



University of HUDDERSFIELD

University of Huddersfield Repository

Alegh, Nurudeen Eshiemoghie

Condition Monitoring of Machine Tool Ball Screw Feed Drives Through Signal Analysis and Artificial Intelligence

Original Citation

Alegh, Nurudeen Eshiemoghie (2021) Condition Monitoring of Machine Tool Ball Screw Feed Drives Through Signal Analysis and Artificial Intelligence. Doctoral thesis, University of Huddersfield.

This version is available at <http://eprints.hud.ac.uk/id/eprint/35724/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

THE UNIVERSITY OF HUDDERSFIELD

Condition Monitoring of Machine Tool Ball Screw Feed Drives Through Signal Analysis and Artificial Intelligence

by

NURUDEEN ESHIEMOGHIE ALEGEH

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

in the

School of Computing and Engineering

University of Huddersfield

October 2021

Copyright Statement

- i. The author of this thesis (including any appendices and/ or schedules to this thesis) owns any copyright in it (the “Copyright”) and s/he has given The University of Huddersfield the right to use such Copyright for any administrative, promotional, educational and/or teaching.
- ii. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the University Details of these regulations may be obtained from the Librarian. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.
- iii. The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the “Intellectual Property Rights”) and any reproductions of copyright works, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Acknowledgement

I want to express my gratitude to everyone that was instrumental in the success of my PhD. These include my supervisory team (Prof Andrew Longstaff and Dr Simon Fletcher), the CPT support team, and my fellow researchers. I would also like to thank Machine Tool Technologies (MTT) Ltd. and the partners on the Innovate United Kingdom (UK) collaborative research project “Metrology and Digital Manufacturing for Servitisation of Manufacturing Machines (Grant Ref: 102787).

I would also like to acknowledge my wife (Grace), my parents (Mr and Mrs Y. Alegeh) and my sisters (Ilamosi, Avuedoya, Omoeghie and Munirah) for their emotional support and encouraging words.

Special thanks to Andrew Bell for his assistance during my experiments.

Abstract

This thesis is set in the context of the large volume of work directed to improving the overall equipment effectiveness (OEE) of manufacturing machines. Of the three OEE factors, performance receives much research attention since it provides simple metrics of parts per hour produced. However, availability and quality, which are the other two factors, can play an equally important role. In the past, high availability has been achieved by time-based or preventive maintenance techniques, which can be expensive and wasteful due to needless repairs and replacement of useful parts. This research aims to develop a cost-effective strategy for machine tool maintenance that improves availability and accuracy by adopting a condition-based or predictive maintenance approach.

The approaches under investigation use both machine learning and deep learning techniques to analyse continuous time-series signals to assess a machine tool's condition. For this research, the focus is on applying the techniques to the ball screw assembly of axis feed drives. This is one of the most common machine tool parts whose degradation can affect its availability and positional accuracy. This research data is obtained from experiments on a gantry-type machine tool with two ball screws, where one is good, and the other is worn.

In the machine learning approach, wavelet and fast Fourier transforms are employed for data processing on time-series vibration readings before extracting useful features for model training. These extracted features consistently show better accuracy across several machine learning algorithms than those obtained via classical methods. Deep learning is then investigated as an alternative method of analysing time-series data. The chosen approach utilises a pre-trained deep learning neural network based on convolution, which had been successfully used to learn from image files. The novelty in this research arises from the use of convolution-based deep learning on time series data. It does this by the conversion of the vibration signals to image files. The method of converting time-series data streams to images relevant for this analysis has been established and verified.

Test results show that the wavelet and fast Fourier transform (FFT) features used in the machine learning approach can outperform the statistical features in classifying the condition of the ball screw. With at least a 98 % accuracy across the examined machine learning networks compared to a range of 87 % (support vector machine) to 96 % (k nearest neighbour). On the other hand, the deep learning technique can achieve at least 98 % or 100 % accuracy when trained with raw and processed data, respectively. The deep learning approach has the advantage of requiring less data processing and better accuracy than the machine learning approach.

This research project will contribute to the manufacturing industry by improving the overall equipment effectiveness at a low cost. Furthermore, it can lead to real-time online condition monitoring with less overhead since there is no need for a data processing stage. This research's natural progression would be applying this approach to other parts of a machine tool or equipment. Furthermore, investigating and identifying specific faults and their progression would lead to a more sophisticated system for widespread deployment.

List of publications arising from this research

1. Alegeh, N. E., Shagluf, A., Longstaff, A. P. et al., "Accuracy in detecting failure in ballscrew assessment towards machine tool servitisation," 2019. International Journal of Mechanical Engineering and Robotics Research, 8 (5). pp. 667-673. ISSN 2278-0149 [1].

Parts of this paper contributes to chapters 3, 5 and 6 of this thesis.

2. Alegeh, N., Shagluf, A., Longstaff, A. et al., "Degradation monitoring of machine tool ballscrew using deep convolution neural network," 2020. Conference Proceedings, pp. 209-212. Presented in the 20th Euspen International Conference & Exhibition in Geneva, Switzerland held from 8th to 11th June [2].

Parts of this paper contributes to chapters 4, 5, 6 and 7 of this thesis. Specifically, Figure 6.13 is reproduced from this publication. The paper presented preliminary results comparing CNN on raw numeric data with DCNN on images visualising the data and showed the DCNN approach gave more promising results.

3. Alegeh, N., Thottoli, M., Mian, N. S. et al., "Feature extraction of time-series data using DWT and FFT for ball screw condition monitoring," 2021. Conference Proceedings. Presented at the 18th International Conference in Manufacturing Research in Derby, UK, held from 7th to 10th September [3].

Parts of this paper contributes to chapters 3, 5, 6 and 7 of this thesis. Specifically, Figure 6.4 is reproduced from this paper.

Table of Contents

Copyright Statement	ii
Acknowledgement	iii
Abstract	iv
List of publications arising from this research	v
Table of Contents	vi
List of Figures	xxiii
List of Tables	xxvii
Abbreviations and acronyms	xxviii
Chapter 1 Introduction	1
1.1 Background	1
1.2 Motivation	3
1.2.1 Servitisation of machine tools	3
1.2.2 Modes of failure for machine tool	5
1.2.3 Failure modelling techniques	6
1.2.3.1 Vibration analysis	6
1.2.3.2 Acoustic emission analysis	7
1.2.3.3 Temperature analysis	7
1.2.3.4 Oil analysis.....	8
1.2.3.5 Electrical signal analysis.....	8
1.2.4 Summary of motivation	9
1.3 Thesis structure	9
1.3.1 Chapter 1 Introduction.....	9

1.3.2 Chapter 2 Literature review	10
1.3.3 Chapter 3 Machine learning	10
1.3.4 Chapter 4 Deep learning	10
1.3.5 Chapter 5 Experiments and analysis.....	10
1.3.6 Chapter 6 Results and validations	11
1.3.7 Chapter 7 Conclusions and future work	11
1.4 Chapter summary	11
Chapter 2 Literature review	13
2.1 Machine tool maintenance and condition monitoring.....	13
2.1.1 Maintenance strategies	13
2.1.1.1 Operate-to-failure or reactive maintenance	14
2.1.1.2 Time-based or preventive maintenance	14
2.1.1.3 Condition-based or predictive maintenance.....	15
2.2 Vibration analysis.....	16
2.3 Ball screw.....	17
2.4 Ball screw faults.....	18
2.4.1 Fatigue	18
2.4.2 Cracks	18
2.4.3 Fretting	19
2.4.4 False brinelling	19
2.4.5 Corrosion	19
2.4.6 Wear.....	20
2.5 Artificial intelligence.....	20
2.5.1 White-box modelling.....	23

2.5.2 Black-box modelling	24
2.5.3 Grey-box modelling.....	24
2.6 Deep learning	25
2.7 Aim and objectives of the research	27
2.8 Scope	28
2.9 Novelty of the research and contribution to knowledge	29
2.9.1 Potential impact of the research on industry	30
2.10 Methodology	31
2.11 Chapter summary	32
Chapter 3 Machine learning	33
3.1 Signal pre-processing	33
3.1.1 Fast Fourier transform (FFT).....	34
3.1.2 Wavelet transforms.....	36
3.1.2.1 Continuous wavelet transforms.....	38
3.1.2.2 Discrete wavelets transform.....	39
3.2 Machine learning algorithms.....	40
3.2.1 Decision tree	41
3.2.1.1 Decision tree splitting rules	42
3.2.1.1.1 Information gain.....	43
3.2.1.1.2 Gain ratio	43
3.2.1.1.3 Gini index.....	44
3.2.1.2 Tree pruning.....	45
3.2.2 Support vector machine	47
3.2.3 K – nearest neighbour.....	52

3.2.4 Artificial neural network	54
3.3 Feature extraction.....	57
3.3.1 Types of extracted features.....	57
3.3.1.1 Mean	58
3.3.1.2 Standard deviation	58
3.3.1.3 Kurtosis.....	59
3.3.1.4 Skewness.....	60
3.3.1.5 Root mean square.....	60
3.3.1.6 Peak-to-peak value.....	61
3.3.2 Combination of features	61
3.3.3 Choice of signal processing for feature extraction	62
3.4 Chapter summary	64
Chapter 4 Deep learning	66
4.1 Deep learning overview	66
4.2 Structure of deep learning algorithms	68
4.2.1 Input layer.....	68
4.2.2 Output layer	68
4.2.3 Hidden layer	69
4.2.3.1 Convolution layer.....	69
4.2.3.2 Pooling layer	70
4.2.3.3 Sequence layer	71
4.2.3.4 Dropout layer	71
4.2.3.5 Normalisation layer.....	72
4.2.3.6 Activation layer.....	73

4.3 Activation function.....	73
4.3.1 Step function.....	73
4.3.2 Sigmoid function	74
4.3.3 Hyperbolic function.....	74
4.3.4 ReLU function	75
4.4 Deep learning models.....	76
4.4.1 Recurrent neural network	76
4.4.2 Deep-belief Network	79
4.4.3 Convolution neural network	80
4.5 The deep convolution neural network algorithm	84
4.6 Chapter summary	87
Chapter 5 Experiments and analysis of the dataset	88
5.1 Design of experiment	88
5.1.1 Pre-test.....	88
5.1.1.1 CNC machine.....	88
5.1.1.2 Accelerometer	90
5.1.1.3 Acoustic emission sensor	91
5.1.1.4 Data acquisition kit	93
5.1.1.5 Laboratory Virtual Instrumentation Engineering Workbench.....	94
5.1.1.6 Temperature sensor.....	95
5.1.1.7 Personal computer.....	97
5.1.2 Test setup.....	97
5.1.3 Test procedure	99
5.2 Description of conditions for each dataset	100

5.2.1 Baseline test dataset.....	101
5.2.2 Misaligned sensor axis test dataset.....	103
5.2.3 Thermal perturbation test datasets.....	103
5.2.4 Non-centuple feed rates test dataset	104
5.2.5 AE sensor test dataset.....	105
5.3 Analysis of datasets.....	106
5.3.1 Temperature sensor data.....	106
5.3.2 Accelerometer data.....	106
5.3.3 AE sensor data.....	111
5.4 Chapter summary	114
Chapter 6 Results and validations.....	115
6.1 Selection of optimal wavelet.....	115
6.2 Validation of wavelet and FFT features.....	120
6.3 Application of machine learning.....	123
6.4 Application of the DCNN	125
6.4.1 Data processing for DCNN.....	126
6.4.2 Results from training with DCNN.....	130
6.5 Validation of DCNN	134
6.5.1 Comparing the different datasets.....	134
6.5.2 Comparing the effect of fixed axes and grids.....	135
6.6 Chapter summary	136
Chapter 7 Conclusions and future work.....	138
7.1 Summary of the research work.....	138

7.2 Conclusions	139
7.3 Contribution to knowledge.....	142
7.4 Limitations of research work.....	143
7.5 Further work.....	144
References	146
Appendices.....	158
Appendix A: MATLAB code.....	158
A1: DCNN code _ improved version	158
A2: DCNN code _ original version	159
A3: Helper function for DCNN	159
Appendix B: DCNN results for misaligned sensor axis test	160
B1: Training progress for 9:1 ratio on training and test set.....	160
B2: Confusion matrix for 9:1 ratio on training and test set.....	160
B3: Training progress for 8.5:1.5 ratio on training and test set.....	161
B4: Confusion matrix for 8.5:1.5 ratio on training and test set.....	161
B5: Training progress for 4:1 ratio on training and test set.....	162
B6: Confusion matrix for 4:1 ratio on training and test set.....	162
B7: Training progress for 7:3 ratio on training and test set.....	163
B8: Confusion matrix for 7:3 ratio on training and test set.....	163
B9: Training progress for 3:2 ratio on training and test set.....	164
B10: Confusion matrix for 3:2 ratio on training and test set.....	164
Appendix C: DCNN results for 5 °C thermal perturbation test	165
C1: Training progress for 9:1 ratio on training and test set.....	165
C2: Confusion matrix for 9:1 ratio on training and test set.....	165

C3: Training progress for 8.5:1.5 ratio on training and test set.....	166
C4: Confusion matrix for 8.5:1.5 ratio on training and test set.....	166
C5: Training progress for 4:1 ratio on training and test set.....	167
C6: Confusion matrix for 4:1 ratio on training and test set.....	167
C7: Training progress for 7:1 ratio on training and test set.....	168
C8: Confusion matrix for 7:1 ratio on training and test set.....	168
C9: Training progress for 3:2 ratio on training and test set.....	169
C10: Confusion matrix for 3:2 ratio on training and test set.....	169
Appendix D: DCNN results for 10 °C thermal perturbation test.....	170
D1: Training progress for 9:1 ratio on training and test set	170
D2: Confusion matrix for 9:1 ratio on training and test set.....	170
D3: Training progress for 8.5:1.5 ratio on training and test set.....	171
D4: Confusion matrix for 8.5:1.5 ratio on training and test set.....	171
D5: Training progress for 4:1 ratio on training and test set	172
D6: Confusion matrix for 4:1 ratio on training and test set.....	172
D7: Training progress for 7:3 ratio on training and test set	173
D8: Confusion matrix for 7:3 ratio on training and test set.....	173
D9: Training progress for 3:2 ratio on training and test set	174
D10: Confusion matrix for 3:2 ratio on training and test set.....	174
Appendix E: DCNN results for 15 °C thermal perturbation test	175
E1: Training progress for 9:1 ratio on training and test set.....	175
E2: Confusion matrix for 9:1 ratio on training and test set	175
E3: Training progress for 8.5:1.5 ratio on training and test set.....	176
E4: Confusion matrix for 8.5:1.5 ratio on training and test set	176
E5: Training progress for 4:1 ratio on training and test set.....	177

E6: Confusion matrix for 4:1 ratio on training and test set	177
E7: Training progress for 7:3 ratio on training and test set.....	178
E8: Confusion matrix for 7:3 ratio on training and test set	178
E9: Training progress for 3:2 ratio on training and test set.....	179
E10: Confusion matrix for 3:2 ratio on training and test set	179
Appendix F: DCNN results for 20 °C thermal perturbation test.....	180
F1: Training progress for 9:1 ratio on training and test set	180
F2: Confusion matrix for 9:1 ratio on training and test set	180
F3: Training progress for 8.5:1.5 ratio on training and test set	181
F4: Confusion matrix for 8.5:1.5 ratio on training and test set	181
F5: Training progress for 4:1 ratio on training and test set	182
F6: Confusion matrix for 4:1 ratio on training and test set	182
F7: Training progress for 7:3 ratio on training and test set	183
F8: Confusion matrix for 7:3 ratio on training and test set	183
F9: Training progress for 3:2 ratio on training and test set	184
F10: Confusion matrix for 3:2 ratio on training and test set	184
Appendix G: DCNN results for 25 °C thermal perturbation test.....	185
G1: Training progress for 9:1 ratio on training and test set	185
G2: Confusion matrix for 9:1 ratio on training and test set.....	185
G3: Training progress for 8.5:1.5 ratio on training and test set.....	186
G4: Confusion matrix for 8.5:1.5 ratio on training and test set.....	186
G5: Training progress for 4:1 ratio on training and test set	187
G6: Confusion matrix for 4:1 ratio on training and test set.....	187
G7: Training progress for 7:3 ratio on training and test set	188
G8: Confusion matrix for 7:3 ratio on training and test set.....	188

G9: Training progress for 3:2 ratio on training and test set	189
G10: Confusion matrix for 3:2 ratio on training and test set.....	189
Appendix H: DCNN results for non-centuple feed rates test.....	190
H1: Training progress for 9:1 ratio on training and test set	190
H2: Confusion matrix for 9:1 ratio on training and test set.....	190
H3: Training progress for 8.5:1.5 ratio on training and test set.....	191
H4: Confusion matrix for 8.5:1.5 ratio on training and test set.....	191
H5: Training progress for 4:1 ratio on training and test set	192
H6: Confusion matrix for 4:1 ratio on training and test set.....	192
H7: Training progress for 7:3 ratio on training and test set	193
H8: Confusion matrix for 7:3 ratio on training and test set.....	193
H9: Training progress for 3:2 ratio on training and test set	194
H10: Confusion matrix for 3:2 ratio on training and test set.....	194
Appendix I: DCNN results for baseline test database.....	195
I1: Training progress for 9:1 ratio on training and test set	195
I2: Confusion matrix for 9:1 ratio on training and test set	195
I3: Training progress for 8.5:1.5 ratio on training and test set	196
I4: Confusion matrix for 8.5:1.5 ratio on training and test set	196
I5: Training progress for 4:1 ratio on training and test set	197
I6: Confusion matrix for 4:1 ratio on training and test set	197
I7: Training progress for 7:3 ratio on training and test set	198
I8: Confusion matrix for 7:3 ratio on training and test set	198
I9: Training progress for 3:2 ratio on training and test set	199
I10: Confusion matrix for 3:2 ratio on training and test set	199
Appendix J: Sample data for DCNN speed classification for baseline test	200

J1A: 1000 mm/min with fixed grid and axis	200
J1B: 1000 mm/min without grid and axis	200
J2A: 2500 mm/min with fixed grid and axis	201
J2B: 2500 mm/min without grid and axis	201
J3A: 3000 mm/min with fixed grid and axis	202
J3B: 3000 mm/min without grid and axis	202
J4A: 5000 mm/min with fixed grid and axis	203
J4B: 5000 mm/min without grid and axis	203
J5A: 7500 mm/min with fixed grid and axis	204
J5B: 7500 mm/min without grid and axis	204
J6A: 9000 mm/min with fixed grids and axis	205
J6B: 9000 mm/min without grids and axis	205
J7A: 10000 mm/min with fixed grids and axis	206
J7B: 10000 mm/min without grids and axis	206
Appendix K: DCNN results on speed classification on baseline test	207
K1A: Training progress for 9:1 ratio on training and test data (with grid/axis).....	207
K1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis).....	207
K1C: Training progress for 9:1 ratio on training and test data (no grid/axis).....	208
K1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis).....	208
K2A: Training progress for 4:1 ratio on training and test data (with grid/axis).....	209
K2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis).....	209
K2C: Training progress for 4:1 ratio on training and test data (no grid/axis).....	210
K2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis).....	210
K3A: Training progress for 7:3 ratio on training and test data (with grid/axis).....	211
K3B: Confusion matrix for 7:3 ratio on training and test data (with grid/axis).....	211

K3C: Training progress for 7:3 ratio on training and test data (no grid/axis).....	212
K3D: Confusion matrix for 7:3 ratio on training and test data (no grid/axis).....	212
K4A: Training progress for 3:2 ratio on training and test data (with grid/axis).....	213
K4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis).....	213
K4C: Training progress for 3:2 ratio on training and test data (no grid/axis).....	214
K4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis).....	214
K5: Comparing results between grid/axis and no grid/axis.....	215
Appendix L: DCNN results on speed classification on misaligned sensor axis test.....	215
L1A: Training progress for 9:1 ratio on training and test data (with grid/axis).....	215
L1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis).....	216
L1C: Training progress for 9:1 ratio on training and test data (no grid/axis).....	216
L1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis).....	217
L2A: Training progress for 4:1 ratio on training and test data (with grid/axis).....	217
L2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis).....	218
L2C: Training progress for 4:1 ratio on training and test data (no grid/axis).....	218
L2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis).....	219
L3A: Training progress for 7:3 ratio on training and test data (with grid/axis).....	219
L3B: Confusion matrix for 7:3 ratio on training and test data (with grid/axis).....	220
L3C: Training progress for 7:3 ratio on training and test data (no grid/axis).....	220
L3D: Confusion matrix for 7:3 ratio on training and test data (no grid/axis).....	221
L4A: Training progress for 3:2 ratio on training and test data (with grid/axis).....	221
L4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis).....	222
L4C: Training progress for 3:2 ratio on training and test data (no grid/axis).....	222
L4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis).....	223
L5: Comparing results between grids/axis and no grid/axis.....	223

