complex and Intelligent Systems*, 9(4):3787–3799, 8 2022.

[163] Ilya Shashkov, Alexey Zaytsev, Nikita Balabin, and Evgeny Burnaev. Transfer learning for ensembles: reducing computation time and keeping the diversity. *ACM International Conference Proceeding Series*, pages 8–13, 6 2022.

[164] Bin Shi and Helmi Attia. Current status and future direction in the numerical modeling and simulation of machining processes: A critical literature review, 4 2010.

[165] Fei Shi, Hongrui Cao, Yuke Wang, Boyuan Feng, and Yufei Ding. Chatter detection in high-speed milling processes based on ON-LSTM and PBT. *International Journal of Advanced Manufacturing Technology*, 111(11-12):3361–3378, 12 2020.

[166] Fei Shi, Hongrui Cao, Yuke Wang, Boyuan Feng, and Yufei Ding. Chatter detection in high-speed milling processes based on ON-LSTM and PBT. *International Journal of Advanced Manufacturing Technology*, 111(11-12):3361–3378, 12 2020.

[167] Konstantin Shmelkov, Cordelia Schmid Inria, and Karteek Alahari. Incremental Learning of Object Detectors without Catastrophic Forgetting. *International Conference on Computer Vision*, 2017.

[168] Barry Siegel. Industrial Anomaly Detection: A Comparison of Unsupervised Neural Network Architectures. *IEEE Sensors Letters*, 4(8), 8 2020.

[169] Joseph Sill, Gabor Takacs, Lester Mackey, and David Lin. Feature-Weighted Linear Stacking. *arXiv*, 11 2009.

[170] Kamilya Smagulova and Alex Pappachen James. A survey on LSTM memristive neural network architectures and applications. *Eur. Phys. J. Special Topics*, 228:2313–2324, 2019.

[171] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4754 LNAI:13–31, 2007.

[172] Peter Sollich and Anders Krogh. Learning with ensembles: How over-fitting can be useful. *Advances in Neural Information Processing*, 8, 1995.

[173] Salvatore Sorrentino, Francesco Manetti, Arianna Bresci, Federico Vernuccio, Chiara Ceconello, Silvia Ghislanzoni, Italia Bongarzone, Renzo Vanna, Giulio Cerullo, and Dario Polli. Deep ensemble learning and transfer learning methods for classification of senescent cells from nonlinear optical microscopy images. *Frontiers in Chemistry*, 11:1213981, 6 2023.

[174] Staff. A Look into the History of Milling Machines - Summit Machine Tool.

[175] Staff. Golden Era of Onshoring — Manufacturing.net, 2022.

[176] Staff Writer. CNC Machining Projected to Become $129 Billion Industry by 2026, 2020.

[177] Nick Stonnington. Why Reshoring U.S. Manufacturing Could Be The Wave Of The Future, 2020.

[178] Federico Succetti, Andrea Ceschini, Francesco Di Luzio, Antonello Rosato, and Massimo Panella. Time Series Prediction with Autoencoding LSTM Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12862 LNCS:306–317, 2021.

[179] Chuang Sun, Meng Ma, Zhibin Zhao, Shaohua Tian, Ruqiang Yan, and Xuefeng Chen. Deep Transfer Learning Based on Sparse Autoencoder for Remaining Useful Life Prediction of Tool in Manufacturing. *IEEE Transactions on Industrial Informatics*, 15(4):2416–2425, 4 2019.

[180] Shixu Sun, Xiaofeng Hu, and Yingchao Liu. An imbalanced data learning method for tool breakage detection based on generative adversarial networks. *Journal of Intelligent Manufacturing*, 2021.

[181] Shixu Sun, Xiaofeng Hu, and Wenjuan Zhang. Detection of tool breakage during milling process through acoustic emission. *The International Journal of Advanced Manufacturing Technology*, 2020.

[182] Yuwen Sun, Meng Zheng, Shanglei Jiang, Danian Zhan, and Ruoqi Wang. A State-of-the-Art Review on Chatter Stability in Machining Thin Walled Parts. *Machines 2023, Vol. 11, Page 359*, 11(3):359, 3 2023.

[183] Onur Surucu, Stephen Andrew Gadsden, and John Yawney. Condition Monitoring using Machine Learning: A Review of Theory, Applications, and Recent Advances. *Expert Systems with Applications*, 221:119738, 7 2023.