Appendix M: DCNN results on speed classification on baseline test database	224
M1A: Training progress for 9:1 ratio on training and test data (with grid/axis)	224
M1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis)	224
M1C: Training progress for 9:1 ratio on training and test data (no grid/axis)	225
M1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis)	225
M2A: Training progress for 4:1 ratio on training and test data (with grid/axis)	226
M2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis)	226
M2C: Training progress for 4:1 ratio on training and test data (no grid/axis)	227
M2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis)	227
M3A: Training progress for 7:3 ratio on training and test data (with grid/axis)	228
M3B: Confusion matrix for 7:3 ratio on training and test data (with grid/axis)	228
M3C: Training progress for 7:3 ratio on training and test data (no grid/axis)	229
M3D: Confusion matrix for 7:3 ratio on training and test data (no grid/axis)	229
M4A: Training progress for 3:2 ratio on training and test data (with grid/axis)	230
M4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis)	230
M4C: Training progress for 3:2 ratio on training and test data (no grid/axis)	231
M4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis)	231
M5: Comparing results between grid/axis and no grid/axis	232
Appendix N: DCNN results on speed classification on 5 °C thermal perturbation test	232
N1A: Training progress for 9:1 ratio on training and test data (with grid/axis)	232
N1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis)	233
N1C: Training progress for 9:1 ratio on training and test data (no grid/axis)	233
N1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis)	234
N2A: Training progress for 4:1 ratio on training and test data (with grid/axis)	234
N2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis)	235

N2C: Training progress for 4:1 ratio on training and test data (no grid/axis).....	235
N2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis).....	236
N3A: Training progress for 7:1 ratio on training and test data (with grid/axis).....	236
N3B: Confusion matrix for 7:1 ratio on training and test data (with grid/axis).....	237
N3C: Training progress for 7:1 ratio on training and test data (no grid/axis).....	237
N3D: Confusion matrix for 7:1 ratio on training and test data (no grid/axis).....	238
N4A: Training progress for 3:2 ratio on training and test data (with grid/axis).....	238
N4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis).....	239
N4C: Training progress for 3:2 ratio on training and test data (no grid/axis).....	239
N4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis).....	240
N5: Comparing results between grid/axis and no grid/axis.....	240
Appendix O: Normalized FFT data for baseline test with fixed axis.....	241
O1A: Normalized FFT at 10000 mm/min for good ball screw	241
O1B: Normalized FFT at 10000 mm/min for bad ball screw.....	241
O2A: Normalized FFT at 9000 mm/min for good ball screw	242
O2B: Normalized FFT at 9000 mm/min for bad ball screw.....	242
O3A: Normalized FFT at 7500 mm/min for good ball screw	243
O3B: Normalized FFT at 7500 mm/min for bad ball screw.....	243
O4A: Normalized FFT at 5000 mm/min for good ball screw	244
O4B: Normalized FFT at 5000 mm/min for bad ball screw.....	244
O5A: Normalized FFT at 3000 mm/min for good ball screw	245
O5B: Normalized FFT at 3000 mm/min for bad ball screw.....	245
O6A: Normalized FFT at 2500 mm/min for good ball screw	246
O6B: Normalized FFT at 2500 mm/min for bad ball screw.....	246
O7A: Normalized FFT at 1000 mm/min for good ball screw	247

O7B: Normalized FFT at 1000 mm/min for bad ball screw.....	247
Appendix P: DCNN results for normalized FFT of baseline test	248
P1: Training progress for 9:1 ratio on training and test set	248
P2: Confusion matrix for 9:1 ratio on training and test set	249
P3: Training progress for 4:1 ratio on training and test set	249
P4: Confusion matrix for 4:1 ratio on training and test set	250
P5: Training progress for 7:3 ratio on training and test set	250
P6: Confusion matrix for 7:3 ratio on training and test set	251
P7: Training progress for 3:2 ratio on training and test set	251
P8: Confusion matrix for 3:2 ratio on training and test set	252
Appendix Q: DCNN results for normalized FFT of misaligned sensor axis test.....	252
Q1: Training progress for 9:1 ratio on training and test set	252
Q2: Confusion matrix for 9:1 ratio on training and test set.....	253
Q3: Training progress for 4:1 ratio on training and test set	253
Q4: Confusion matrix for 4:1 ratio on training and test set.....	254
Q5: Training progress for 7:3 ratio on training and test set	254
Q6: Confusion matrix for 7:3 ratio on training and test set.....	255
Q7: Training progress for 3:2 ratio on training and test set	255
Q8: Confusion matrix for 3:2 ratio on training and test set.....	256
Appendix R: DCNN results for normalized FFT of 5 °C thermal perturbation test	256
R1: Training progress for 9:1 ratio on training and test set.....	256
R2: Confusion matrix for 9:1 ratio on training and test set.....	257
R3: Training progress for 4:1 ratio on training and test set.....	257
R4: Confusion matrix for 4:1 ratio on training and test set.....	258
R5: Training progress for 7:3 ratio on training and test set.....	258

R6: Confusion matrix for 7:3 ratio on training and test set	259
R7: Training progress for 3:2 ratio on training and test set.....	259
R8: Confusion matrix for 3:2 ratio on training and test set.....	260
Appendix S: DCNN results for normalized FFT of 10 °C thermal perturbation test	260
S1: Training progress for 9:1 ratio on training and test set	260
S2: Confusion matrix for 9:1 ratio on training and test set	261
S3: Training progress for 4:1 ratio on training and test set	261
S4: Confusion matrix for 4:1 ratio on training and test set	262
S5: Training progress for 7:3 ratio on training and test set	262
S6: Confusion matrix for 7:3 ratio on training and test set	263
S7: Training progress for 3:2 ratio on training and test set	263
S8: Confusion matrix for 3:2 ratio on training and test set	264
Appendix T: DCNN results for normalized FFT of 15 °C thermal perturbation test	264
T1: Training progress for 9:1 ratio on training and test set.....	264
T2: Confusion matrix for 9:1 ratio on training and test set	265
T3: Training progress for 4:1 ratio on training and test set.....	265
T4: Confusion matrix for 4:1 ratio on training and test set	266
T5: Training progress for 7:3 ratio on training and test set.....	266
T6: Confusion matrix for 7:3 ratio on training and test set	267
T7: Training progress for 3:2 ratio on training and test set.....	267
T8: Confusion matrix for 3:2 ratio on training and test set	268
Appendix U: DCNN results for normalized FFT of 20 °C thermal perturbation test.....	268
U1: Training progress for 9:1 ratio on training and test set	268
U2: Confusion matrix for 9:1 ratio on training and test set.....	269
U3: Training progress for 4:1 ratio on training and test set	269

U4: Confusion matrix for 4:1 ratio on training and test set.....	270
U5: Training progress for 7:3 ratio on training and test set	270
U6: Confusion matrix for 7:3 ratio on training and test set.....	271
U7: Training progress for 3:2 ratio on training and test set	271
U8: Confusion matrix for 3:2 ratio on training and test set.....	272
Appendix V: Temperature chart for thermal perturbation test.....	272
V1: Temperature chart for 10 °C thermal perturbation test	272
V2: Temperature chart for 15 °C thermal perturbation test	273
V3: Temperature chart for 20 °C thermal perturbation test	273
V4: Temperature chart for 25 °C thermal perturbation test	274

List of Figures

Figure 2.1 Typical ball screw structure (Adapted from [63]).....	18
Figure 2.2 Diagrammatic depiction of modelling techniques.....	25
Figure 2.3 Machine learning versus Deep learning block diagram	27
Figure 3.1 Frequency vs time resolution for (a) DFT, (b) STFT, and (c) DWT [131].....	36
Figure 3.2 Wavelet transform breakdown of a signal [132]	40
Figure 3.3 Block diagram of a machine learning process.....	41
Figure 3.4 Decision tree structure	42
Figure 3.5 An unpruned decision tree.....	45
Figure 3.6 A pruned decision tree.....	46
Figure 3.7 SVM algorithm in an n-dimensional space	49
Figure 3.8 A maximal margin hyperplane with its support vectors.....	49
Figure 3.9 KNN algorithm.....	53
Figure 3.10 ANN structure	54
Figure 3.11 Neuron model structure	55
Figure 3.12 Training and validation error during training in ANN	56
Figure 3.13 Flow diagram of the proposed methodology.....	64
Figure 4.1 Artificial intelligence hierarchy.....	67
Figure 4.2 Block diagram of a general deep learning network.....	67
Figure 4.3 An illustration of the Convolution process [112].....	69
Figure 4.4 An illustration of the Pooling process [112].....	70
Figure 4.5 Example of before and after dropout [170]	71
Figure 4.6 Plot of Step activation function	74
Figure 4.7 Plot of Sigmoid activation function.....	74
Figure 4.8 Plot of Hyperbolic activation function	75

Figure 4.9 Plot of ReLU activation function	76
Figure 4.10 RNN structure [169]	77
Figure 4.11 An LSTM network with three gates [169]	78
Figure 4.12 Basic RBN structure [169]	79
Figure 4.13 DBN structure [172].	80
Figure 4.14 A typical CNN with two stacked set of convolution, activation, and pooling layers and two fully connected layers [169]	80
Figure 4.15 Convolution example with a stride of 2 [169]	81
Figure 4.16 Zero padding for convolution [169]	82
Figure 4.17 2x2 max-pooling without overlapping [169]	83
Figure 4.18 3x3 max pooling with overlapping [169]	83
Figure 4.19 DCNN architecture	85
Figure 5.1 The 5-axis gantry-type CNC machine used in this study	89
Figure 5.2 A CAD model of the 5-axis gantry-type CNC machine	90
Figure 5.3 Tri-axial accelerometer used for model calibration and testing	91
Figure 5.4 Acoustic emission sensor kit	92
Figure 5.5 AE sensor mounted on a ball screw assembly during a test	92
Figure 5.6 Data acquisition kit as wired for the experiment	93
Figure 5.7 Front panel of LabVIEW software	94
Figure 5.8 Three DS18B20 temperature sensors mounted on the CNC machine frame.	95
Figure 5.9 WinTcal interface	96
Figure 5.10 Accelerometer on ballnut with axes aligned with the machine axes (plan view)	98
Figure 5.11 Temperature profile of the ballnut for the entire duration of baseline test	102
Figure 5.12 Temperature of the ballnut after each feed rate during baseline test	102

Figure 5.13 Accelerometer on ballnut with sensor X- and Y- axes not aligned with the machine axes (Plan view).....	103
Figure 5.14 Temperature of the ballnut after each feed rate during the 5 °C thermal perturbation test.....	104
Figure 5.15 Temperature of the ballnut after each feed rate during the non-centuple feed rates test.....	105
Figure 5.16 Temperature of the ballnut after each feed rate during the AE sensor test	105
Figure 5.17 Accelerometer data of worn and good ball screw at 1000 and 10000 mm/min.	107
Figure 5.18 FFT of accelerometer data of worn/good ball screw at 1000 and 10000 mm/min	107
Figure 5.19 FFT of accelerometer data of worn and good ball screw with fixed axes.....	108
Figure 5.20 FFT of vibration signals at different feed rates for good ball screw.	109
Figure 5.21 FFT of vibration signals at different feed rates for worn ball screw	110
Figure 5.22 FFT of vibration signals at different feed rates for good ball screw with normalized frequency.....	111
Figure 5.23 AE sensor data of worn and good ball screw at 1000/10000 mm/min.....	112
Figure 5.24 FFT of AE sensor data of worn and good ball screw at 1000/10000 mm/min ..	113
Figure 5.25 FFT of AE sensor data of good ball screw at 10000 mm/min for the first 2 kHz frequency range.....	113
Figure 6.1 Bar chart of the accuracy of levels 1,2 and 3 detail coefficients of sym5	120
Figure 6.2 Wavelet and FFT features scatter plot of Kurtosis vs mean.....	121
Figure 6.3 Wavelet and FFT features scatter plot of Kurtosis of magnitude vs Average of magnitude.....	122
Figure 6.4 Bar chart of classification accuracy from multiple feature type.....	123
Figure 6.5 Comparison of results for kurtosis/skewness vs all features.....	125

Figure 6.6 Vibration data at 1000 mm/min feed rate with axes and grids.....	126
Figure 6.7 Vibration data at different feed rates with axes and grids	127
Figure 6.8 Sample plot of a vibration signal without axes and grids.....	127
Figure 6.9 Vibration signal with fixed axes and grids	128
Figure 6.10 Vibration signal without axes and grids	128
Figure 6.11 Vibration signal of all feed rates with fixed axes and grids	129
Figure 6.12 Vibration signal of all feed rates without fixed axes and grids	130
Figure 6.13 DCNN confusion matrix showing 94 % accuracy	131
Figure 6.14 Training progress for raw data	132
Figure 6.15 DCNN confusion matrix showing 98.8 % accuracy	132
Figure 6.16 Training progress for processed data.....	133
Figure 6.17 DCNN confusion matrix showing 100 % accuracy	133

List of Tables

Table 3.1 Wavelet families and abbreviations	37
Table 3.2 kernel functions.....	51
Table 3.3 Proximity metrics.....	53
Table 4.1 DCNN layers.....	84
Table 5.1 PC specifications.....	97
Table 5.2 Summary of experimental activities	101
Table 6.1 Wavelet validation results.....	116
Table 6.2 Best performing wavelet features	118
Table 6.3 Worst performing wavelet features.....	118
Table 6.4 Accuracy of level 1,2 and 3 detail coefficients of sym5.....	119
Table 6.5 Machine learning classification accuracy	120
Table 6.6 Classification accuracy from multiple feature type	123
Table 6.7 Machine learning algorithms accuracy for Kurtosis and skewness features	124
Table 6.8 DCNN results from several datasets to classify good and worn ball screw	134
Table 6.9 DCNN results to classify feed rates of different datasets	136

Abbreviations and acronyms

AE	Acoustic Emission
AI	Artificial Intelligence
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
BPTT	Backpropagation Through Time
CAD	Computer-Aided Design
CNC	Computer Numerical Control
CNN	Convolution Neural Network
ConvNet	Convolution Neural Network
CPT	Centre for Precision Technology
CR&D	Collaborative Research and Development
CWT	Continuous Wavelet Transforms
DBN	Deep-Belief Network
DCNN	Deep Convolution Neural Network
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
FMEA	Failure Mode Effect Analysis
ILSVRC	ImageNet Scale Visual Recognition Challenge
KNN	k-Nearest Neighbour
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
LSTM	Long Short-Term Memory
MATLAB	Matrix Laboratory
MTT	Machine Tool technologies

MTTF	Mean Time To Fail
MTTR	Mean Time To Repair
NI	National Instrument
NI-cDAQ	National Instrument compact Data Acquisition
OEE	Overall Equipment Effectiveness
PC	Personal Computer
RBM	Restricted Boltzmann Machine
RCT	Randomize Controlled Trial
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
RPN	Risk Priority Number
SGD	Stochastic Gradient Decent
SME	Small and Medium scale Enterprises
STFT	Short-Time Fourier Transform
SVM	Support Vector Machines
UK	United Kingdom
USB	Universal Serial Bus
VI	Virtual Instrument
Y1	Left hand (as viewed from the front) Y-axis on the test machine – Good
Y2	Right hand (as viewed from the front) Y-axis on the test machine - Worn

Chapter 1 Introduction

1.1 Background

The industrial need for this research was identified as a direct offspring of a collaborative research and development (CR&D) project¹ that focused on the “servitisation” of machine tools. Servitisation of machine tools is a business model where instead of a manufacturer procuring a machine tool as a capital asset, they procure the useful time available on the machine tool [4]. The concept of servitisation has been successfully applied in one form or another in different sectors of the economy by several companies, including Alstom (transport sector) and Rolls-Royce (aviation sector) [5]. For example, Rolls-Royce developed a servitisation model where the service buyers purchase a fixed warranty and operating charge based on an hourly rate for the effective running of jet engines [6].

The rationale for the project was that by adopting the business model on machine tools, manufacturers would be contractually provided with machines with high overall equipment effectiveness (OEE) [7] metrics of quality, performance, and availability. If the machines do not meet these metrics, then the service provider bears the cost of rectifying the machine. Machine tool users, who are not experts in machining strategies, thus relinquish the responsibility of maintaining the machine tool to the experts, who are machine tool builders or service companies. The model also enhances the ability of small and medium-scale enterprises to get access to expensive, cutting-edge machine tool technology since they do not need to commit to high up-front costs. Case in summary, the business model means that the manufacturer can focus all of their energy on the machine tool's effective and productive usage with the least possible downtime risk due to breakdown. This implies that the machine OEE is

¹ An innovate UK funded project – Metrology and Digital Manufacturing for Servitisation of Manufacturing Machines (grant number: 102787).

professionally and competently achieved through the machine tool experts, with advanced technological support, while the manufacturer accomplishes the production of high-value goods.

However, the machine tool service industry is currently established on contractual agreements that allow for scenarios where equipment operation and availability are achieved through preventative [8] or reactive [9] maintenance. In preventive maintenance, machines have parts replaced on a timed basis to avoid unplanned breakdowns. This can lead to waste since components with remaining lifespan can be taken out of service prematurely. In reactive maintenance strategy, machines are mended when components fail, which usually compromises availability and performance. Often, the machine tool has been running at reduced performance and lower accuracy for some time before a consequential failure occurs. This means that maintenance is delayed and is only undertaken when a breakdown occurs, which is inefficient and costly. Neither of these approaches is suitable for a servitisation service provider, who needs to guarantee performance without incurring the unnecessary cost of needless preventative actions.

The third maintenance paradigm which is often promoted in research articles is predictive maintenance [10]. With the enormous interest in the interconnectedness of systems offered by the “industry 4.0” initiative [11], there is significant interest in what condition and performance monitoring can be achieved within manufacturing [12]. Current automated "condition monitoring" schemes are mostly focused on what "can" be measured by applying available sensors in convenient locations in preference of what "should" be measured. This ultimately could lead to two problem scenarios that eliminate trust:

1. the machine can fail irrespective of the sensor output or;
2. the sensors indicate a problem, but the OEE is unaffected.

The servitisation project addressed this by applying the "condition monitoring" approach in the most logical method, from sensor selection to determining appropriate data processing. The level of monitoring is then tailored to the specific needs of the customer, and more specifically, the contract. Taking the sensor data in tandem with measurement information from the quality department, analysing it and using the results to determine the actions needed to maintain and improve the OEE would be the ultimate goal of the technical solution to support this business model. Since many components and subsystems of a CNC machine tool are not directly observable, the detail of the internal construction of the system is unknown. Therefore, there is the need to develop a system that is able to detect a fault even though the exact nature of the fault is not known. The ball screw drive system is one such example, where the geometry of the screw, internal working of the nut, etc. may not be known by the end-user. The machining spindle is similarly supplied as a sealed part of the machine, often without detailed drawings to indicate location and magnitude of important elements, such as the bearing arrangement. In each case, the cause of any damage from wear, impact, etc. may not have been monitored, and the effect is not deterministic. Therefore, the project is driven by manufacturers and machine suppliers' desire to improve their machine tools' overall control and longevity while not suffering from the loss of accuracy that would cause production losses and scrap. Manufacturing processes will become more efficient and flexible due to technical innovations in the manufacturers' mechanical conversion and capability development within the supply chain for machine tool products and services.

1.2 Motivation

1.2.1 Servitisation of machine tools

Servitisation has been described as an evolution from the sale of a product to the sale of a system that offers services for the use of its product [13, 14]. It is seen as an approach focused on the product by providing availability contracts and maintenance services with the product

[15, 16]. There is a continuous requirement in the manufacturing industry to improve production quality, performance, and machine tools availability [17]. This is due to the competitiveness of the manufacturing sector, customer desire, and increasing technological progress. A useful technique to achieve this is to allow experts with a good understanding of the machine tool characteristics to be responsible for their maintenance activities. This is the premise upon which the servitisation business plan is built. The responsibility to guarantee machine availability and high-performance integrity in this business plan is left to the machine tool manufacturer.

The servitisation model helps diversify the component manufacturer approach to machining, creating room for experimentation into novel means of advanced manufacturing technology to meet and perhaps even surpass customer requirements. This style of professional manufacturing support is ground-breaking for component manufacturers. It challenges the present state of affairs - which is based mainly on predictive maintenance to achieve machine availability and reliability. It could increase output, boost manufacturing sustainability across supply chains, and retain manufacturing technology in line with technological progress [18, 19]. Adopting servitisation would create a profound change in granting small and medium scale enterprises (SMEs) access to computer numerical control (CNC) machines with cutting-edge technology. This improves machine tool control and robustness, decreases losses and scraps in production, reduces carbon footprint, and increases overall energy consumption during production, which gives SMEs a chance to compete in the global market [13]. It also creates an environment that aids machine tool manufacturers to develop relationships with their customers and increases their profitability [20-22]. Although servitisation for manufacturing companies is an attractive strategy, it also has specific challenges or risks. These risks range from financial to technical in nature [23] and can be borne by both the service provider and the manufacturer. It can also be challenging to define the product-service parameters since services

are fuzzy by nature [24]. The definition of the product-service parameters determines how the problem of which party bears the burden is handled. While the benefits to the end-user have been explained, the problem that remains to be addressed is how to minimise the risk to the service provider since the customer is operating the machine in a remote site [25, 26] according to their manufacturing processes. The machine tool manufacturer needs to be able to establish if a machine tool malfunction or loss of accuracy is due to misuse or normal machine degradation to determine which party is responsible. More importantly, for a successful business model, is to predict any future loss of accuracy or availability, to minimise disruption by correcting future faults before they adversely affect the OEE. To do this requires putting in place an advanced technology to perform condition monitoring and predictive analytics on the machine tool operation.

1.2.2 Modes of failure for machine tool

Modes of failure for machine tools or indeed any device can be investigated using failure mode and effect analysis (FMEA). FMEA was originally developed and standardised in the late 1940s by the US military, and its procedure was described in the USA Department of Defence, MIL-P-1629 [27]. Although originally developed by the military, FMEA is applied in many other industries like automotive [28], health [27], and manufacturing [29, 30] as a reliability study tool. FMEA is a valuable methodology used in evaluating different levels of a system to assess risk, availability, and reliability to identify critical failure mode [31]. Applying this methodology to machine tools, components and subsystems with critical failure modes can be identified. Generally, FMEA combines qualitative (expert knowledge) and quantitative data (mean time to failure (MTTF) and mean time to repair (MTTR)) to develop a highly structured, systematic technique for failure mode analysis [32]. An FMEA can be carried out on a machine tool by adhering to the guidance detailed in the MIL-STD-1629A. The process is documented for a CNC turning centre by Patil and Kothavale [31]. The FMEA evaluates all components

and subsystems of a machine tool, including mechanical, electrical, and electronics. Patil and Kothavale [31] observed that even though the electrical and electronics subsystem has a higher failure probability than the mechanical subsystems, they are not classified as critical subsystems. The identified critical subsystems are mainly subsystems that are engaged in continuous motion (rotational, translational and linear) during the operation of the machine tool like the main transmission, axes, and spindle subsystems. The criticality of a component or subsystem is determined based on their risk priority number (RPN). An RPN is the product of the severity of failure ranking, likelihood of occurrence of failure ranking and, the likelihood of detection of failure ranking [31]. These rankings range from 1 to 10, hence the RPN will range from 1 to 1000. In term of criticality, the higher RPN score the more critical the subsystem or component.

1.2.3 Failure modelling techniques

There are many different methods and strategies for investigating failure in machine tools, some of which are discussed in the following subsections:

1.2.3.1 Vibration analysis

Vibration analysis is the most commonly used method of failure investigation technique, especially in commercially available condition monitoring systems [33, 34]. It is mainly used for diagnosing mechanical faults in moving and vibrating components like gearbox, ball screw, shaft, bearing, and rotor [35]. In order to be effective, the vibration sensor is placed as close as possible to the source of vibration or the faulty components. There are different types of vibration sensors like accelerometers, velocity sensors, position sensors and displacement sensors. Out of all the vibration sensors, the accelerometer is the most widely used because it has the widest frequency response range, typically 1 to 30 kHz [33]. In terms of application, vibration analysis has been effectively used with good results for the condition monitoring of machine components like gearbox [36], bearing [37], and rotor [38]. However, vibration

signals typically have noise content and are not particularly viable for detecting faults at their early stages [33]. Due to the fact that vibration sensors are required to be installed as close as possible to the source of vibration, access to the sensor could be a problem during machine operation, especially in bulky machines. Also, in rotor fault analysis, it is required to have detailed knowledge of the frequency response function at different sensor positions making the fault identification process quite complex [39].

1.2.3.2 Acoustic emission analysis

Acoustic emissions (AE) are ultrasonic sound waves emitted when metallic materials are altered or subjected to stress [40]. AE signals, unlike vibration signals, have a high signal-to-noise ratio, making them suitable for incipient fault detection. AE sensors have also been applied, just like vibration sensors, has been effectively used with good results for the condition monitoring of machine components like gearbox [41], bearing [42], and rotor [43]. However, this method typically requires the use of multiple AE sensors, with each sensor having a dedicated data acquisition system [33]. This makes the installation and setup to be tedious and unnecessarily complicated due to wiring complexity. It also makes the setup to be quite expensive since multiple data acquisition kits are required. This method is affected by external environmental noise from the surrounding, making it unsuitable for most practical applications. It may also suffer from signal attenuation, so the sensor must be close to the signal source [44].

1.2.3.3 Temperature analysis

Many components or subsystems of a machine (especially those that move or vibrate) typically would experience temperature rise during operation. This rise in temperature is expected to stay within a certain range for a component or subsystem in good condition. Any temperature deviation above this normal working range is an indication of a fault in that component or subsystem. Hence, temperature measurement can be used in this way for condition monitoring [33]. Temperature measurements can be collected by means of a thermal camera or by

installing temperature sensors. Temperature analysis has successfully been used for the condition monitoring of generators [45] and geared systems [46]. However, this approach may be easily prone to systematic error in condition monitoring of gear systems as there is no clear correlation between the gear system change in temperature and its fault state [46]. Also, it may be difficult to identify the actual source of temperature rise as the sensor might be detecting heat from another component or subsystem via heat transfer.

1.2.3.4 Oil analysis

Oil analysis is the process of analysing samples of the lubricating oil of a machine (typically off-line) for contamination, viscosity, moisture, pressure or temperature [34]. The particular characteristics of this parameter indicate the condition of the monitored component or subsystem of a machine. This method is effective in detecting early signs of a fault and faults from components with little or no moving parts. Oil analysis finds application in gearbox condition monitoring [47] and remaining useful life prediction with promising results [48]. However, since this method is mostly off-line, samples can be taken too far apart, which leads to potentially late detection of a fault. This situation can be overcome by performing the analysis on-line but this can be quite expensive as additional instrumentation and sensors is required [33].

1.2.3.5 Electrical signal analysis

Electrical signal analysis is the use of voltage, current and resistance analysis for the condition monitoring of electric equipment like motors, generators and accumulators. It is effective for the detection of mechanical and structural faults and also the early detection of electrical faults. This is demonstrated by Lu et al. [49] in the diagnosis of gear tooth in gearboxes and Gong et al. [50] in the early detection of bearing fault. This method is often easy and cheap to implement as the required signals are usually already built into the control system of the machine [33]. Hence, there might be a requirement for additional instrumentation, sensors, or data acquisition

kits. Also, this method is suited for online analysis and consequently the early detection of faults. However, it suffers from the disadvantage that it can only be used for electrical equipment.

1.2.4 Summary of motivation

This research is geared toward providing a technique that improves the accuracy and availability of a machine tool using intelligent condition monitoring. The technique can be used to predict catastrophic failure by analysing input data from a machine using artificial intelligence. The analysis of existing FMEA for Machine Tool use (section 1.2.2) led to the identification of the ball screw assembly as a critical subsystem and hence the concentration on the ball screw feed drive as a major cause of failure. A significant cause of concern for a servitisation company would be how to model and predict the failures on a wide variety of machines without onerous testing regimes on every ball screw.

Although the work in this thesis was motivated by a larger project working towards “servitisation” of machine tools, this is just a more stringent requirement for remote, automated monitoring since the servitising company does not have direct access to the machine assets during everyday operation. The techniques derived in this work would be equally applicable for companies undertaking predictive maintenance of their machines.

1.3 Thesis structure

The thesis structure is arranged into chapters, from chapter one to chapter seven, plus abstract, references, and appendices. A synopsis of all the chapters is given here. Detailed information will be found in the specific chapters.

1.3.1 Chapter 1 Introduction

This chapter introduces the PhD research work by giving its background and general overview. It also establishes the main motivation of the PhD project and provides details of the thesis

structure, including chapters preview. The chapter is concluded with a summary of the content discussed in chapter one.

1.3.2 Chapter 2 Literature review

This chapter presents a review of literature relating to condition monitoring, artificial intelligence and deep learning. Topics such as machine tool maintenance strategies, modelling techniques and the different types of deep learning techniques were discussed. It also establishes the specific aim, objectives and scope of the PhD project. An overview of the contribution of this research to industry is highlighted. This is followed by the research methodology, which details the processes undertaken to achieve the final results and conclusions. The chapter is concluded with a summary of the content discussed in chapter two.

1.3.3 Chapter 3 Machine learning

This chapter analyses feature extraction for machine learning and the process of achieving it as utilised in this research. It focuses on fast Fourier transform (FFT) and wavelet transform-based signal processing for feature extraction. It includes details of machine learning algorithms like decision tree, support vector machine, K – nearest neighbour, and artificial neural network. This is followed by a presentation of the proposed algorithm that is based on machine learning. The chapter is concluded with a summary of the content discussed in chapter three.

1.3.4 Chapter 4 Deep learning

This chapter presents the proposed deep learning algorithm that will be used in this thesis. It focuses on its structure, types of activation functions, models, inputs, hidden layer and output. The chapter is concluded with a summary of the content discussed in chapter four.

1.3.5 Chapter 5 Experiments and analysis

This chapter details the design of experiment, which includes the pre-test activities, test setup, and the test procedure. It also introduces the datasets used for this research and details the

process of applying the proposed algorithms of chapters four and five on the datasets. An analysis of the collected datasets from the experiment was highlighted. The chapter is concluded with a summary of the content discussed in chapter five.

1.3.6 Chapter 6 Results and validations

In this chapter, the results of the analysis conducted are presented. The validation of the algorithms and models is also made, which includes the validation of the wavelet transform, the extracted feature for machine learning and the deep learning model used in this research. The chapter is concluded with a summary of the content discussed in chapter six.

1.3.7 Chapter 7 Conclusions and future work

This chapter summarises the research work in this thesis. It also details the conclusion derived in order of importance, the contributions of this research to knowledge, and then presents further work that are direct offshoots of this project.

1.4 Chapter summary

This research is as a result of the need to achieve machine tool servitisation. Servitisation of machine tools is a business model where a manufacturer procures the useful time available on the machine tool instead of the capital assets. The implication of this is that highly skilled personnel with the right resources are responsible for the machine tool maintenance. This will lead to improved productivity and increased accuracy in production and granting SMEs access to CNC machines with cutting-edge technology. This should improve machine tool control and robustness, decreases losses and scraps in production, reduces carbon footprint, and increases overall energy consumption efficiency. However, since the machine will be used in a remote location, the problem arises of how to effectively monitor the machine tool such that availability and accuracy are guaranteed. That is the problem that this research aims to address by using vibration analysis and artificial intelligence (AI) to monitor the condition of ball

screws used in CNC machine tool feed drive systems. The next chapter delves into the literature review, aim, objectives, scope and methodology of the PhD research.

Chapter 2 Literature review

This chapter presents the review and critical appraisal of literature of methods of condition monitoring applied to machine tools and other similar systems where the techniques could be transferable. It starts by providing an overview of the different techniques used for investigating condition monitoring. This is followed by a discussion of the application of artificial intelligence (AI) techniques to obtaining an accurate predictive model. This review synthesises the important ideas, claims, arguments, and conclusions mentioned in previous research to identify the important issues and establish a theoretical basis for the proposed prediction model. Materials used for this review include journals, conference papers, and textbooks.

2.1 Machine tool maintenance and condition monitoring

2.1.1 Maintenance strategies

Machine tools are designed to operate under many different conditions. However, within these operating conditions, it is desired that they perform optimally. Optimal performance implies that the machine tools perform as per their designed and installed capacity. However, due to heat, misalignment, and wear and tear, this is not the case. Herein comes the subject of machine tool maintenance strategy. Machine tool maintenance strategies are designed to optimise the accuracy and availability of a machine tool. The type of maintenance strategy adopted usually depends on cost and company size. There are mainly three machine maintenance strategies used in the manufacturing industry [51, 52]. They are namely:

1. Operate-to-failure or reactive maintenance strategy.
2. Time-based or preventive maintenance strategy.
3. Condition-based or predictive maintenance strategy.

2.1.1.1 Operate-to-failure or reactive maintenance

Operate-to-failure or reactive maintenance (also known as corrective maintenance) strategy does not take any preliminary actions to detect the initiation or prevention of failure. As long as the machine tool remains operational and functioning, all repair activities are put on hold. The need for maintenance or repair work will usually occur without warning. This type of maintenance strategy has the following characteristics;

1. It has a high mean time to fail (MTTF).
2. In general, the failure is catastrophic.
3. It has a high cost of maintenance
4. It is usually accompanied by a lengthy period of machine unavailability.
5. It results in revenue loss and production time.
6. It reduces the waste of spare parts due to preventive maintenance activities.
7. It can only be beneficial for machines; (a) that are low-cost and not critical to production, (b) whose effect of failure is negligible in terms of production losses, (c) whose cost of repairing after failure is considerably less than that required in the case of implementing a different maintenance strategy, (d) whose failure is severe and critical but the consequences are not immediate hence required repairs can be effected in this time.

2.1.1.2 Time-based or preventive maintenance

Time-based or preventive maintenance strategy, as the names imply, implements a schedule of maintenance activities fixed at regular intervals based on manufacturers' recommendations and past experiences. The idea is to replace, maintain or service a machine tool before it fails. This type of maintenance strategy has the following characteristics;

1. It allows for the scheduling of maintenance actions in advance.

2. It can result in the needless repair and replacement of parts that still have a long useful life.
3. It can result in wastage of spare parts, man-hours, manpower, and revenue.
4. It does not guarantee availability as certain failures can still occur between maintenance activities.
5. It can reduce the occurrence of critical failure.
6. It is highly beneficial in situations where the MTTF can be estimated reasonably accurately.

2.1.1.3 Condition-based or predictive maintenance

Condition-based or predictive maintenance is defined by Davies [53] as "the performance of periodic or continuous comparative measurement on parameters which are suspected of reflecting the condition of a component, sub-assembly or system with the object that on analysis, the measurements may indicate the items current condition and the future trend of its possible deterioration." It guarantees that all decisions are based on reliable, independently verified diagnostic data. Thus, presenting a foundation for cost-effective and coherent decision-making by assisting in the planning of upcoming maintenance activities. Sensors and transducers are used to collect data from the machine, which is then analysed and interpreted using suitable technology to determine the machine's condition. This type of maintenance strategy has the following characteristics;

1. It requires a high initial setup cost.
2. It enables maintenance activities to be performed at the optimum time
3. It leads to efficient use of resources: spare parts, man-hours, manpower, and revenue.
4. It helps to prevent unnecessary machine tool downtime.
5. It helps reduce the cost of machine tool maintenance.
6. It increases machine tool useful life.

7. Implementation on smaller machines could be economically disadvantageous.
8. It is beneficial for large machines that are critical to production.
9. It requires a reliable and consistent condition monitoring technique and tools.

2.2 Vibration analysis

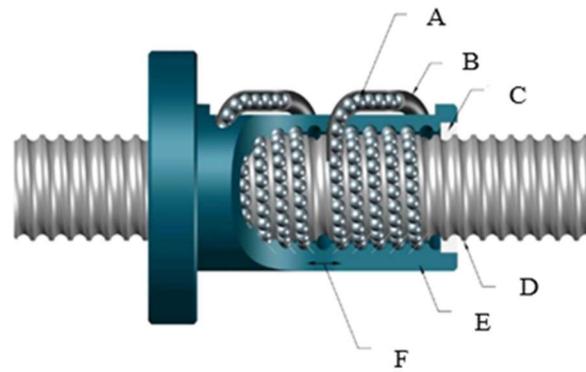
Vibration analysis is concerned with dynamic events like force and displacement as functions of time. All machines, even in good condition, are characterised by these dynamic events as a result of the movements of their parts. These events can be referred to as the vibration signature, which changes depending on the machine's working state. This may include a change in speed, load, cutting force, and nature of the workpiece. Hence a faulty machine will exhibit a vibration signature, which can be related to the fault. Likewise, a non-faulty machine will display a vibration signature, which can be related to its non-fault state [54]. The change in the vibration signature is due to the changes in the amplitude and frequency of the generated signal (wave) from cracks and other forms of permanent deformations as the machine operates. In order to measure vibration, sensors measure physical parameters in which vibration can be expressed, such as displacement, velocity, and acceleration [54]. Typical instruments for measuring vibration signals include proximity probes, velocity transducers, accelerometers, dual vibration probes and laser vibrometers. Some of the advantages of vibration analysis that make it suitable for machine tool condition monitoring include [54]

1. Its reaction time is instant in terms of response to machine component structural and physical change.
2. It easily isolates the source component of the fault.
3. It can easily be adapted for real-time online condition monitoring
4. It can be used with signal processing tools to detect weak fault signals before they become more prominent.

A secondary effect of vibration is acoustic emission (AE), which are high-frequency solid-borne vibration signals. These signals are collected by placing the AE sensor on the metallic component's surface closest to the source of the vibration. Hence the reason why many researchers are equally focused on investigating acoustic emission [55-57].

2.3 Ball screw

There are several methods of realising translation motion of the spindle of a CNC machine, and they include ball screw and ballnut, rack and pinion, leadscrew, and belt drive. Nevertheless, most high-precision machine tools utilize ball screw and ballnut drive systems. This is due to its high accuracy and efficiency in converting rotary motion to linear motion, resulting in good dynamic performance and minimal backlash [58]. This makes them well suited for applications that require accurate positioning and levelling, like in aeroplanes [59]. They are also widely used in applications that require a high number of cycles due to their high mechanical endurance capability [60]. However, ball screws can wear out their raceways and ball bearings (especially if poorly or not lubricated), generating heat as a result of contact between metal surfaces which causes geometric and thermal deformation of the machine tool. If not corrected or compensated for, these deformations will have a negative impact on the precision of the machine tool [61]. Ball screw degradation is often caused by wear from unbalanced forces, friction, or misalignment during installation. In recent times, most condition monitoring applications for the detection of degradation of ball screws are based on vibration or Acoustic Emission (AE) characteristics. This is because there is a significant correlation between vibration and noise levels as it relates to increasing bearing degradation [62]. Figure 2.1 shows a typical ball screw structure and its ballnut.



- A: Balls;
- B: Ball Return Tube;
- C: Wiper;
- D: Ballscrew;
- E: Ballnut;
- F: Integral Lead Shaft

Figure 2.1 Typical ball screw structure (Adapted from [63])

2.4 Ball screw faults

Ball screws can develop certain faults or failures due to material, installation, usage and maintenance [64]. Some of the faults of ball screws include:

2.4.1 Fatigue

This refers to the weakened condition in the material makeup of the ball screw exhibited as cracks due to the repeated stress. Fatigue in ball screw drive systems is load and cycle-dependent and is caused by many different factors like misalignment, overtight fittings, ovality, axial thrust, and overloading [65, 66]. Fatigue in ball screws can be prevented by lubrication, routine maintenance, and the use of materials with high fracture toughness.

2.4.2 Cracks

These are long narrow openings or fissures that develop in the ball screw material. Ball screw cracks are caused by overloading, vibrations, misalignment, and substandard materials [64]. Ball screw cracks can be prevented by proper installation of ball screw, use of seals to damp vibrations, use of standard materials, and routine maintenance.

2.4.3 Fretting

Fretting is caused by the rubbing contacts in oscillatory motion under load [64]. The resulting surface degradation is accompanied by the transfer of material and then the oxidation of the exposed surface and the metallic debris. This oxidation forms much harder deposits than the original material that was oxidised, which then acts as an abrasive agent, causing more fretting. Fretting can be overcome by lubrication, using material combination with good friction behaviour (less likely to fret when rubbed together), and using seals to damp vibrations and prevent access to oxygen.