[184] Agust Susanto, Keiji Yamada, Mani Kyousuke, Ryutaro Tanaka, and Katsuhiko Sekiya. Vibration Analysis in Milling of Thin-Walled Workpieces using Hilbert-Huang Transform. In *Proceedings of The 9th International Conference on Leading Edge Manufacturing in 21st century*, 2017.

[185] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11141 LNCS:270–279, 2018.

[186] Yingshui Tan, Baihong Jin, Alexander Nettekoven, Yuxin Chen, Yisong Yue, Ufuk Topcu, and Alberto Sangiovanni-Vincentelli. An encoder-decoder based approach for anomaly detection with application in additive manufacturing. In *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, pages 1008–1015. Institute of Electrical and Electronics Engineers Inc., 12 2019.

[187] Yuyang Tang, Jun Zhang, Hui Tian, Hongguang Liu, and Wanhua Zhao. Optimization method of spindle speed with the consideration of chatter and forced vibration for five-axis flank milling. *International Journal of Advanced Manufacturing Technology*, 125(7-8):3159–3169, 4 2023.

[188] Changle Tian, Guanghui Zhou, Junjie Zhang, and Chao Zhang. Optimization of cutting parameters considering tool wear conditions in low-carbon manufacturing environment. *Journal of Cleaner Production*, 226:706–719, 7 2019.

[189] J. Tlusty and F. Ismail. Basic Non-Linearity in Machining Chatter. *CIRP Annals - Manufacturing Technology*, 30(1):299–304, 1981.

[190] Xin Tong, Qiang Liu, Liuquan Wang, and Pengpeng Sun. A Parameter Optimization Method for Chatter Stability in Five-Axis Milling. *Machines 2023, Vol. 11, Page 79*, 11(1):79, 1 2023.

[191] Hasan Torabi, Seyedeh Leili Mirtaheri, and Sergio Greco. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, 6(1):1–13, 12 2023.

[192] Minh-Quang Tran, Hoang-Phuong Doan, Viet Q. Vu, and Lien T. Vu. Machine Learning and IoT-based Approach for Tool Condition Monitoring: A Review and Future Prospects. *Measurement*, page 112351, 12 2022.

[193] Minh Quang Tran, Meng Kun Liu, and Mahmoud Elsisi. Effective multi-sensor data fusion for chatter detection in milling process. *ISA Transactions*, 125:514–527, 6 2022.

[194] M. M. Uddin, Stufflestreet B. A, and Johnson K. V. Identifying the Need for Trained Machinists in the Greater Tri-cities Area: A Survey of Employers to Evaluate the Future of Machining . In *ASEE Virtual Annual Conference*, 2021.

[195] Hakki Ozgur Unver and Batihan Sener. Exploring the Potential of Transfer Learning for Chatter Detection. In *Procedia Computer Science*, volume 200, pages 151–159. Elsevier B.V., 2022.

[196] Hakki Ozgur Unver and Batihan Sener. Exploring the Potential of Transfer Learning for Chatter Detection. In *Procedia Computer Science*, volume 200, pages 151–159. Elsevier B.V., 2022.

[197] Hakki Ozgur Unver and Batihan Sener. Exploring the Potential of Transfer Learning for Chatter Detection. In *Procedia Computer Science*, volume 200, pages 151–159. Elsevier B.V., 2022.

[198] Gido M Van De Ven and Andreas S Tolias. Three scenarios for continual learning. *CoRR*, 2019.

[199] Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence 2022 4:12*, 4(12):1185–1197, 12 2022.

[200] Rajiv Kumar Vashisht and Qingjin Peng. Online chatter detection for milling operations using LSTM neural networks assisted by motor current signals of ball screw drives. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 143(1), 1 2021.

[201] Rajiv Kumar Vashisht and Qingjin Peng. Online chatter detection for milling operations using LSTM neural networks assisted by motor current signals of ball screw drives. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, 143(1), 1 2021.

[202] Guofeng Wang, Zhiwei Guo, and Lei Qian. Online incremental learning for tool condition classification using modified Fuzzy ARTMAP network. *Journal of Intelligent Manufacturing*, 25(6):1403–1411, 11 2014.

[203] Huiju Wang, Mengxuan Li, and Xiao Yue. IncLSTM: Incremental Ensemble LSTM Model towards Time Series Data. *Computers & Electrical Engineering*, 92:107156, 6 2021.

[204] Juncheng Wang, Bin Zou, Mingfang Liu, Yishang Li, Hongjian Ding, and Kai Xue. Milling force prediction model based on transfer learning and neural network. *Journal of Intelligent Manufacturing*, 32(4):947–956, 4 2021.