2.4.4 False brinelling

False brinelling is a type of surface degradation that is similar to fretting. It occurs when the stationary ballnuts of a ball screw drive system is induced to oscillate or vibrate by an external force [67]. This oscillation causes the ballnuts to rub against each other and other contact surfaces, creating wear and debris. This debris, just like in the case of fretting, gets oxidised and form deposits that are much harder than the original material that was oxidised, which then acts as an abrasive agent, causing more false brinelling. False brinelling can be combated overcome by proper lubrication, routine maintenance, and the use of seals to damp vibrations and prevent access to oxygen.

2.4.5 Corrosion

This usually occurs in storage for new ball screws or during an extended period of inactivity for ball screws already in service. It is caused by water or moisture getting into the ballnuts of the ball screw and deteriorating the ball screw material by chemically reacting with it [68]. Corrosion can be prevented by proper storage, lubrication, routine maintenance, and putting a protective cover on an in-service ball screw to protect against moisture.

2.4.6 Wear

Wear in ball screws is due to misalignment and lack of or inadequate lubrication. Wear is characterised by deformation and surface material removal caused by frictional forces when multiple surfaces rub against one another [66]. This may result in vibration when moving, which could cause the ball screw to bend. These vibrations could initiate fretting and consequently false brinelling resulting in even more damage to the ball screw. To avoid wear in ball screws, it is important to ensure that there is no misalignment during installation. Other ways to prevent wear in ball screws include lubrication, routine maintenance, and proper protective covering to prevent dirt from entering the ball screw.

2.5 Artificial intelligence

Artificial intelligence (AI) continues to grow in usage among researchers as it meets their need for accuracy and practicality. It has already been applied to many different fields like in the financial sector for risk analysis [69], in the tourism industry to forecast arrivals [70], in the medical industry for medical imaging applications [71], in the agriculture industry for digital mapping of carbon fractions in soil [72], in the music industry for predicting hit songs [73], in international relations for detecting provocation patterns in the military [74], in engineering for condition monitoring [75] and many more. In all of the above research, the solution does not depend upon an expert agent in the final decision making in the process. The proposed model in all of these cases seeks to deskill the classical approach, which is heavily reliant on an expert agent. This approach is particularly advantageous for situation where there is a lot of uncertainty about the physical and mechanical make-up of the concerned system or when the system experiences poorly defined perturbations. The literature therefore indicates that such AI techniques would be suitable in the many situations where mechanical systems and subsystem are sealed by the manufacturer. It also reduces the influence of personal biases and misconception that could influence the output of the model. AI generally refers to machine

learning and deep learning. One of the many AI methods already being applied to condition monitoring is artificial neural network (ANN). Ozel and Karpat [76] used ANN for tool wear condition monitoring and surface roughness prediction for varied cutting conditions. The developed model has the capacity to describe more complicated non-linearities and interactions especially compared to regression models (linear and exponential), which make its predictive performance better. Others include support vector machines (SVM), Lim et al. [77] used SVM analysis on thermal image data for fault diagnosis in rotating machines. This method provides an alternative contactless approach to condition monitoring of machine components or subsystems to cases where it may be impossible or harmful to implement a direct contact approach. This research shows that this method achieves good accuracy of up to 96 % and compares favourably with vibration analysis, especially in detecting incipient faults.

Muralidharan et al. [78] establish a decision tree methodology for the condition monitoring of a self-aligning carrying idler. The self-aligning carrying idler is a critical subsystem of a belt conveyor that consists of the bearing, shaft, labyrinth seal, and outer roller. The self-aligning carrying idler was analysed under five different conditions, this includes: good, bearing fault, shaft fault, labyrinth fault and outer roller fault. The developed method used eleven features to train the decision tree with kurtosis as the root feature and achieved an accuracy of 99.52 %. However, this strategy is highly prone to over-fitting, which means that the proposed model is likely not to perform as well when presented with a new dataset.

Abdulshahed [79], in his research, used ANFIS for the compensation of thermal errors on CNC machine tools. Two predictive models were designed based on grid and c-means clustering methods, and grey theory was used for optimal temperature sensor selection. This approach was able to develop an ANFIS and c-means clustering model with superior predictive ability but with fewer rules by achieving 94 % improvement in thermal error compensation. However, the temperature data used to train the network was instantaneous rather than continuous. Hence,

the obtained results were for a particular point in time with no influence from the past temperature data.

Gunerkar and Jalan [80] developed a novel approach that combines vibration and acoustic emission data for fault diagnosis of bearings. This combines the advantages offered by vibration and acoustic emission sensors to achieve improved results compared to when either sensor is used. The process involves multiple data acquisition, signal analysis, feature extraction, feature classification with KNN, high-level data fusion, and domain-specific expert decision-making knowledge. The result obtained indicates that the proposed method outperforms situations where vibration or acoustic emission sensors are used. However, this method may be more prone to systematic error as the level of expert knowledge required is high.

These AI methods offer good accuracy, real-time analysis and generally outperform classical methods of condition monitoring with the added advantage of reduce expert agent influence on the final decision making of the models.

Creating reliable models is a major challenge in the engineering field (as well as in any other field) due to the problem of causation, correlation, and coincidence. All models aim to relate an effect to a cause; this will make the model accurate. A model that unwittingly relates an effect to a correlation or a coincidence would be inaccurate. In some cases, it might be evident that an effect is attributed to a correlation or coincidence, and in others, it might not. For example,

1. The crow of a rooster causes daybreak
2. Citrus fruit cures scurvy

In example 1, we know that the typical sequence of dawn, morning, afternoon, evening, and night will still occur even if the rooster does not crow for some reason. There are also situations where roosters crow during the day. So, relating the rooster crow to daybreak is clearly a case

of correlation or coincidence at best. In the second example, though, it is not so clear that the assertion that citrus fruit cures scurvy is a correlation or a coincidence. Based on the observation that scurvy patients taking citrus fruit are cured tends to point to a causal relationship. However, from more research and randomised controlled trial (RCT), it was established that ascorbic acid or vitamin C (a constituent of citrus fruit) is the cure for scurvy. Hence example 2 shows a correlation, not a cause. The correlation and coincidence problem can be significantly minimised and ultimately eliminated by performing RCTs and removing confounders from the experimentation process.

Another modelling challenge is the non-linear nature of real-world problems, making it problematic to represent with an adequate mathematical or statistical model. The best scientists and engineers can do is create a model that comes as close as possible to characterising the problem, making the analysis somewhat subjective as opposed to being truly objective. There are typically three broad modelling methods (1) white-box, (2) black-box, and (3) grey-box.

2.5.1 White-box modelling

In white-box modelling, mathematical equations representing the analogous physical law are used to characterise the problem in question [81]. In formulating these equations, certain assumptions are made based on experience, knowledge, and logic. The solution to these equations, using analytic or numerical analysis, gives a realistic and tangible representation of the problem. Hence this model is more clearly open for understanding and clarification from an observer and a proposer point of view. Nevertheless, this modelling approach suffers from problems of accuracy. This could be as a result of; (a) system complexity, (b) the analogous physical law does not adequately correspond to the problem, (c) the assumptions made either oversimplifies the system, which changes the system characteristics or is wrong for the system. Models in mathematics, statistics, and physics mainly employ this approach, and some examples are regression analysis, correlation analysis, and fuzzy logic.

2.5.2 Black-box modelling

Black-box modelling is a data-driven modelling process [82, 83]. This implies that the model parameters are defined by or dependant on data collection and analysis. This modelling method relies on the principle of cause and effect to establish a dependence between inputs and outputs. The main advantage of this approach is that it does not require expert knowledge of the physical process to be modelled. It also provides better accuracy in contrast with the white-box method [76, 84]. However, it is susceptible to the problem of correlation and coincidence, which cannot be tested for in any given data. It is also not open for clear understanding and clarification from an observer and a proposer's point of view. It can be noted that the accuracy of this approach, by its nature, is based heavily on the accuracy of the data used. Any error associated with the sensor, data collection process, experiment procedure, and confounders will be accounted for by the model. This modelling approach can be seen in artificial neural networks (ANN) and deep learning.

2.5.3 Grey-box modelling

Grey-box modelling seeks to positively combine certain white-box and black-box methods [85-87]. This implies that it combines the advantages of the white-box and the black-box approach. The Adaptive Neuro-Fuzzy Inference System (ANFIS), which is a combination of ANN (black-box modelling) and fuzzy logic (white-box modelling), are examples of a grey-box modelling method. The three modelling techniques are illustrated in Figure 2.2

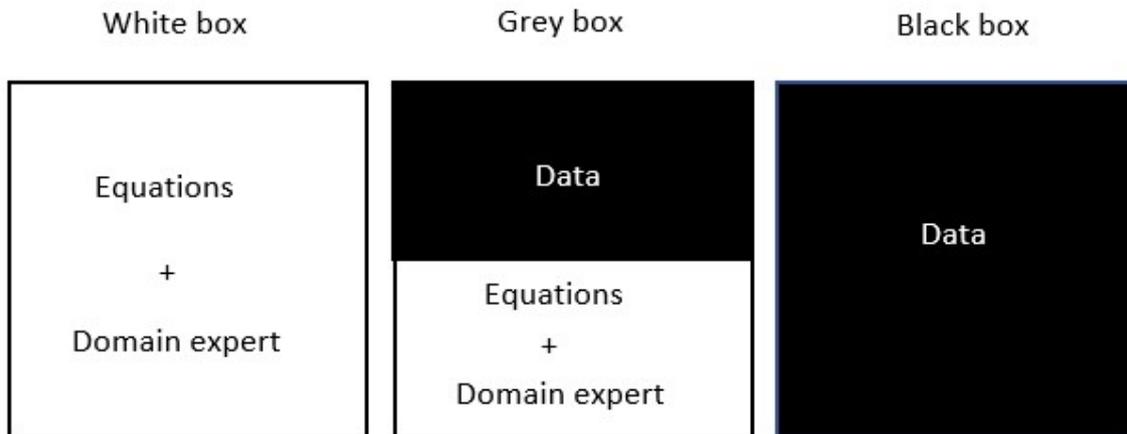


Figure 2.2 Diagrammatic depiction of modelling techniques

2.6 Deep learning

Deep learning is a subset of machine learning which involves the use of multiple hidden layers in an ANN. These multiple hidden layers constitute what is referred to as deep architecture, hence the name 'deep learning' [88]. Deep learning as a concept was originally proposed as a concept in the early 1940s but suffered some major setbacks due to lack of sufficient data, computer resources, and training efficiency [88, 89]. However, with the advancement in science and technology in terms of computing resources and the availability of 'big' data and highly efficient training algorithms [90], the stage was set for deep learning to thrive. Deep learning uses representative learning to perform multiple levels of abstraction on raw data using its multiple hidden layers. This method of learning enables deep learning to reveal intricate structures in large datasets and compute the representation in each layer from the representation of the previous layer [91]. Unlike conventional machine learning, which was limited in processing raw data and requires a domain expert to perform signal processing and feature extraction from the original data, deep learning can utilise raw data for training and classification (as illustrated in Figure 2.3). Many aspects of modern life have been greatly influenced by deep learning. These include; cell phones, internet search engines, smart devices, social media, speech to text transcription, voice command, and match news items. Recently, many researchers have contributed to the body of knowledge of deep learning by adapting the

technology to solve problems that otherwise would be difficult to solve using other methods in diverse fields of study.

Hinton [92] developed a breakthrough concept of efficiently teaching individual neurons, thereby introducing Deep Belief Networks (DBN). It achieves this by taking advantage of the restricted Boltzmann machine (RBM), a learning algorithm that greedily trains one layer at a time. The proposed DBN achieved considerably high precision in phonetic categorization for automatic voice recognition than the conventional machine learning approach and was the first large commercial use of deep learning. On the other hand, Alex Krizhevsky [93] achieved a massive breakthrough in image recognition in 2012 when his team won first place at the world competition of image recognition called ImageNet scale visual recognition challenge (ILSVRC). Their algorithm, also known as AlexNet, achieved a low error rate of 15.3 % compared to 26.2 % achieved by its closest competitor. Researches by Graves et al. [94] and Pascanu [95] shows that recurrent neural network (RNN) stacked in a deep architecture produces significantly better performance (achieving an error rate of about 17.7 %) than its conventional counterpart. Other fields of application of deep learning include image recognition [96, 97], speech recognition [98, 99], medicine [100, 101], engineering [88, 102], natural language analysis [103-105], social media [106, 107]. These deep learning approaches further reduce the influence of an expert agent, while offering good accuracy, in the decision making as there is no need for feature extraction. This makes the black-box approach (machine learning and deep learning) good candidates to investigate the particular problem scenario for this thesis. Which is, the ball screw assembly is not directly observable and the detail of the internal construction of the system is unknown. Also, the exact fault condition is unknown, this make it impossible to design a white-box model based on physical properties that are not known. Hence, the decision to use machine learning and deep learning for this research.

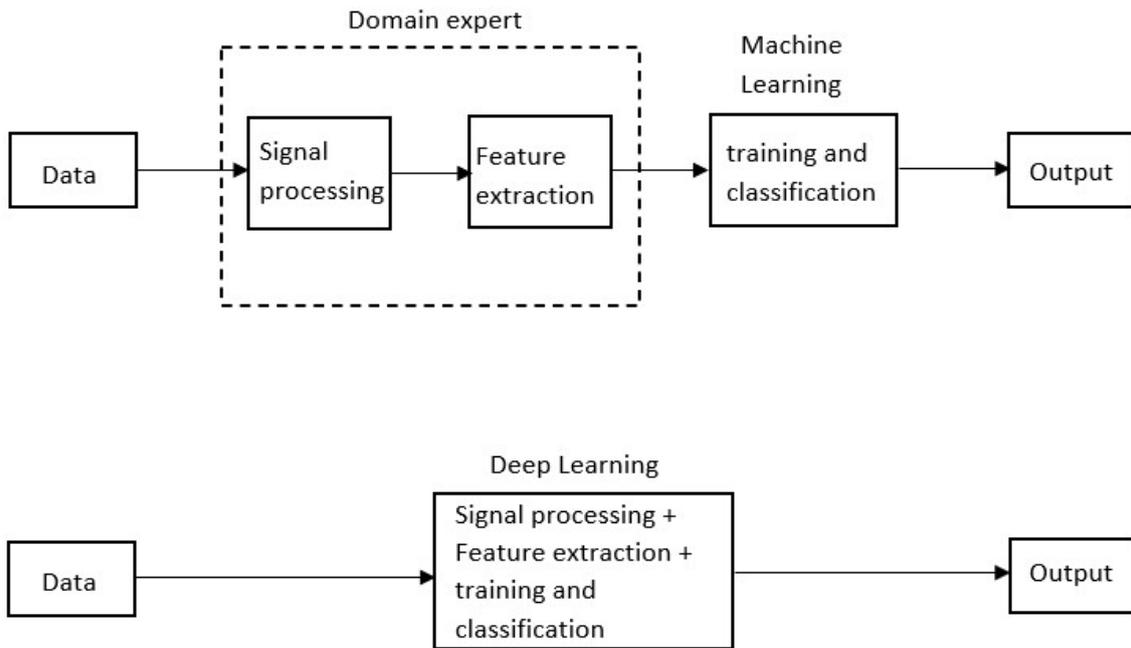


Figure 2.3 Machine learning versus Deep learning block diagram

2.7 Aim and objectives of the research

This research aims to develop an intelligent condition monitoring system for machine tools balls screw feed drives based on sensor data and directed by artificial intelligence. The proposed solution will lay the technical foundation upon which the machine tool servitisation business model can be supported by condition monitoring technology. The solution would perform early and accurate fault detection such that maintenance activities are scheduled before catastrophic failure occurs.

The aim is addressed by the following research questions:

1. What method of sensing is most appropriate for monitoring degradation in ball screw feed drive systems?
2. What method of AI provides the best balance between efficiency and accuracy when detecting a damaged ball screw?

The following project management objectives represent the step-by-step method of how these research questions were answered:

1. Conduct a detailed literature review of the historical and current state of condition monitoring and the methods used to determine the most suitable sensors or signals to be used for this condition monitoring problem domain.
2. Conduct a detailed literature review of the historical and current state of feature extraction, machine learning, and signal analysis.
3. Develop a feature extraction method based on the most suitable technique identified in the literature review to obtain valuable information from continuous data suitable for machine learning algorithms.
4. Validate and optimise the proposed machine learning algorithm.
5. Define and develop an appropriate architecture for the deep learning model to detect the faulty ball screw.
6. Validate the proposed deep learning algorithm.
7. Critically appraise the machine learning and deep learning algorithms used for monitoring the degradation of the ball screw through testing on a machine tool test rig.

2.8 Scope

The scope of this work is to perform condition monitoring of the ball screw of a CNC machine tool under two states (good and worn) using artificial intelligence and signal processing. The focus is on using machine learning and deep learning methodology to accurately predict the state of the ball screw, rather than traditional frequency and mathematical analysis that can be prone to bias and preconceptions. This process involves literature review, design of experiment, signal processing, data analysis to obtain results, and validation of the results. Vibration and acoustic emission sensors were used to collect continuous time-series data from the ball screws of a gantry-type CNC machine tool, which is then analysed with machine learning and deep learning. The validation test is also conducted on the same machine by performing several experiments with varying conditions.

The scope does not include the investigation of the progressive degradation of the ball screw through the accelerated wear on a test rig. This was as a result of time and logistical constraints due to the covid pandemic; this could not be achieved. Instead, it has been included as further work. The results from this thesis have laid the foundation for a larger project to quantify degradation at different stages of wear.

2.9 Novelty of the research and contribution to knowledge

This research investigates the application of AI to condition monitoring in machine tools, which includes comparing machine learning and deep learning. Systems applying AI to condition monitoring have been a source of extensive research in the past decade. The overarching novelty in the thesis is the use of a deep learning algorithm based on convolution analysis for the assessment of the health condition of a ball screw assembly in a machine tool. The main contribution to knowledge derives from critically comparing this technique with optimised machine learning models to provide guidance for future application and research in this area.

Other novelties that arose from delivering this outcome include the method of signal processing employed to pre-process the data in the machine learning models, whereby wavelet transform and FFT are utilised in preparing the data for effective feature extraction. The results are compared with existing methods to show their benefit, and other researchers could investigate whether they would be suitable for different applications.

The second supplementary novelty is the method of formatting the time-series data into image files, making it suitable for use in the convolution deep learning model and analysing the influencing factors when undertaking this pre-processing. It is anticipated that this approach could be applied to other time-series signals in manufacturing or other problem domains.

2.9.1 Potential impact of the research on industry

This research, as it fits into the servitisation model, presents several potential contributions to knowledge and industry that are highlighted below:

1. Its approach will improve manufacturing industries' in-service support, which will boost efficiency, improve technological advancement, and enhance manufacturing sustainability across the supply chain.
2. Companies, especially SMEs, will gain access to cutting-edge CNC machines with guaranteed quality, performance, and availability. This will lead to a more productive and sustainable manufacturing facility, which will lead to an increase in the company's market share.
3. It supports the simple manufacturing infrastructure and is therefore applicable to all sectors where precision machining is necessary (aerospace, automotive, and power generation).
4. It could help lower maintenance costs as there will be no overheads in supporting a full maintenance department. This will lead to savings in service cost, reduced average machine downtime, and an increased throughput volume without any increase in labour costs.
5. It could lead to improved data management and analysis techniques that can be used in other fields where long-term monitoring is necessary.
6. The improved maintenance resulting from this project will improve maintenance culture, including reducing intensity and type of repair actions, materials and spares use, rate and amount of failure, and waste from scrap. This will lead to low energy consumption levels and an even lower carbon footprint.
7. It has the potential to improve the health and safety benefits for companies since it reduces the risk of catastrophic failures that might be injurious to workers.

2.10 Methodology

The methodology of this research involves the use of two AI algorithms for detecting and distinguishing the vibration signature of different states of a machine tool in order to adequately represent the condition of the machine tool at any given time. The first of these is machine learning, while the other is deep learning. In the machine learning process, wavelet transform and FFT are used for signal processing, followed by feature extraction. The extracted features are then processed with several machine learning algorithms to assess the quality of extracted feature in representing the condition of the machine tool. In the deep learning process, an intelligent image processing algorithm capable of detecting and distinguishing vibration signatures based on predetermined classes is developed. It uses the principle of convolution analysis for image processing, feature identification, and extraction.

The datasets used for the research were obtained from experiments performed on-site at the Centre Precision Technology (CPT) machine workshop at the University of Huddersfield. The experiment was performed on a gantry-type 5-axis CNC machine tool with two similar ball screws. One ball screw assembly had been starved of lubrication due to a fault. This allowed a good and a worn balls screw to be compared for a single test, i.e., reducing uncertainty from doing good/bad tests on different machines. The datasets were collected under various conditions, including different feed rates and thermal perturbation (see Chapter 5 for full detail). The datasets were used for developing, testing, and validating the AI models proposed. For the purpose of preventing overfitting in the proposed AI models during training, a portion of the datasets is designated as the validation set. The validation set is used to determine the epoch of training required such that overfitting does not occur.

2.11 Chapter summary

There are three different approaches to machine maintenance: operate-to-failure or reactive maintenance, time-based or preventive maintenance, and condition-based or predictive maintenance. Among these three, condition-based maintenance is arguably the best because it offers the machine tool user a good balance between maintenance cost and availability (and accuracy). To implement an effective condition monitoring strategy, different modelling techniques can be explored. AI techniques like machine learning and deep learning are always often used. The popularity of these techniques extends beyond condition monitoring and engineering, with applications in politics, agriculture, medicine, tourism, finance, and many more. The next chapter delves into machine learning, signal analysis, and feature extraction.

Chapter 3 Machine learning

This chapter provides a more detailed literature review and appraisal of machine learning as it applies to this research, leading to the proposed methods that are most likely to produce good results for this problem domain.

It starts by providing a review of the methods of signal analysis commonly used for data preparation. This is followed by a description and appraisal of some machine learning algorithms and their applications and their relation to this thesis's problem domain. Next is a description of the extraction of statistical features often used in machine learning applied in this research. This chapter highlights all the important concepts that make machine learning achievable for condition monitoring of ball screw feed drive systems. Materials used for this chapter include journals, conference papers, and textbooks.

3.1 Signal pre-processing

Typically, there must be some form of signal processing after data acquisition before the data can be utilised. This is done in most cases because of the need to reduce the amount of data, the noise that accompanies the data, or in other instances, to rearrange or reorganise the dataset. In general, signal processing aims to improve signal components in noisy measurements or transform measured datasets so that new features are available. Hence, data processing serves as a significant prerequisite for feature extraction. However, the viability of the features that can be extracted from a given signal depends on the quality of the signal processing technique used. There are several techniques for signal processing, and they include wavelet transform [108, 109], Fourier analysis [110], short-time Fourier transform (STFT) [111], correlation [112], and Hilbert-Huang [113, 114].

3.1.1 Fast Fourier transform (FFT)

The Fourier transform of a continuous signal $x(t)$ decomposes the signal into its constituent frequency components over the analysis window, is given by (3.1). It provides a quick insight into a given signal's frequency content, which is important for certain applications. However, it has a major flaw (which is remedied in STFT), which is, it has no information on time-domain localisation.

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2j\pi ft} dt \quad (3.1)$$

Where f is frequency and t is time.

STFT tries to achieve both time localisation information and frequency constituent with a fixed window. Unfortunately, due to the uncertainty principle, it is not possible to accurately determine the time localisation and the frequency localisation information simultaneously in the time-frequency plane. In order to accurately determine the time localisation information, the time analysis window has to be short. Whereas to accurately determine the frequency localisation information, the time analysis window has to be long. This results in a trade-off between the time and frequency localisation information [115]. Mathematically, the STFT of a continuous signal $x(t)$ is given by (3.2), which shows that the signal is multiplied by a nonzero window function $g(\cdot)$ for a short time τ . Then the Fourier transform of the resultant signal is taken as the window moves across the time axis. This results in a time-frequency representation of the signal.

$$STFT(\tau, f) = \int_{-\infty}^{\infty} x(t) g(t - \tau) e^{-2j\pi ft} dt \quad (3.2)$$

Fast Fourier transform (FFT) is a collection of several fast discrete Fourier transform (DFT) computation methods. DFT is a descriptive mathematical method used to investigate the correlation between the time domain and the frequency domain representation of discrete data.

DFT can generally be defined as the transformation of discrete signals from the time domain to the frequency domain and vice versa. Typically, the original signal is in the time domain, and it is required to convert to the frequency domain for analysis. Such frequency constituents are the sinusoidal composition of the original signal in its particular amplitude and frequency value. FFT analytical methodology was first used in the 1960s and steadily increased in popularity [116]. Many FFT algorithm depends on the fact that $e^{-j2\pi/N}$ will yield unity when raised to power N and hence, can be applied to comparable transforms over any Galois Field (a field that contains a finite number of elements). The DFT is mathematically denoted as [116];

$$\bar{X}_k = \sum_{m=0}^{N-1} x_m W^{mk} \quad (3.3)$$

Where \bar{X}_k is the DFT of sequence x_m with N consecutive samples, $W = e^{-j2\pi/N}$, $j = \sqrt{-1}$ and $k = 0, \dots, N - 1$. The inverse transform y_l is mathematically denoted as [116];

$$y_l = \frac{1}{N} \sum_{k=0}^{N-1} \bar{X}_k W^{-lk} \quad (3.4)$$

Where $l = 0, \dots, N - 1$.

To verify the validity of (3.4), substitute (3.3) into it. This implies that;

$$y_l = \sum_{m=0}^{N-1} x_m \frac{1}{N} \sum_{k=0}^{N-1} W^{(m-l)k} \quad (3.5)$$

Since $W^N = 1$, $m - l$ is defined as modulo N .

For $m - l \equiv 0 \pmod{N}$,

$$S = \sum_{k=0}^{N-1} W^{(m-l)k} = N \quad (3.6)$$

For $m - l \not\equiv 0 \pmod{N}$,

$$S = \frac{[W^{N(m-l)} - 1]}{W^{m-l} - 1} \quad (3.7)$$

and since $W^{m-l} \neq 1$, $S = 0$.

Therefore, the only nonzero case occurs when $l \equiv m$, which results in $y_l = x_m$.

3.1.2 Wavelet transforms

Wavelets were first introduced during the early 1980s by a French geophysicist named J. Morlet as a mathematical tool for analysing seismic data [117]. Wavelet transform is a method of spectral approximation that can be used to express set functions as an infinite series of wavelets [118]. This is done by scaling (dilating and contracting) and shifting a function as a linear combination of a specific set of functions (wavelets) with the help of a mother wavelet, scaling parameter, and translation parameter. The original signal can be reconstituted by an algebraic summation of the wavelet functions weighted by their corresponding wavelet coefficients. Wavelet transform is quite similar to DFT and STFT in terms of signal theory. However, wavelets offer time-frequency localisation of a signal [119, 120]. This means that wavelet transform has a high-frequency resolution at low frequencies while also having a high time resolution at high frequencies, unlike STFT, which offers a constant frequency resolution based on predefined window size at low frequencies and high frequencies [121]. On the other hand, DFT can only provide analysis in the frequency domain over the analysis window and, as such, does not provide any information on time-domain localisation. Figure 3.1 shows an illustration of this phenomenon.

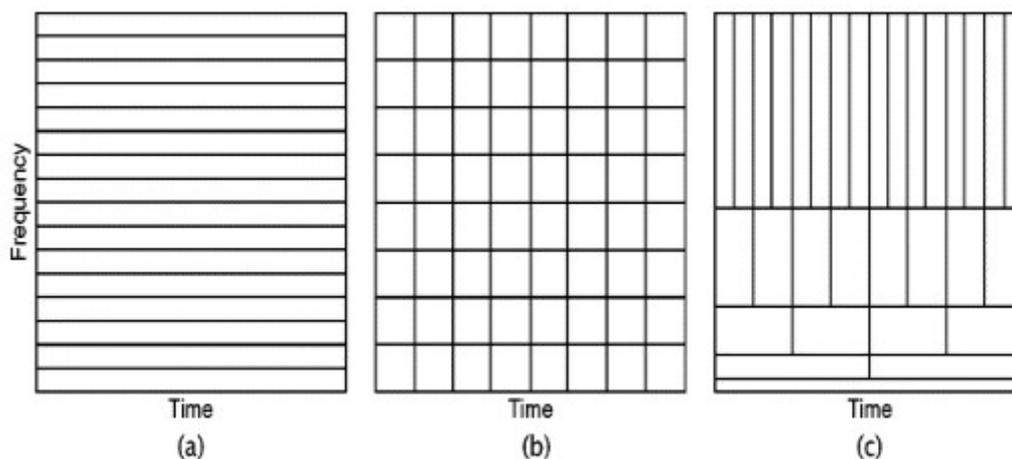


Figure 3.1 Frequency vs time resolution for (a) DFT, (b) STFT, and (c) DWT [121].

Gao and Yan [122] assert that the signal-to-noise ratio of data can be improved by employing wavelet transform. This is realised by letting the change in time (Δt) and change in frequency (Δf) resolution vary in the time-frequency plane in order to achieve a multi-resolution analysis. In other words, it allows the frequency responses of the analysis filter to be regularly spread on a logarithmic scale. A significant advantage of wavelet transform over fast Fourier transform (FFT) or short-time Fourier transform (STFT) is that it can capture and expose frequencies otherwise hidden from FFT and STFT. Also, it can adequately analyse and construct filters for stationary and non-stationary signals [123]. Mathematically, a wavelet must satisfy the following conditions:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \quad (3.8)$$

And

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (3.9)$$

There are many different wavelet families; some of them are shown in Table 3.1. with their corresponding abbreviations.

Table 3.1 Wavelet families and abbreviations

Wavelet family	Abbreviation
Haar	haar
Daubechies	db
Symlets	sym
Coiflets	coif
BiorSplines	bior
ReverseBior	rbio
DMeyer	dmey
Fejer-Korovkin	fk

They are two broad types of wavelet transform analysis; continuous wavelet transforms (CWT), and discrete wavelet transform (DWT).

3.1.2.1 Continuous wavelet transforms

For a given input signal $x(t)$, the CWT is a product of the convolution of the signal $x(t)$ and a complex conjugate of the mother wavelets $\psi(t)$, this is expressed mathematically as;

$$cwt(s, \tau) = \frac{1}{\sqrt{s}} \int x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt \quad (3.10)$$

where the symbols s represents the scaling parameter, τ represents the shifting parameter, the term $\psi^*(\cdot)$ represents the complex conjugate of the scaled and shifted mother wavelet. From (3.10), it can be observed that the CWT is analogous to the Fourier transform. The difference is that the sine and cosine functions are replaced by the mother wavelet function, which contains two parameters (scaling and shifting parameters). The implication of the mother wavelet is that the signal will be projected into a two-dimensional plane, a time-frequency plane, instead of one, as in the case of Fourier transform. In the frequency domain, (3.10) is mathematically expressed as;

$$\begin{aligned} CWT(s, f) &= F\{cwt(s, \tau)\} \quad (3.11) \\ &= \frac{1}{\sqrt{s}2\pi} \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt \right) e^{-j2\pi f \tau} d\tau \\ &= \sqrt{s}X(f)\Psi^*(sf) \end{aligned}$$

where the symbol F represents the Fourier transform operator, $X(f)$ represents the Fourier transform of $x(t)$ and $\Psi^*(f)$ represents the Fourier transform of $\psi(t)$. This results in a two-dimensional transform in which the scaling and shifting parameters are changed continuously.

By taking the inverse Fourier transform of (3.11) in the time domain results in;

$$cwt(s, t) = F^{-1}\{CWT(s, f)\} = \sqrt{s}F^{-1}\{X(f)\Psi^*(sf)\} \quad (3.12)$$

where the symbol F^{-1} represents the inverse Fourier transform operator. This will technically require an infinite number of coefficients which makes it computationally tedious and inefficient.

3.1.2.2 Discrete wavelets transform

The problem of high computationally overhead and inefficiency associated with the CWT can be reduced by discretizing the CWT [124]. This can be achieved by using dyadic scaling and shifting parameters. This implies that s and τ from (3.10) are replaced by 2^j and $2^j k$ respectively. The resulting equation is the DWT and it is given by;

$$wt(j, k) = \langle x(t), \psi_{j,k}(t) \rangle = \frac{2}{\sqrt{2^j}} \int_{-\infty}^{\infty} x(t) \psi^* \left(\frac{t - k2^j}{2^j} \right) dt \quad (3.13)$$

Where j is the depth of decomposition, $\psi_{j,k}(t)$ is the wavelet function in the time domain, ψ^* is the complex conjugate of the wavelet function in this equation, also $j \in \mathbb{R}, k \in \mathbb{R}, t \in \mathbb{R}$ [122, 125]. The DWT decomposes the original signal A_0 into approximations A_{-1} and details D_{-1} , then the approximation A_{-1} is further decomposed into approximations A_{-2} and details D_{-2} , and so on until the desire level is attended. This is illustrated in Figure 3.2 and (3.14) gives the mathematical expression for the coefficients of the approximation component $A_{j,k}$ and the coefficients of the detail components $D_{j,k}$

$$DWT_{f(n)} = \begin{cases} A_{j,k}(n) = \sum_n f(n) G_j^*(n - 2^j k) \\ D_{j,k}(n) = \sum_n f(n) H_j^*(n - 2^j k) \end{cases} \quad (3.14)$$

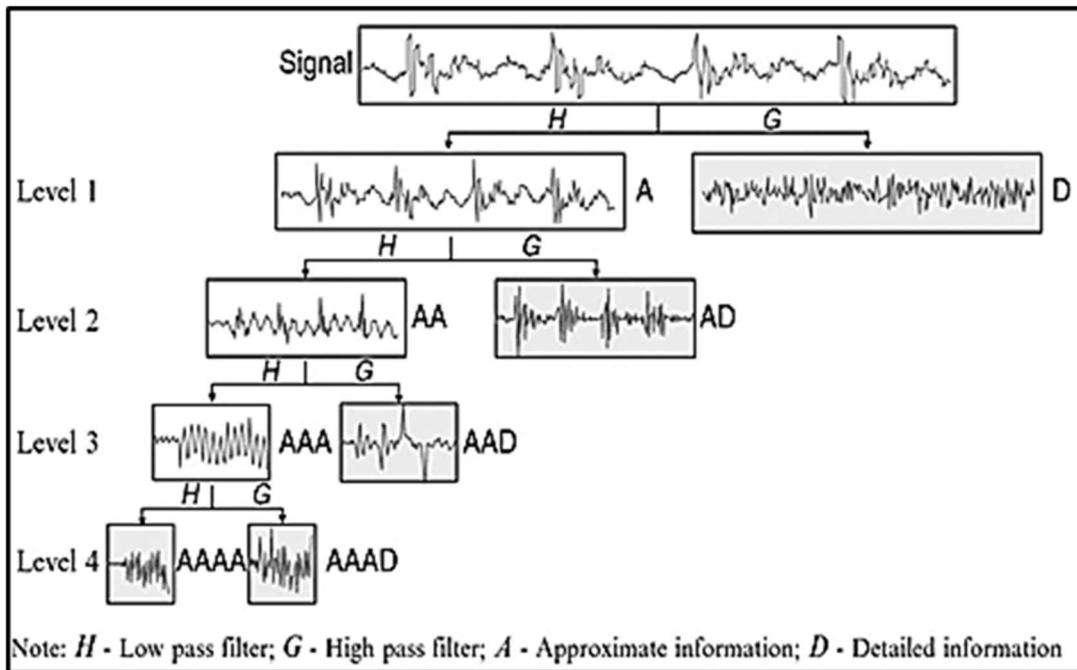


Figure 3.2 Wavelet transform breakdown of a signal [122]

From the diagram in Figure 3.2, *G* is a high pass filter, and *H* is a low pass filter. The impulse response of the high and low pass filters $g(k)$ and $h(k)$ respectively is related by (3.15) [125]

$$g(k) = -1^{1-k}h(1 - k) \quad (3.15)$$

3.2 Machine learning algorithms

Machine learning is a part of artificial intelligence that can learn and predict outcomes automatically from data input without being explicitly programmed. It identifies patterns within data structures and their relationships to particular outcomes, using it to learn and predict outcomes from future datasets from the same process. Machine learning is broadly divided into two; supervised and unsupervised. The major difference between the two is that supervised learning will have its outcome predefined, whereas unsupervised learning will not. Machine learning is becoming more common among researchers and particularly amenable to modelling varied systems, from the stock market to complex engineering processes. This is due to its demonstrable superior predictive ability compared to traditional (statistical) methods [81, 84]. However, before machine learning can occur, a process known as feature extraction must have

been done on the data. This helps to facilitate the pattern identification and inference ability of the machine learning algorithm. Figure 3.3 shows the block diagram of the machine learning process.

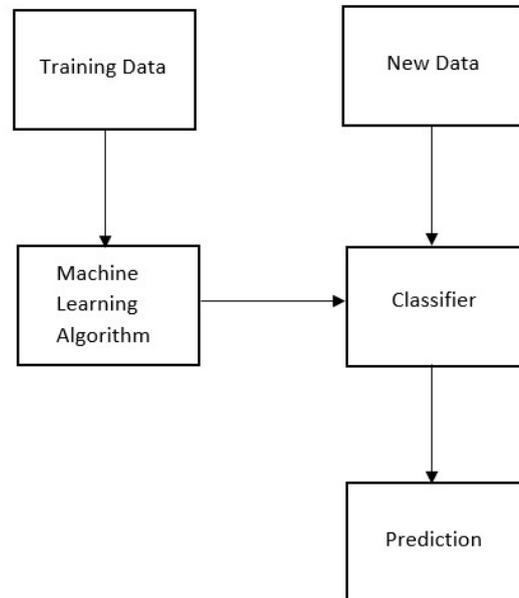


Figure 3.3 Block diagram of a machine learning process

There are several machine learning algorithms, and some of the ones that have shown to give good results for intelligent fault diagnosis of rotating machines [126] include; decision tree, support vector machine, neural network, k - nearest neighbour,

3.2.1 Decision tree

Decision trees are a supervised machine learning algorithm that is mostly used for classification problems. It is a classifier algorithm that asks a question and then makes classification decisions based on the answer to those questions. They start at the top, and each answer progresses along a path until a point is reached where the path ends, and there are no further questions. The questions' responses can be numerical, ranked, multiple-choice, or simple “yes/no” answers. Figure 3.4 shows a typical decision tree; the top of the tree (A) is referred to as the root node (or root), B, C, D, and E are called nodes (they have arrows pointing to them and away from them), F and G are the leaves (they have arrows pointing to them, just like the nodes, but no arrows pointing away from them).

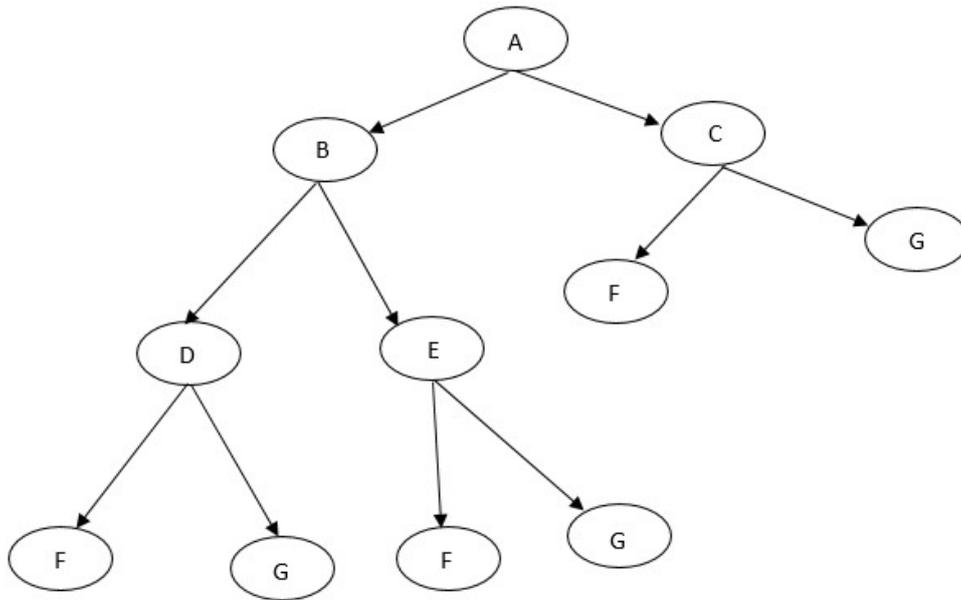


Figure 3.4 Decision tree structure

Decision trees have a flowchart structure in which each node represents the criteria for classification. Each branch represents the criteria' attribute, each leaf represents a class, and the topmost nodes represent the root [127]. Decision trees can be used for regression and classification type problems. Regression trees are used when the dependent variable is continuous, and classification (or splitting) is determined using the mean values.

On the other hand, classification trees are used when the dependent variable is continuous, and classification (or splitting) is determined using the modal class of values. This splitting process is continued until a user-defined stopping criterion is reached; this results in a fully grown tree. A fully grown tree can be computationally costly because of the number of nodes. In order to counter this, the number of nodes can be reduced by a process called pruning. Pruning is the process of terminating nodes on a decision tree with negligible additional information. The process is aimed at reducing the size and hence complexity of the tree without compromising accuracy.