[205] Juncheng Wang, Bin Zou, Mingfang Liu, Yishang Li, Hongjian Ding, and Kai Xue. Milling force prediction model based on transfer learning and neural network. *Journal of Intelligent Manufacturing*, 32(4):947–956, 4 2021.

[206] Xiaoding Wang, Sahil Garg, Hui Lin, Jia Hu, Georges Kaddoum, Md Jalil Piran, and M. Shamim Hossain. Toward Accurate Anomaly Detection in Industrial Internet of Things Using Hierarchical Federated Learning. *IEEE Internet of Things Journal*, 9(10):7110–7119, 5 2022.

[207] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time Series Data Augmentation for Deep Learning: A Survey. *IJCAI International Joint Conference on Artificial Intelligence*, pages 4653–4660, 2 2020.

[208] Jake Williams, Abel Tadesse, Tyler Sam, Huey Sun, and George D. Montañez. Limits of Transfer Learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12566 LNCS, pages 382–393. Springer Science and Business Media Deutschland GmbH, 2020.

[209] David H Wolpert and William G Macready. An Efficient Method To Estimate Bagging's Generalization Error. *Machine Learning*, 35:41–55, 1999.

[210] Guile Wu, Shaogang Gong, and Pan Li. Striking a Balance between Stability and Plasticity for Class-Incremental Learning. In *International Conference on Computer Vision*, 2021.

[211] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. In *International Conference on Computer Vision*, 2019.

[212] Guangda Xu, Huicheng Zhou, and Jihong Chen. CNC internal data based incremental cost-sensitive support vector machine method for tool breakage monitoring in end milling. *Engineering Applications of Artificial Intelligence*, 74:90–103, 9 2018.

[213] Wei Qi Yan. Transfer Learning and Ensemble Learning. In *Computational Methods of Deep Learning*, pages 191–203. Springer, Singapore, 2023.

[214] Xiaoliang Yan, Shreyes Melkote, Anant Kumar Mishra, and Sudhir Rajagopalan. A digital apprentice for chatter detection in machining via human–machine interaction. *Journal of Intelligent Manufacturing*, pages 1–14, 8 2022.

[215] Haw-Ching Yang. Roughness of Milling Process, 2020.

[216] Qing Yang, Yudi Gu, and Dongsheng Wu. Survey of incremental learning. *Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019*, pages 399–404, 6 2019.

[217] Melih C. Yesilli, Firas A. Khasawneh, and Andreas Otto. On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition. *CIRP Journal of Manufacturing Science and Technology*, 28:118–135, 1 2020.

[218] Melih C. Yesilli, Firas A. Khasawneh, and Andreas Otto. On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition. *CIRP Journal of Manufacturing Science and Technology*, 28:118–135, 1 2020.

[219] Roozbeh Yousefzadeh. Deep Learning Generalization and the Convex Hull of Training Sets. *Computer Science Bibliography*, 1 2021.

[220] Yu Yue Yu, Dong Zhang, Xiao Ming Zhang, Xiao Bo Peng, and Han Ding. Online stability boundary drifting prediction in milling process: An incremental learning approach. *Mechanical Systems and Signal Processing*, 173:109062, 7 2022.

[221] Fabio C. Zegarra, Juan Vargas-Machuca, and Alberto M. Coronado. Comparison of CNN and CNN-LSTM Architectures for Tool Wear Estimation. In *Proceedings of the 2021 IEEE Engineering International Research Conference, EIRCON 2021*. Institute of Electrical and Electronics Engineers Inc., 2021.

[222] Chao Zhang, Guanghui Zhou, Han Li, and Yan Cao. Manufacturing Blockchain of Things for the Configuration of a Data- And Knowledge-Driven Digital Twin Manufacturing Cell. *IEEE Internet of Things Journal*, 7(12):11884–11894, 12 2020.

[223] X. Y. Zhang, X. Lu, S. Wang, W. Wang, and W. D. Li. A multi-sensor based online tool condition monitoring system for milling process. In *Procedia CIRP*, volume 72, pages 1136–1141. Elsevier B.V., 2018.

[224] Xiaoyang Zhang, Xin Lu, Weidong Li, and Sheng Wang. Prediction of the remaining useful life of cutting tool using the Hurst exponent and CNN-LSTM. *The International Journal of Advanced Manufacturing Technology*, 2021.

[225] Binjie Zhao, Jiahao Nie, Siwei Guan, Han Wang, Zhiwei He, and Mingyu Gao. Hierarchical Feature Fusion based Reconstruction Network for Unsupervised Anomaly Detection. In *IEEE 27th International Conference on Emergin Technologies and Factory Automation*, 2022.