3.2.1.1 Decision tree splitting rules

Decision trees use several splitting rules (or attribute selection measures) to split a node into two or more sub-nodes. The sub-nodes are split according to the homogeneity of the data. A

splitting rule is a heuristic for the splitting criterion to achieve the best data homogeneity within a class out of the number of specified classes. The commonly used splitting rules are gain ratio, Gini index, chi-square, information gain, reduction variance, Gini impurity.

3.2.1.1.1 Information gain

This splitting rule uses the attribute with the highest information gain as the node for splitting attributes. The information gain is given by [127]

$$Gain(A) = info(D) - info_A(D) \quad (3.16)$$

Where

$$info(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (3.17)$$

And

$$info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j) \quad (3.18)$$

Where D is the tuple of A attributes having v number of classes, p_i is the probability that a random tuple in D fits in class C_i and is given by the expression $|C_{i,D}|/|D|$, m represents the total number of distinct leaf nodes and D_j represents the group of homogeneous data within a node.

The attribute with the highest gain becomes the node. This process proceeds recursively on each of the child nodes until each partition is pure (with homogenous data), and the node becomes a leaf node. The process is similar when considering continuous data; however, it occurs at a higher computational cost.

3.2.1.1.2 Gain ratio

The gain ratio of a dataset D , of A attributes having several classes, is given by

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \quad (3.19)$$

Where

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|} \right) \quad (3.20)$$

The attribute that has the highest gain ratio becomes the node. This process proceeds recursively on each of the child nodes until each partition is pure (or until all attributes have been examined), and the node becomes a leaf node. Note that as the denominator in (3.19) tends to zero, the gain ratio becomes unstable. This is mitigated by introducing a constraint in which the information gain of the attribute selected as node must be larger than the average gain from all attributes.

3.2.1.1.3 Gini index

The Gini index is a measure of the impurity in the tuple D and is given as

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2 \quad (3.21)$$

The Gini index is used for examining the binary partitioning of attribute A . if the tuple D of attribute A is to be partitioned into subsets D_1 and D_2 , then the Gini index A is

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \quad (3.22)$$

This is a weighted sum of the impurities in each partition. The weighted sum is used because typically D_1 and D_2 are impure, and $|D_1| \neq |D_2| \neq |D_3|$. The attribute with the lowest Gini index $Gini_A(D)$ becomes the node. This process proceeds recursively on each of the child nodes until a point is reached where $Gini(D)$ is less than all examined $Gini_A(D)$. At this point, the current node becomes a leaf node, and the class is determined by voting. Note that this will be an impure leaf node.

3.2.1.2 Tree pruning

The goal of decision trees is to have leaf nodes without impurities. However, this compromises the decision tree's accuracy as any new data that does not match the pattern in the data used to design the decision tree will be misclassified. This phenomenon is known as overfitting. Overfitting is when a decision tree (or any other machine learning algorithm) performs well for the training data but poorly on independent test data. This problem can be addressed by pruning the decision tree. Pruning is the process of removing less reliable branches from a decision tree in order to generalise it for unseen data. Figure 3.5 and Figure 3.6 show diagrams of an unpruned tree versus a pruned decision tree. A pruned tree is less complex but computationally efficient with the added advantage that it is not overfitting on the training data; hence it has better accuracy than its unpruned version. Pruning is achieved by imposing stoppage criteria such that if those criteria are met during the building of the tree at a specific node, that node becomes a leaf node. Even though pruning trades accuracy for simplicity at its core, its objective is to create an adequately accurate and compact model that is not overfitted [128, 129].

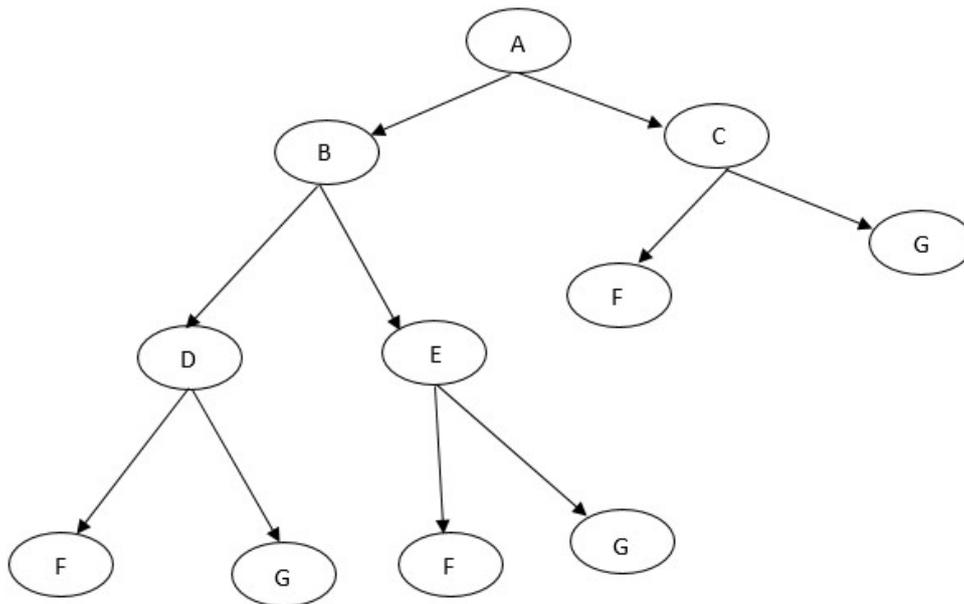


Figure 3.5 An unpruned decision tree

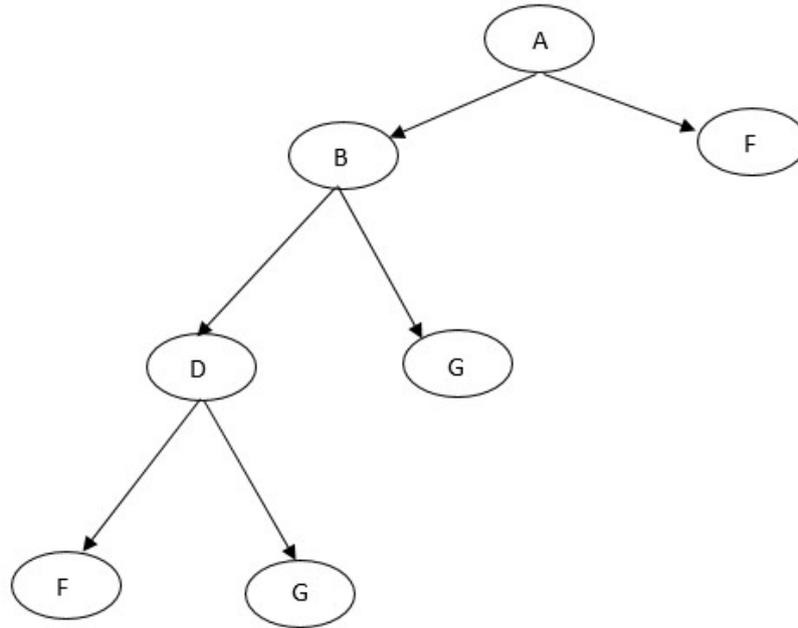


Figure 3.6 A pruned decision tree

There are many different ways of achieving pruning, and they include; cost complexity pruning, reduced error pruning, minimum error pruning, and pessimistic pruning [128]. The cost complexity pruning (or weakest link pruning) uses the lowest apparent error rate increase per pruned leaf as the criteria for pruning branches, given by:

$$\alpha = \frac{\varepsilon(\text{pruned}(T, t), S) - \varepsilon(T, S)}{|\text{leaves}(T)| - |\text{leaves}(\text{pruned}(T, t))|} \quad (3.23)$$

Where $\varepsilon(T, S)$ is the error rate of the tree T over the sample S , $|\text{leaves}(T)|$ is the number of leaves in the tree, and $\text{pruned}(T, t)$ is the tree obtained after pruning a node t on tree T . All the trees T_0, T_1, \dots, T_k (where T_0 is the initial tree before pruning and T_k is the root node) obtained from the different instances of pruning is accessed. Then the tree with the lowest generalization error is selected as the pruned tree.

Reduced error pruning is a pruning strategy that analysis the nodes of a tree from the bottom up to determine whether replacing it with a leaf node will reduce its accuracy. If the accuracy is not reduced, the tree is pruned. If not, the node is left unpruned. A pruning set is typically required to achieve this.

The minimum error pruning involves using (3.24) to access the error rate ε' before and after pruning a particular internal node, the node is pruned if the error rate before pruning is greater than the error rate after pruning.

$$\varepsilon'(t) = 1 - \max_{c_i \in \text{dom}(y)} \frac{|\sigma_{y=c_i} S_t| + (l)p(y = c_i)}{|S_t| + l} \quad (3.24)$$

Where $p(y = c_i)$ is the a-priori probability of y having the value c_i , l is the weight of $p(y = c_i)$ and S_t is the instances that have reached a leaf t .

Pessimistic pruning advocates the use of the continuity correction for binomial distribution, given by:

$$\varepsilon'(T, S) = \varepsilon(T, S) + \frac{|\text{leaves}(T)|}{2|S|} \quad (3.25)$$

An internal node is pruned if the error rate satisfies the following condition:

$$\varepsilon'(\text{pruned}(T, t), S) \leq \varepsilon'(T, S) + \sqrt{\frac{\varepsilon'(T, S)(1 - \varepsilon'(T, S))}{|S|}} \quad (3.26)$$

Where T is a subtree with root node t and S is the part of the training data that refers to the node t .

3.2.2 Support vector machine

Support vector machine (SVM) is a supervised machine learning algorithm that uses decision boundaries for classification and regression analysis. It was first discovered in 1992 by a mathematician named Vladimir Vapnik and his colleagues [130]. SVM achieves classification analysis by transforming the input dataset into an n -dimensional space (where n corresponds to the number of features) and separating them by class with an optimal hyperplane. In order to separate two classes of data points, several possible hyperplanes can be drawn (see Figure 3.7). SVM's goal is to find a hyperplane with the maximum distance between the support vectors (the data points that lie closest between the two classes) [131]. This is illustrated in

Figure 3.8. The simplest form of SVM is used for linear classification and is based on the idea of maximising margins. However, more complex forms can be applied to non-linear classifications by using kernels [132]. Given a set of input-output pair of training data $\{(v_j, y_j) : j = 0, 1, \dots, n - 1\}$; where the class label is indicated by $y_j \in \{+1, -1\}$ in an n -dimensional space R^m , the two classes can be separated by a number of possible hyperplanes. The most efficient separation is achieved by maximising the margins of the support vector. The hyperplane at this instance is the optimal hyperplane; it is parallel to the two planes defined by the support vector called the positive and negative hyperplane, as shown in Figure 3.8. The positive and negative hyperplane satisfies the (3.27) and (3.28) [132].

$$w^T v_j + b \geq 1 \quad (3.27)$$

for

$$y_i = +1$$

and

$$w^T v + b \leq 1 \quad (3.28)$$

for

$$y_i = -1$$

Where W and b are the weight vector and bias, respectively. Equations (3.27) and (3.28) can be combined to form;

$$y_j(w^T v_j + b) \geq 1 \quad (3.29)$$

for

$$j = 0, 1, \dots, n - 1.$$

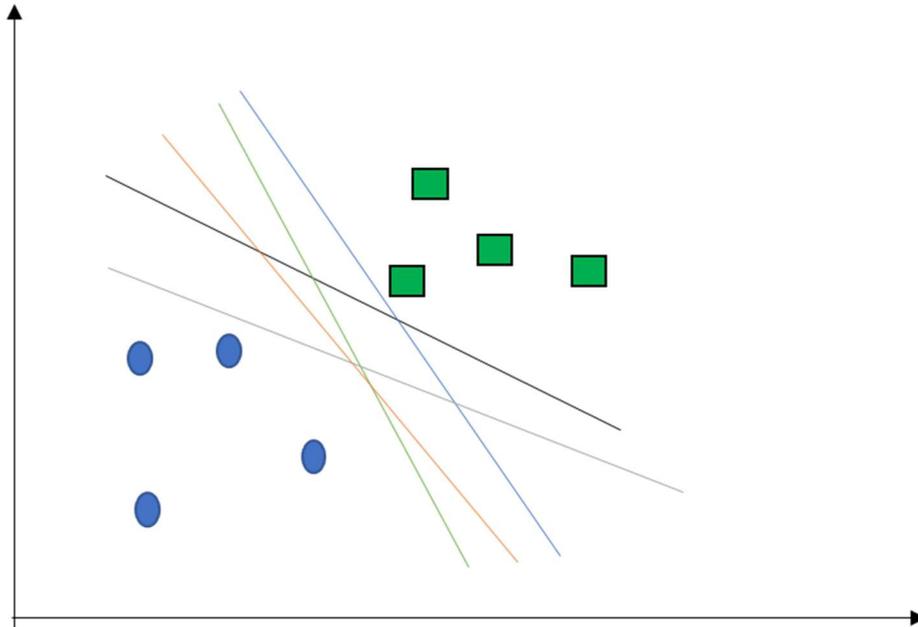


Figure 3.7 SVM algorithm in an n -dimensional space

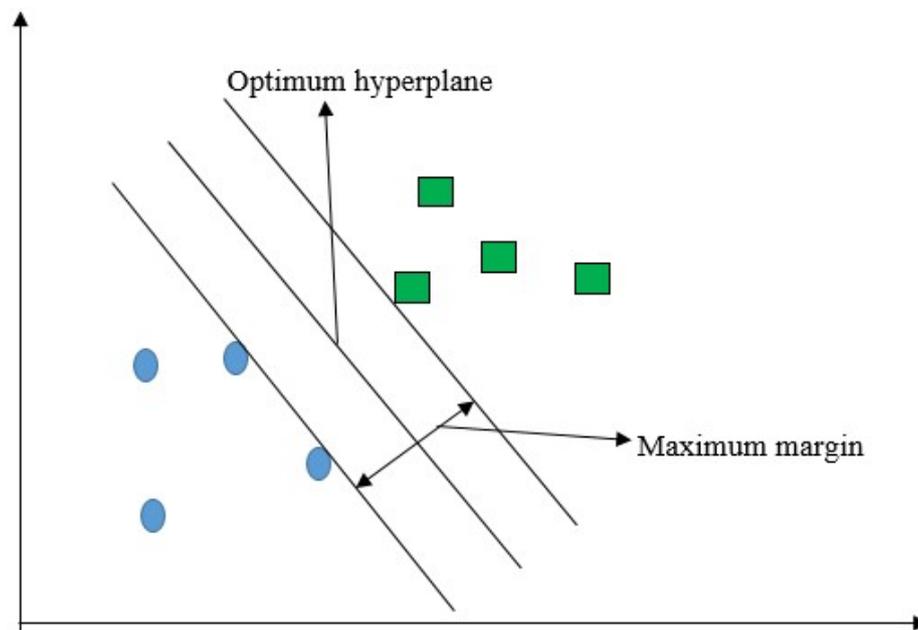


Figure 3.8 A maximal margin hyperplane with its support vectors

The positive, negative, and optimal hyperplane can be described by (3.30), (3.31), and (3.32), respectively.

$$w^T v + b = 1 \quad (3.30)$$

$$w^T v + b = -1 \quad (3.31)$$

$$w^T v + b = 0 \quad (3.32)$$

The distance from a point (x_0, y_0) to a line, $Ax + By + c = 0$ is given by

$$\frac{|Ax_0 + By_0 + c|}{\sqrt{A^2 + B^2}} \quad (3.33)$$

Hence the distance γ between two corresponding margins in an SVM multidimensional space is

$$\gamma(w, b) = \min_{\{v|y=+1\}} \frac{w^T v}{\|w\|} - \max_{\{v|y=-1\}} \frac{w^T v}{\|w\|} = \frac{1}{\|w\|} \quad (3.34)$$

Hence the maximum distance γ_{max} between the positive and the negative hyperplane is given by

$$\gamma(w^*, b^*) = \frac{2}{\|w\|} \quad (3.35)$$

Where W^* is the optimal weight and b^* is the optimal bias. It is evident from (3.34) and (3.35) that in order to maximize the margin of separation of the positive and the negative hyperplane, the term $\|w\|$ must be minimised. To solve this optimisation problem, the Lagrange function is introduced;

$$L_p(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{j=0}^{n-1} \alpha_j (y_i (w^T v_j + b) - 1) \quad (3.36)$$

Where $\alpha_j \geq 0$ are Lagrange multipliers. The solutions can be found by differentiating the Lagrange function with respect to W and b , imposing stationarity, to obtain (3.37) and (3.38)

[133]

$$w^* = \sum_{j=0}^{n-1} \alpha_j y_j v_j \quad (3.37)$$

$$\sum_{j=0}^{n-1} \alpha_j y_j = 0 \quad (3.38)$$

Substituting (3.37) and (3.38) into (3.36) gives;

$$L_p(w, b, \alpha) = \sum_{i=0}^{n-1} \alpha_i - \frac{1}{2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \alpha_i \alpha_j y_i y_j v_i^T v_j \quad (3.39)$$

Among the solutions of maximising the Lagrange function will exist points located on the two optimal margins with nonzero coefficients. These vectors are referred to as support vectors and are responsible for influencing the SVM classifier. The bias is calculated from (3.40).

$$b^* = -\frac{1}{2} \left(\min_{v_j | y_j = +1} w^{*T} v_j + \max_{v_j | y_j = -1} w^{*T} v_j \right) \quad (3.40)$$

Consequently, the decision function can be written as;

$$f(x) = \sum_{j=0}^{n-1} \alpha_j y_j v_j^T v + b^* \quad (3.41)$$

The above solution is for situations where the problem is linearly separable. In order to find the decision boundary for non-linear problems, SVM transforms the input data to suitable higher-dimensional space. However, this strategy poses a problem of substantial computational overhead. SVM fixes this by the use of kernel functions to systematically find classifiers in the higher dimension. Kernel function (K) calculate the relationship between every pair of points as if they are in a higher dimension without actually transforming them to a higher dimension. This phenomenon is known as the kernel trick. In this way, the kernel trick helps reduce the amount of computation required for SVM. Table 3.2 gives some examples of kernel function and their formulas.

Table 3.2 kernel functions

Kernel name	Kernel function
Linear	$K(v_i, v_j) = v_i^T \cdot v_j$
Gaussian	$K(v_i, v_j) = e^{\left(\frac{-\ v_i - v_j\ ^2}{2\sigma^2}\right)}$
Polynomial	$K(v_i, v_j) = (1 + v_i^T v_j)^d$

In Table 3.2, σ represents the Gaussian noise, and d represents the order of the polynomial. The Gaussian kernel is arguably the best because of its computational efficiency, flexibility, and adaptability in handling complex parameters [134-136]. This is achieved by the use of a set of linear equations for its training instead of the computationally tasking quadratic programming problem used by the others.

This implies that for non-linear boundaries, (3.39) becomes (3.42).

$$L_p(w, b, \alpha) = \sum_{i=0}^{n-1} \alpha_i - \frac{1}{2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \alpha_i \alpha_j y_i y_j K(v_i, v_j) \quad (3.42)$$

Also, (3.40) becomes (3.43).

$$b^* = -\frac{1}{2} \left(\min_{i:y_j=+1} \left(\sum_{j=0}^{n-1} \alpha_j y_i K(v_i, v_j) \right) + \max_{i:y_j=-1} \left(\sum_{j=0}^{n-1} \alpha_j y_j K(v_i, v_j) \right) \right) \quad (3.43)$$

Furthermore, (3.41) becomes (3.44).

$$f(x) = \sum_{j=0}^{n-1} \alpha_j y_j K(v_i, v_j) + b^* \quad (3.44)$$

3.2.3 K – nearest neighbour

k-nearest neighbour (KNN) is a fundamental and straightforward technique for data classification. It uses the distance of an object within a feature space to classify it. Objects that are close together (or nearest neighbour) are classified together. KNN is typically referred to as a ‘lazy learner’ because it does not learn a discriminatory function from the training set. It is a memory-based classifier and typically will not undergo a training phase [137]. The “K” in KNN is a parameter that represents the number of nearest neighbours used by the classify for classifying data. Low values of K can be noisy and easily affected by outliers. Large values of K will result in higher computational overhead as each distance must be calculated. Figure 3.9 shows an illustration of KNN; when $K = 3$, the star belongs to class 2 (2 to 1 vote), but when $K = 5$, the star belongs to class 1 (3 to 2 vote).

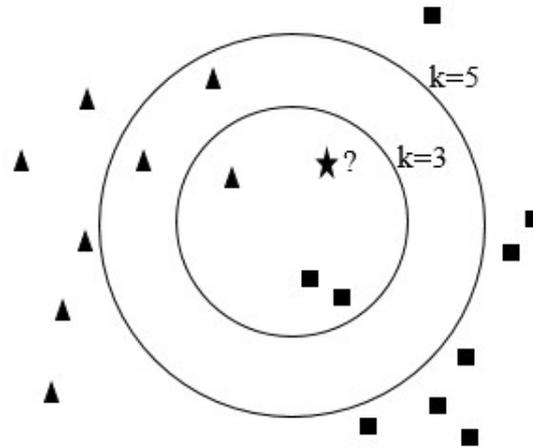


Figure 3.9 KNN algorithm

In order to find an appropriate value of K, a trial-and-error approach can be used. In this approach, a certain amount of the known dataset is randomly set aside for reclassification; the value of K that gives the best result is selected. The value for K chosen should be odd and not a multiple of the number of classes to avoid ties during voting. The proximity metrics d used for decision-making in KNN is generally the Euclidean distance [138]. However, others like Hamming distance, Manhattan or city block distance, Canberra distance, Minkowsky distance, and Chebychev distance can also be used (see Table 3.3).

Table 3.3 Proximity metrics

Proximity metrics	Formula
Euclidean	$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
Canberra	$d(x, y) = \sum_{i=1}^n \frac{ x_i - y_i }{ x_i + y_i }$
Chebychev	$d(x, y) = \max_i \{ x_i - y_i \}$
Manhattan/ City block	$d(x, y) = \sum_{i=1}^n x_i - y_i $
Minkowsky	$d(x, y) = \left(\sum_{i=1}^n x_i - y_i ^r \right)^{1/r}$

The Minkowsky distance has a value of r that can range from 1 to ∞ . If $r = 1$, the Minkowsky distance resembles the Manhattan distance. If $r = 2$, the Minkowsky distance resembles the Euclidean distance. If $r = \text{infinity}$, the Minkowsky distance resembles the Chebychev distance.

3.2.4 Artificial neural network

Artificial neural networks (ANN) are inspired by the biological nervous system [139]. It had its origins in the field of computing and neuroscience [140]. It is a layered arrangement of neurons that can be trained to predict an output from inputs based on the training data. Neural networks take in data, train on the data to recognise the pattern in the data, and then predict the output for a new set of similar data based on the earlier recognised pattern. Figure 3.10 shows the structure of an ANN.

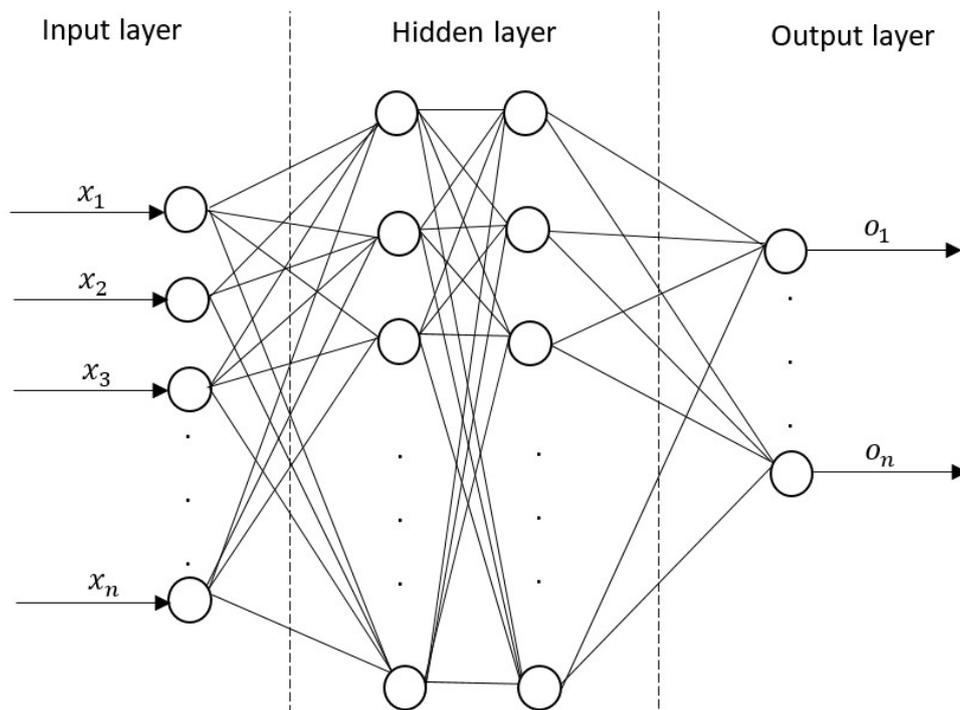


Figure 3.10 ANN structure

The network is divided into three parts; the input, the hidden layer, and the output [141]. The input layer takes in the input data, and the output layer predicts the output. The hidden layer is where all the computation in the network takes place. All of the three layers are made up of

neurons that are interconnected. Each of the connections has a weight value w while each neuron in the hidden layer has a bias value b . The ANN functions by multiplying the input with a weight w and adding it a bias b , then the sum is tested against a threshold to determine if the neuron is activated or not. The thresholding function is known as a transfer function or activation function. The most commonly used transfer function is the sigmoid given by:

$$z = \frac{1}{1 + e^{-(\sum_{i=0}^n w_i x_i + b)}} \quad (3.45)$$

Where z is the output, w_i is the weight corresponding to the input x_i , n is the number of inputs.

Figure 3.11 shows the neuron model structure from input through the hidden layer to the output

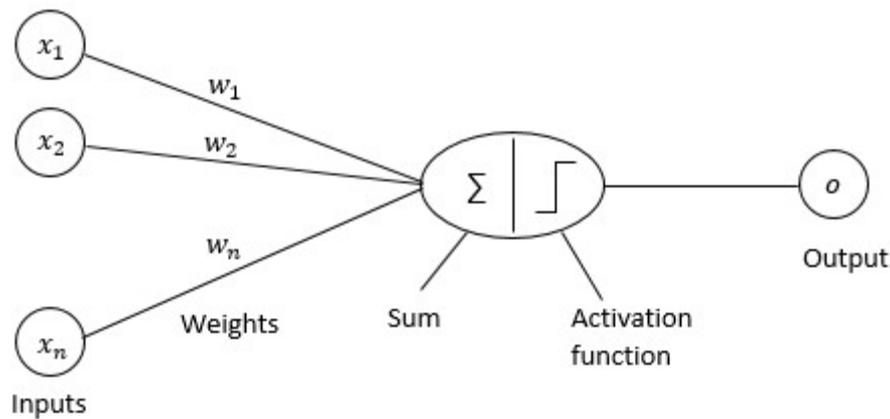


Figure 3.11 Neuron model structure

There are many other activation functions like the step function, tanh, rectified linear unit (ReLU), and linear. Typically, the initial weights and biases are selected at random. The process of fine-tuning the weights and biases such that the network can make an accurate mapping of the input to the output is called training [142]. In order to train an ANN, the network is feed with inputs of known outputs. After every iteration of the network's training, the weights and biases are adjusted based on the accuracy of the prediction. This process is continued until a predetermined condition is met.

Generally, the available data is split into two: the training set and the testing set. The training set is used to train the network, while the testing set is used to access the network's performance [143]. A network is said to have good generalisation if it performs well on the training data.

However, a network could perform well on the training data and poorly on the test data. This phenomenon is caused by over-fitting or over-training. Over-fitting occurs when the network is trained to perfectly model the training data such that its accuracy on any other similar but previously unseen data is poor. A subset of the training data (called validation data) is used to determine when training should stop to prevent over-fitting. During training, the error in the training data reduces, the error in the validation data will also reduce. Until a point is reached when the error in the validation data starts to rise while the error in the test data continues to fall. This is the point where training is stopped, as shown in Figure 3.12.

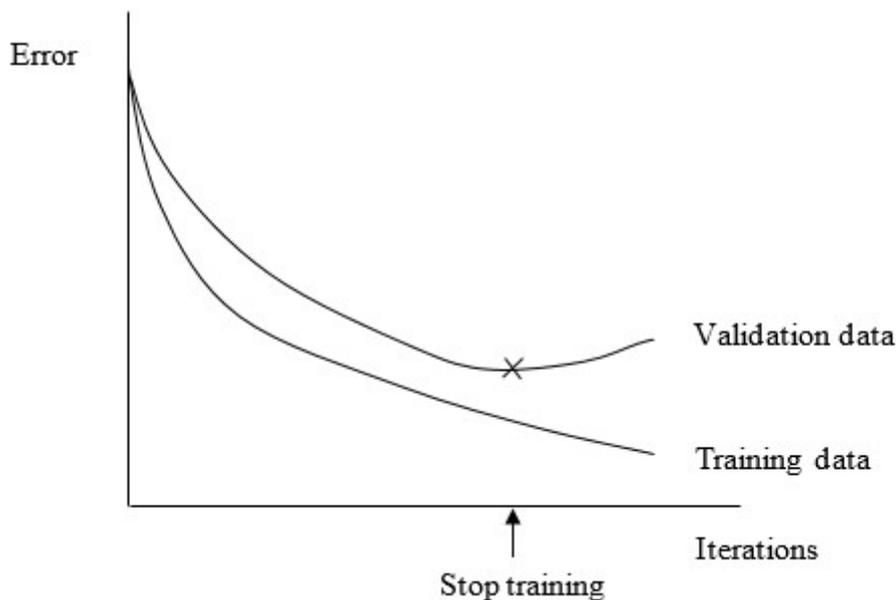


Figure 3.12 Training and validation error during training in ANN

Two critical functions affect training in ANN; they are learning rate and momentum. The learning rate is how much the output error affects the network's internal weights, while momentum determines how much the previous weight affects the current weights [139, 142].

$$\Delta w = \eta \delta x \quad (3.46)$$

$$(\Delta w)_{n+1} = \eta \delta x + \alpha (\Delta w)_k \quad (3.47)$$

The learning rate and momentum values are selected through a trial and error approach [139]. Learning too slowly takes too much time to train the network; learning too fast increases errors in network predictions.

3.3 Feature extraction

In order to use machine learning effectively, it is likely that feature extraction has to occur first. While binary, categorical, or discrete data may not require feature extraction, reviewed literature indicates that continuous data benefits from feature extraction [144]. This is because it helps reduce the dimensionality of the data by eliminating redundant ones and reduces the computational cost of processing the data. Feature extraction is particularly required for this research because of the cyclic nature of the continuous vibration data produced by the rotating ballnuts in the ball screw feed drive assembly when the machine is in operation. For such a signal, it is necessary to access information from the last n number of rotations rather than just instantaneous values. Features that numerically summarise the current and recent behaviour of the system can be extracted for analysis. Machine learning using such a feature is expected to give better results from learning this representation of the cyclic data than an instantaneous point in time, which will be less relevant.

3.3.1 Types of extracted features

Feature extraction is a process of extracting valuable information that is representative of the initial raw continuous data and presenting it to the machine learning algorithm as inputs. Some of the commonly used features found in literature are mean (\bar{x}), root mean square (β), standard deviation (σ), kurtosis (K), skewness (S), and peak to peak (P) [145]. The review of the literature showed that most authors do not specify why any particular feature is chosen and are often simply provided as a menu of options. However, in the context of the research problem in this thesis, each is being considered for the reason presented below.

3.3.1.1 Mean

The mean (average) of a continuous data signal is expressed as the summation of all the individual data points divided by the size of the data [146]. The mean is calculated as shown [75, 147]:

$$\bar{x} = \frac{1}{T} \int_0^T x(t) dt = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.48)$$

Where n is the number of samples, T is periodic time, x is the dataset, and $i = 1, 2, \dots, n$.

The mean is often used because it is relatively easy to calculate and gives a value that is representative of the centre of gravity of a given dataset. For a pure sine wave, centred around zero, the mean value is zero for every complete cycle. However, an irregular waveform, or one composed of several sine waves with a different mean value, will likely have spurious data points and altered symmetry such that the mean moves below or above zero. Even where a non-zero mean exists on a healthy system, the change in value can be an indication of a change in condition. For the ball screw system, it is anticipated that a change in condition would cause superposition of cyclic features and inclusion of random artefacts that would cause the mean value to deviate from the benchmark (baseline) value. However, it is clear that if a change in condition does not increase the amplitude of the vibration, then the mean value would not alter, so it would not provide a useful indicator.

3.3.1.2 Standard deviation

Standard deviation gives the degree of variation of the data points within a signal. The standard deviation is calculated as shown [75, 147]:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (3.49)$$

Where n is the number of samples, \bar{x} is the mean, and $i = 1, 2, \dots, n$.

Whereby a high value for the standard deviation is indicative of a highly disperse signal while a low value means a less disperse signal [148]. This increase in dispersion to a signal is brought about by the additional superimposed content as well as noise. This could potentially cause the signal from a worn ball screw to have a higher standard deviation than that from a good ball screw, especially if the superimposed signal or noise has frequencies with significantly high peaks compared to those of the original signal. However, if this is not the case (particularly with noise signals that are often high-frequency signals with low magnitude), then the standard deviation value would not provide a useful indicator.

3.3.1.3 Kurtosis

Kurtosis is a measure of the degree of outliers or spurious data points within a signal [146, 149]. The kurtosis is calculated as shown [75, 147]:

$$K = \frac{1}{\beta^4 T} \int_0^T x^4(t) dt \quad (3.50)$$

$$= \left\{ \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{\sigma} \right)^4 \right\} - \frac{3(n-1)^2}{(n-2)(n-3)}$$

Where n is the number of samples, T is periodic time, x is the dataset, \bar{x} is the mean, σ is the standard deviation, β is the RMS value, and $i = 1, 2, \dots, n$.

A good ball screw is expected to produce regular repeating cyclic signals if the speed is nominally constant. However, a fault on the ball screw or ballnut will cause interruption to the regularity of the signal. These interruptions are likely to be represented as spurious data points compared to the original signal. This implies that kurtosis could be able to differentiate faulty from good ball screws with high values indicating the former and low values indicating the latter. This parameter is therefore chosen for the investigation to pick up any non-cyclic faults.

3.3.1.4 Skewness

Skewness is a measure of the degree of asymmetry in a dataset [146, 149]. The skewness is calculated as shown [75, 147]:

$$S = \frac{1}{\beta^3 T} \int_0^T x^3(t) dt = \frac{n}{(n-1)(n-2)} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{\sigma} \right)^3 \quad (3.51)$$

Where n is the number of samples, T is periodic time, x is the dataset, \bar{x} is the mean, σ is the standard deviation, β is the RMS value, and $i = 1, 2, \dots, n$.

Kurtosis and skewness have been shown to effectively indicate the accurate position of epileptogenic zones from magnetoencephalography [149]. Marcell et al. were able to show that skewness features accurately indicate fatigue damage in random loading processes [150]. Furthermore, they combined the two features (skewness and kurtosis) to achieve improved results in each case. Even though the particular applications are not the same, there is reason to investigate skewness for ball screw systems. As with kurtosis, the degree of asymmetry resulting from the motion of a faulty ball screw drive is expected to be more compared to that from a good ball screw in situations where there is a mechanical bias.

3.3.1.5 Root mean square

The root mean square (RMS) value can be used to quantify dynamic signals so as to distinguish one from another [151]. It is the square root of the mean square averaged over T second and is calculated as shown [75, 147]:

$$\beta = \sqrt{\frac{1}{T} \int_0^T x^2(t) dt} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \quad (3.52)$$

Where n is the number of samples, T is periodic time, x is the data value under consideration, and $i = 1, 2, \dots, n$.

The RMS value is especially relevant in vibration applications as it indicates the energy content of the vibration signal. It is expected that a change in condition will change the energy in the ball screw feed drive. For example, lack of lubrication or change in its viscosity means that the friction changes, requiring different energy levels to produce the same linear motion. Similarly, misalignment or eccentricity means that energy that should have gone into linear motion is instead expended on translation vibrations, which are measured by the vibration transducer; more misalignment leads to more energy spent in this way. This will result in superimposed content as well as noise being introduced to the original signal. Therefore, RMS is a good candidate for detecting this additional signal content via its energy level.

3.3.1.6 Peak-to-peak value

The peak-to-peak value is the difference between the maximum (positive peak) and the minimum value (negative peak) of a signal. It is calculated as shown [75]:

$$P = X_{max} - X_{min} \quad (3.53)$$

Where X_{max} and X_{min} represents the highest and lowest values of the data, respectively.

This calculation provides a raw value for the maximum amplitude without the attenuation that comes from processing through standard deviation and RMS. However, this feature heavily depends on just two points per cycle, while the standard deviation and RMS use all the points, making it less sensitive to measurement noise. Therefore, categorising based on this feature is likely to emphasise changes in the amount of energy from the ball screw drive being measured.

3.3.2 Combination of features

All the features discussed in section 3.3.1 have their strengths and weaknesses in terms of how well they represent signals. Hence the reason why in most cases, multiple features are used to train machine learning networks. Xiang et al. [149] and Marcell et al. [150] have shown that skewness and kurtosis complement each other well to achieve improved results compared to their individual performance. This is reasonable because those two features measure the

deviation in a signal from a regular repeating pattern. However, as the papers show, in some instances, these deviations are not evident in the value of the skewness or kurtosis alone. That is why their individual performance is not as good as when they are combined. It is also possible that even in their combination, some instances still occur where their value does not show a deviation from a regular repeating pattern. Since the mean, RMS, standard deviation, and peak-to-peak values all measure some form of deviation of a signal from a regular repeating pattern, there is potential for improved results if these features are also combined with skewness and kurtosis. This research will investigate how Xiang's and Marcell's proposal for the combination of skewness and kurtosis features perform with vibration data from ball screw. This will lead to investigating whether combining all the features from section 3.3.1 can provide further improvements.

3.3.3 Choice of signal processing for feature extraction

Feature extraction can be performed on the raw data [152], the output of the FFT of the raw data [153, 154], or the output of the wavelet of the raw data [75]. However, the past research that describes these operations seems to focus more on the improvement and optimisation of the machine learning algorithm and not on the critical review of the feature extraction process, which serves as the input to these algorithms. Ideally, developing an efficient and highly effective machine learning strategy should necessarily involve both the improvement and optimisation of the machine learning algorithm and the feature extraction method.

This research proposes a feature extraction method that combines wavelet and FFT to process the raw data before feature extraction occurs. This procedure combines the positives of both wavelet transform and FFT as described in section 3.1. They are: (1) the temporal localisation of data by wavelet transforms and (2) the conversion of the data into spectral components by FFT. These two pre-processing methods combined with the statistical features described above were explored in this research, and the results will be presented in chapter 6. The different sets

of extracted features were used to train and test different machine learning algorithms. However, the question remains as to which is the most appropriate wavelet transform to use for the application of monitoring ball screw condition since there are many different wavelet families with different characteristics and many different levels of decomposition that can be achieved on each wavelet. One solution, which is difficult to justify, is to select the wavelet transform randomly in terms of the family and decomposition level. It is likely that a more successful strategy is to perform an analysis on a subset of wavelet transforms based on the type of data and characteristics of the wavelet in order to assess which performs best.

Literature was examined to find what approaches had been used by other researchers. In some cases, the authors simply do not specify the wavelet transform they have used [124, 155]. In other cases, a wavelet transform is specified, but there is no justification given for such specification. For example, Daubechies wavelet [75] and Haar wavelet [156] were used by Hong et al. and Lee respectively in research literature without any discussion on why they have been used. Of the twenty papers reviewed on wavelet analysis, six (30 %) stated that a particular form of wavelet transform is used without specifying which one, seven (35 %) specified which wavelet transform it has utilised without explicitly justifying the choice. The remaining seven (35 %) did both specify and justify their choice of wavelet transform. However, only four (20 %) can be characterized as justification for technical reasons driven by the problem domain. The other three (15 %) only referring to usage by other researchers as justification and thus possibly simply promulgating an earlier error.

This research will more thoroughly perform an analysis to justify the selected wavelet transform for the given problem domain using matrix laboratory (MATLAB) software as a tool to achieve this. The two broad wavelet classifications of orthogonal (Symlet and Daubechies) and biorthogonal (biorthogonal and reverse-biorthogonal) wavelet will be considered even though studies suggest that for this particular application, the biorthogonal wavelets can be

discounted. This is because the biorthogonal wavelets are more suited for image processing, while the orthogonal wavelet is more suitable for processing continuous signals [157]. However, research also shows that many different wavelets, regardless of the classification, can be good for various applications, and no particular wavelet can be considered “the best” for any problem [158]. Daubechies [159] advocates that the best method for selecting wavelets is by comparative tests and validation. This analysis will be performed and presented in chapter 6, with a final selection made from the best performing set based on robustness to machine learning and the current problem domain. However, it is worth noting that “the best performing set” might be different for another problem domain. The flow diagram of the proposed methodology is shown in Figure 3.13.