[226] Mingwei Zhao, Caixu Yue, and Xianli Liu. Research on milling chatter identification of thin-walled parts based on incremental learning and multi-signal fusion. *International Journal of Advanced Manufacturing Technology*, 125(9-10):3925–3941, 4 2023.

[227] Rui Zhao, Ruqiang Yan, Jinjiang Wang, and Kezhi Mao. Learning to monitor machine health with convolutional Bi-directional LSTM networks. *Sensors (Switzerland)*, 17(2), 2 2017.

[228] Yuchao Zheng, Chen Li, Xiaomin Zhou, Haoyuan Chen, Hao Xu, Yixin Li, Haiqing Zhang, Xiaoyan Li, Hongzan Sun, Xinyu Huang, and Marcin Grzegorzek. Application of transfer learning and ensemble learning in image-level classification for breast histopathology. *Intelligent Medicine*, 3(2):115–128, 5 2023.

[229] Guanghui Zhou, Shengze Yuan, Qi Lu, and Xiaoping Xiao. A carbon emission quantitation model and experimental evaluation for machining process considering tool wear condition. *The International Journal of Advanced Manufacturing Technology565*, 98:–577, 2018.

[230] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, 12B:11106–11115, 12 2020.

[231] Kunpeng P. Zhu, Yoke San Wong, and Geok Soon Hong. Wavelet analysis of sensor signals for tool condition monitoring: A review and some new results. *International Journal of Machine Tools and Manufacture*, 49(7-8):537–553, 6 2009.

[232] Yitan Zhu, Thomas Brettin, Yvonne A. Evrard, Alexander Partin, Fangfang Xia, Maulik Shukla, Hyunseung Yoo, James H. Doroshow, and Rick L. Stevens. Ensemble transfer learning for the prediction of anti-cancer drug response. *Scientific Reports 2020 10:1*, 10(1):1–11, 10 2020.

APPENDICES

Appendix A

Network design for anomaly detection systems

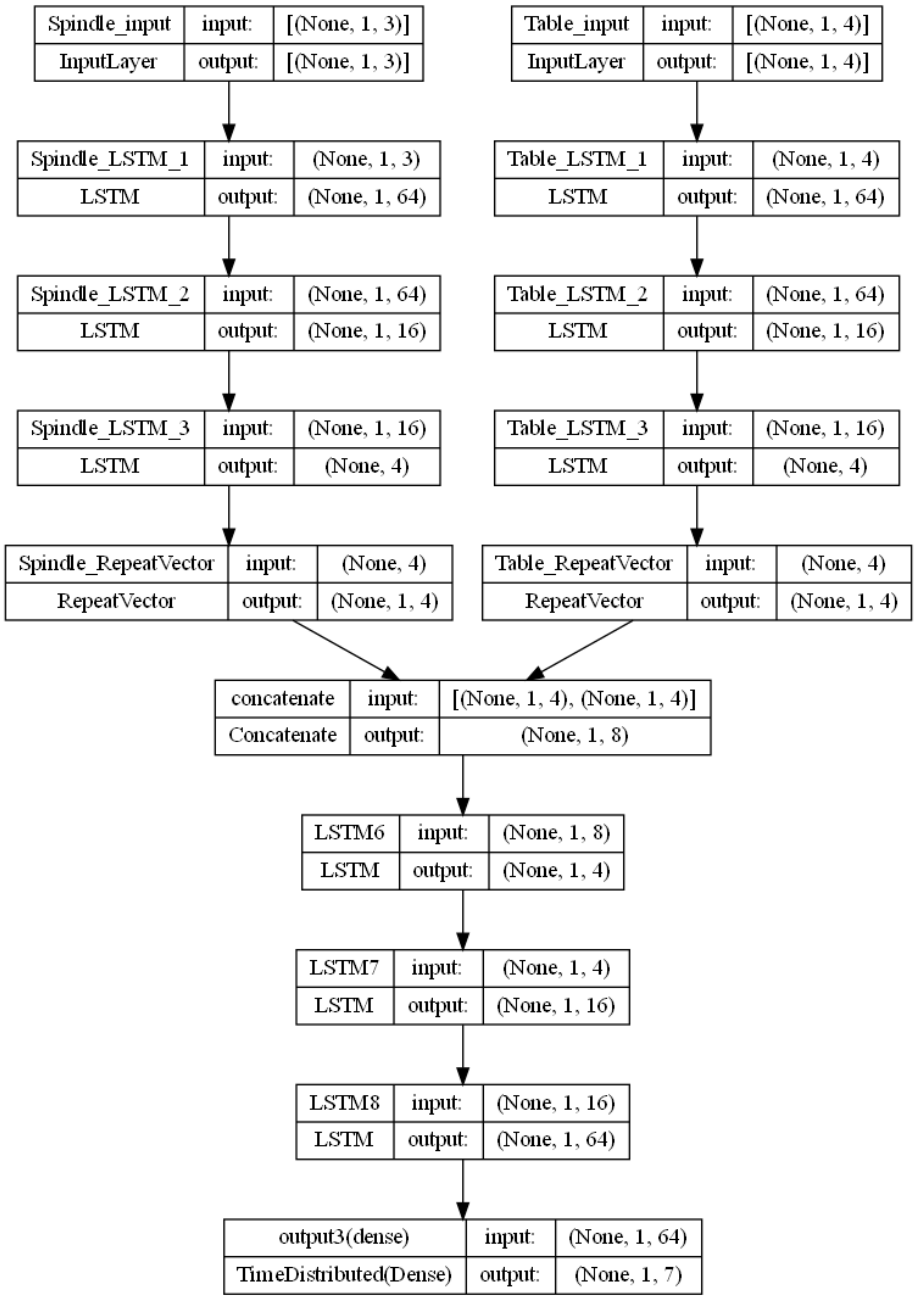


Figure A.1: Network Configuration for Five Axis Anomaly Detection

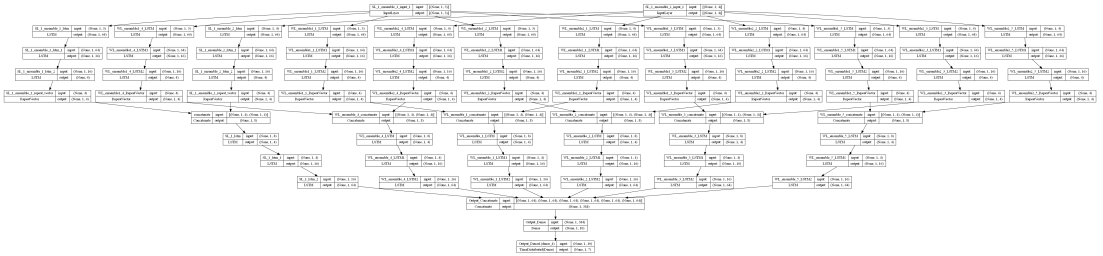


Figure A.2: Network Configuration for Five Axis Anomaly Detection with Ensemble Incremental Learning

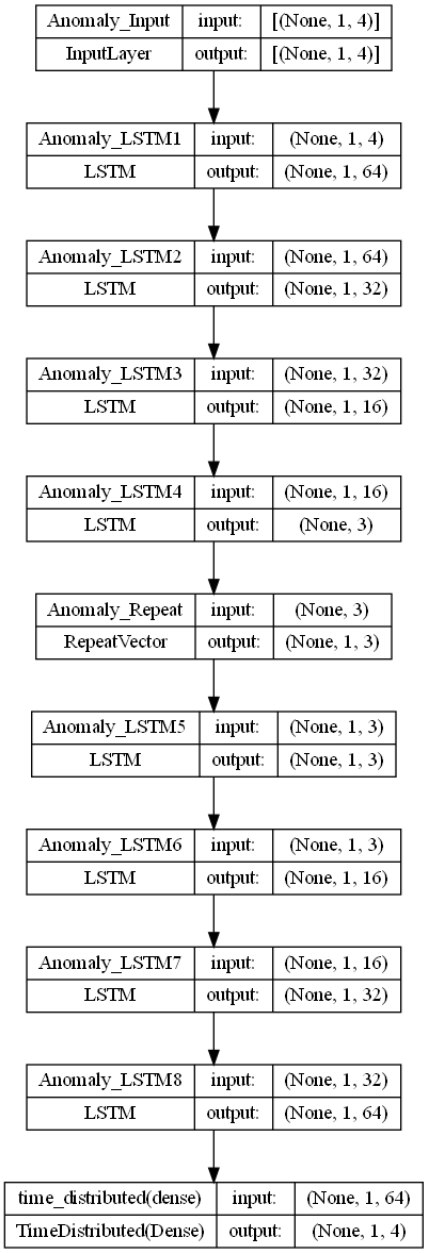


Figure A.3: Network architecture for dimensionality reduction of multi-class network