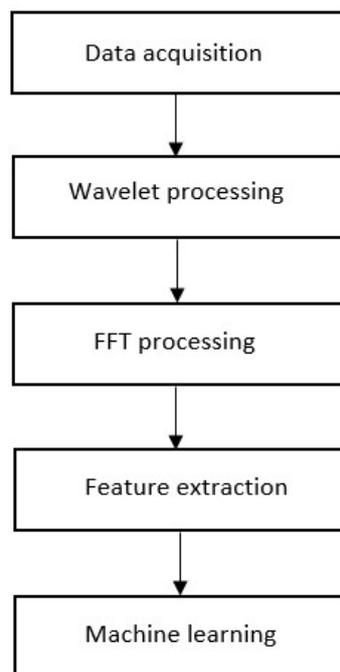


Figure 3.13 Flow diagram of the proposed methodology

3.4 Chapter summary

Machine learning is one of the AI techniques utilised for condition monitoring; some examples of machine learning include decision tree, SVM, KNN, and ANN. One major characteristic of machine learning is that it requires features extracted from the raw data to train the network,

especially if the raw data is continuous. This necessarily requires some form of signal process to be able to identify useful features. This research proposes using wavelet transform and FFT for signal processing and feature identification such that effective feature extraction can be done. The extracted feature is then trained and evaluated with machine learning. Performance is based on how well the proposed strategy can identify the condition of the ball screw from the extracted features. The next chapter delves into deep learning, activation functions, and deep learning models.

Chapter 4 Deep learning

This chapter presents a detailed analysis of deep learning by providing a general overview of the topic. Then a comprehensive look at the internal structure and activation functions used in a typical deep learning algorithm. This is followed by a discussion of the types and applications of some of the most commonly used deep learning algorithms for accurate predictive models. This chapter highlights all the essential concepts that make machine learning achievable for the process of condition monitoring. Materials used for this chapter include journals, conference papers, and textbooks.

4.1 Deep learning overview

In the establish problem area where the particular fault and the internal mechanics of the system is unknown, even though machine learning is able to provide an analytical method that deskills some activities in the classical approach. It still retains the involvement of an expert agent to design feature extraction which could be prone to biases and preconceptions. This also reduces its ease of adaptation for real-time online analysis. The above-mentioned challenges can be resolved with deep learning since it reduces the involvement of an expert agent as the feature extraction occurs with the deep learning network. Hence, making it easier to adapt the deep learning model for real-time online analysis.

Deep learning is a subset of machine learning, and machine learning is a subset of AI; this is illustrated in Figure 4.1. AI is the ability of a robot to replicate intelligent human behaviour. Machine learning is an application of AI in which a predictive system can learn automatically and improve accuracy based on the input data and corresponding output. On the other hand, deep learning is an application of machine learning that uses a complex algorithm with multiple hidden layers to train a model. It learns hierarchical structures, levels of representations, and levels of abstractions in order to be able to map the patterns in a dataset to predetermined

outputs. It differs from machine learning in that it does not require independent feature extraction. Rather it performs automatic feature representation at multiple levels of abstraction, which is then used for training the network [160].

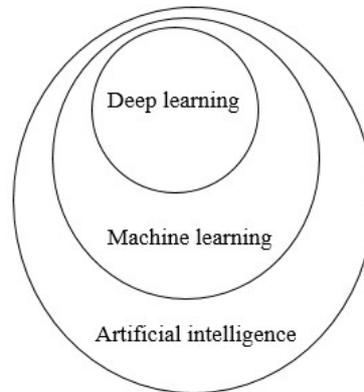


Figure 4.1 Artificial intelligence hierarchy

This allows the deep learning network to perform complex mapping of an input space to the corresponding output space without the intervention of an expert agent. Deep learning networks are data-driven; hence their performance is much dependent on the quality and quantity of data.

Figure 4.2 shows a block diagram of a deep learning network.

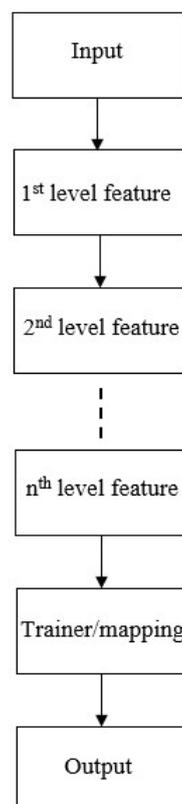


Figure 4.2 Block diagram of a general deep learning network

4.2 Structure of deep learning algorithms

Generally, a deep learning algorithm is divided into three layers, similar to machine learning. These layers include; the input layer, the output layer, and the hidden layer. However, the hidden layer in deep learning is more complex, with multiple hidden layers at a time. Whereas machine learning typically has one or two hidden layers with multiple neurons.

4.2.1 Input layer

The input layer is the layer that receives the raw input data. This data can be in any form, like image data in the form of pixel value, time-series data in the form of continuous numbers, and text data in the form of words. The number of input neurons is determined by the defined size of the input parameter.

4.2.2 Output layer

The output layer is the layer of the network that presents the output value. The output value is typically presented as a classification or a regression based on the initial problem setting. The number of neurons in the output layer is defined by the number of outputs expected. The output layer uses a softmax function to compute the confidence level of each output. The confidence level is the probability of the output belonging to each class. The highest probability determines the output class. For an i th training example out of the total training example m and the j th class out of the total number of classes n , the softmax function is given by the probabilistic expression

$$P(y^{(i)} = n | x^{(i)}; W) = \begin{bmatrix} P(y^{(i)} = 1 | x^{(i)}; W) \\ P(y^{(i)} = 2 | x^{(i)}; W) \\ \vdots \\ P(y^{(i)} = n | x^{(i)}; W) \end{bmatrix} = \frac{1}{\sum_{j=1}^n e^{W_j^T x^{(i)}}} \begin{bmatrix} e^{W_1^T x^{(i)}} \\ e^{W_2^T x^{(i)}} \\ \vdots \\ e^{W_n^T x^{(i)}} \end{bmatrix} \quad (4.1)$$

Where x is the input, y is the output, W is the weight, and $i = 1, 2, \dots, m$.

4.2.3 Hidden layer

The hidden layer is the layer between the input and output layers. It consists of multiple layers performing different mathematical functions. These layers include the convolution layer, activation layer, pooling layer, fully connected layer, sequence layer, normalisation layer, and dropout layer. Each of these layers may belong to one or more hidden layers. Each hidden layer is referred to as a class and performs a task within the network [160].

4.2.3.1 Convolution layer

It is the core building block of any convolution neural network (CNN). It functions by applying sliding convolutional filters to the input data. This is achieved by performing a dot product multiplication between a subarray of the input and a receptive field (convolution filters) with randomised weights. The size of the output reduces after each step of convolution and is given by [102]

$$O = \frac{I - R}{S} + 1 \quad (4.2)$$

Where O is the output size, I is the input size, R is the filter size, and S is the stride size. The optimum values for S , R , and weights of the filter are obtained after training the network. Figure 4.3 shows an example of convolution.

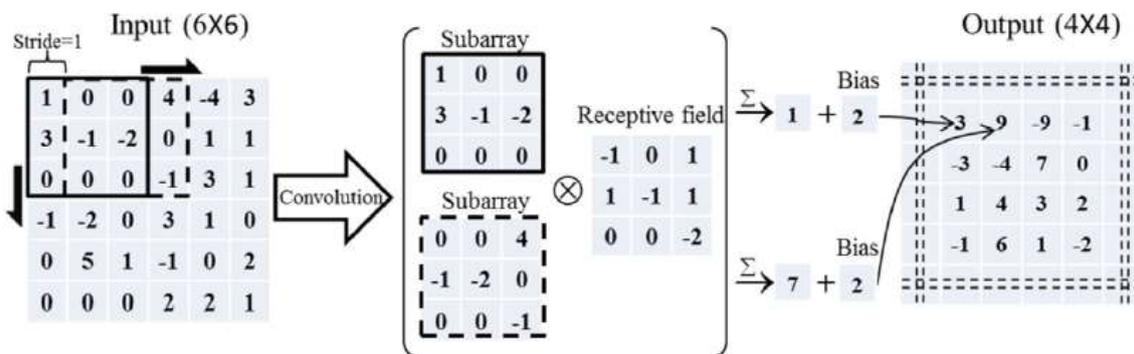


Figure 4.3 An illustration of the Convolution process [102]

The stride defines the amount of shift on the input data required to obtain the next subarray. The larger the stride, the smaller the output size. For an input size of 6×6 , the output size is 4×4 when the stride is 1, and 2×2 when the stride is 2.

4.2.3.2 Pooling layer

The pooling layer typically comes after the convolution layer in CNNs. It performs progressive down-sampling of its input along both spatial dimensions of height and width. It does this by dividing its input into pooling regions. The down-sampling can be achieved by computing the maximum or mean value in each pooling region. Hence the two types of pooling layers are; max-pooling layer and average-pooling layer, respectively. Max-pooling generally gives better accuracy than average-pooling in practical cases and is more widely used [1,3]. The size of the output of the pooling layer is given by [102]

$$O = \frac{I - P}{S} + 1 \quad (4.3)$$

Where O is the output size, I is the input size, and S is the stride size. The optimum values for S and P are obtained after training the network. Figure 4.4 shows an example of pooling.

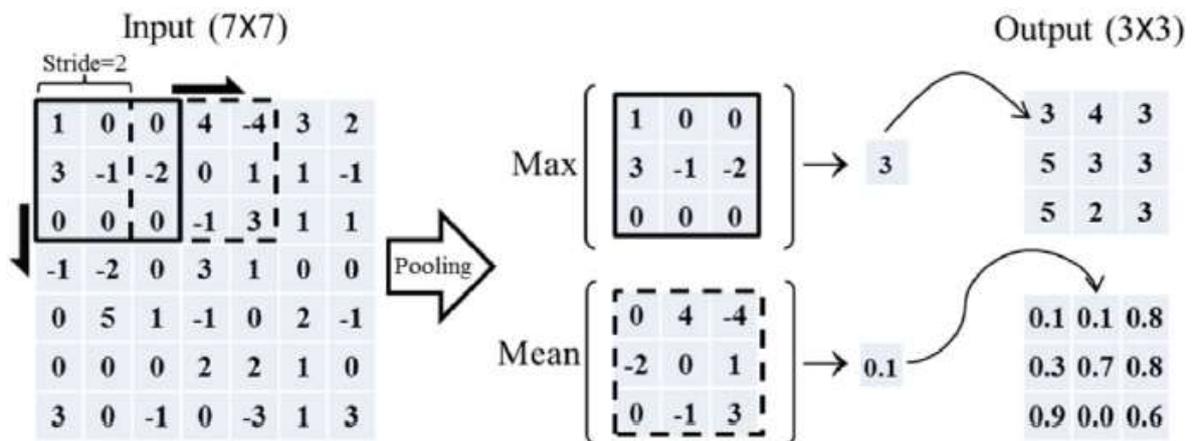


Figure 4.4 An illustration of the Pooling process [102]

The fully connected layer (or dense layer) is often the final layer in the hidden layer of CNNs. It fully connects all activations in its input neurons to the output neurons. In each of the connections, inputs are multiplied by a certain weight and then added to a bias value. Both values (weight and bias) are determined during training. The output of this layer is typically the class score, and the number of classes determines the number of neurons.

4.2.3.3 Sequence layer

The sequence layer is the core building block of recurrent neural networks (RNN) or long short-term memory (LSTM), as it is sometimes referred to as. Its function is to learn long-term relationships between time steps and the time series in continuous data. The fully connected layer typically follows this layer in an LSTM network.

4.2.3.4 Dropout layer

The dropout layer in deep learning networks is used to overcome overfitting and long training time [161]. It does this by dropping all units along with all of its inbound and outbound connections based on a probability criterion. If the probability associated with a particular unit is lower than the set threshold p , then that unit and all of its connections are dropped. This results in a thinner network, as shown in Figure 4.5. The value of p can be selected using a validation set or set at 0.5, as this is often close to the optimum value for most networks [161].

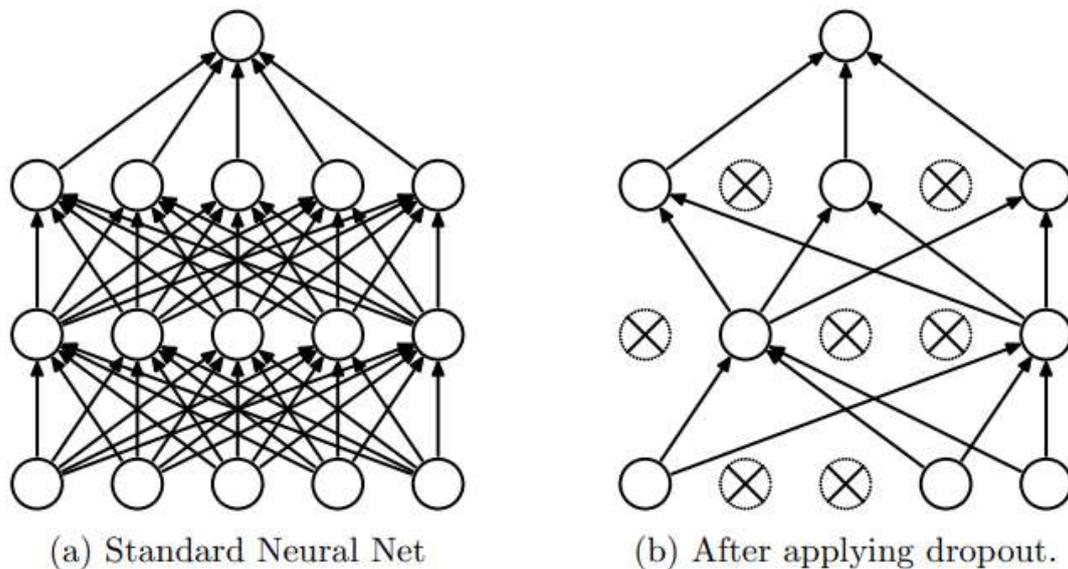


Figure 4.5 Example of before and after dropout [161]

The model description for a standard neural network, as shown in Figure 4.5, is given by;

$$z_i^{(l+1)} = W_i^{(l+1)}y^{(l)} + b_i^{(l+1)} \quad (4.4)$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \quad (4.5)$$

For $l \in \{0, \dots, L - 1\}$ and any hidden unit i , $W^{(l)}$ and $b^{(l)}$ are the weights and biases at layer l . Where L is the number of hidden layers, z is the input, y is the output, and f is an activation function. With dropouts as illustrated in Figure 4.5, the expression becomes [161];

$$r_j^{(l)} \sim \text{Bernoulli}(p) \quad (4.6)$$

$$\tilde{y}^{(l)} = r^{(l)} * y^{(l)} \quad (4.7)$$

$$z_i^{(l+1)} = W_i^{(l+1)} \tilde{y}^{(l)} + b_i^{(l+1)} \quad (4.8)$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \quad (4.9)$$

Where $*$ represents the dot product, $r^{(l)}$ is a vector of independent random Bernoulli variable, each with a probability of p in layer l and $\tilde{y}^{(l)}$ is the thinned output from layer l .

4.2.3.5 Normalisation layer

The normalisation layer typically comes after the rectified linear unit (ReLU) activation function in deep learning networks. Even though ReLUs do not need normalisation in order to avoid saturation. However, they are widely used because they improve network generalisation. The model description for the activity in the normalisation layer is given by the expression [93];

$$b_{x,y}^i = \frac{a_{x,y}^i}{\left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^j)^2\right)^\beta} \quad (4.10)$$

Where $b_{x,y}^i$ represents the normalised response, $a_{x,y}^i$ represents the computed neuron activity of kernel i at position (x, y) , N is the amount of kernel and n is the number of kernel maps at the same spatial location. The constants k , n , α , and β are hyper-parameters whose values are 2, 5, 10^{-4} , and 0.75. These values are defined during training with the help of a validation set [93].

4.2.3.6 Activation layer

The activation layer uses the activation function to determine which neuron is turned on or off during training to achieve the best network generalisation and accuracy. More information about the activation layer is described in section 4.3.

4.3 Activation function

The activation function σ is used to perform thresholding operations on the input elements x such that all input values less than the set threshold are set to m , and those above the threshold are set to p . where m and p are determined based on the particular activation function used. A critical characteristic of an activation function is that it should have a derivative σ' . This is because neural networks use errors and backpropagation to adjust the weights and biases during training. These weights and biases are adjusted at a “rate of change,” depending on the change in error for each training iteration [160]. There is potentially an infinite number of activation functions; however, some of the widely used ones in research literature include step, sigmoid, hyperbolic (tanh), and ReLU.

4.3.1 Step function

The step function is an activation function that performs a threshold operation on each input element such that values that are less than zero are set to zero, and all other values are set to one. The step function is given by the expression;

$$\sigma(x) = \begin{cases} 0, & n < 0, \\ 1, & n \geq 0, \end{cases} \quad (4.11)$$

$$\sigma'(x) = \begin{cases} 0, & n \neq 0, \\ ?, & n = 0, \end{cases} \quad (4.12)$$

Furthermore, the graphical representation is shown in Figure 4.6.

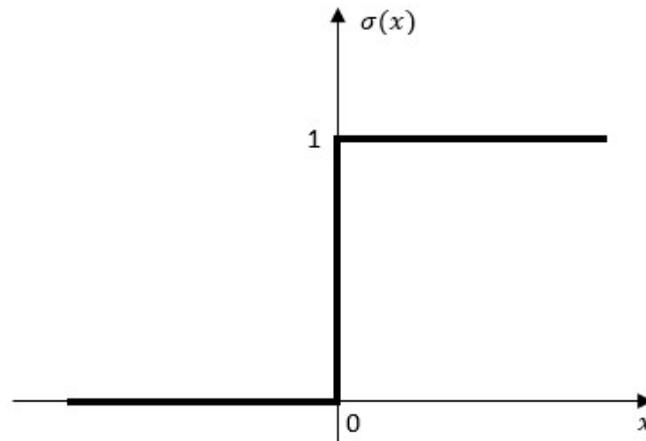


Figure 4.6 Plot of Step activation function

4.3.2 Sigmoid function

The sigmoid function performs threshold operation on input elements such that their corresponding output elements range from 0 to 1. The expression for the sigmoid function is given by;

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4.13)$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (4.14)$$

Furthermore, the graphical representation is shown in Figure 4.7.

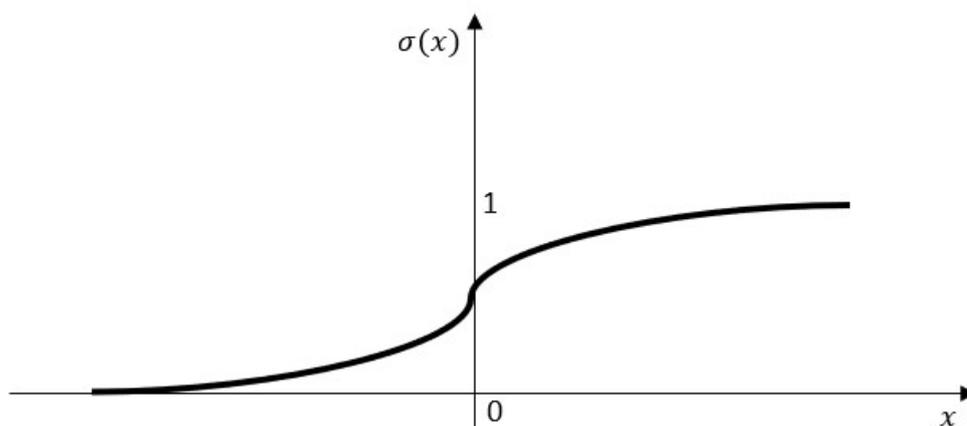


Figure 4.7 Plot of Sigmoid activation function

4.3.3 Hyperbolic function

The hyperbolic function performs threshold operation on input elements such that their corresponding output elements range from -1 to 1. The expression for the hyperbolic function is given by;

$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.15)$$

$$\sigma'(x) = 1 - \sigma(x)^2 \quad (4.16)$$

And the graphical representation is shown in Figure 4.8

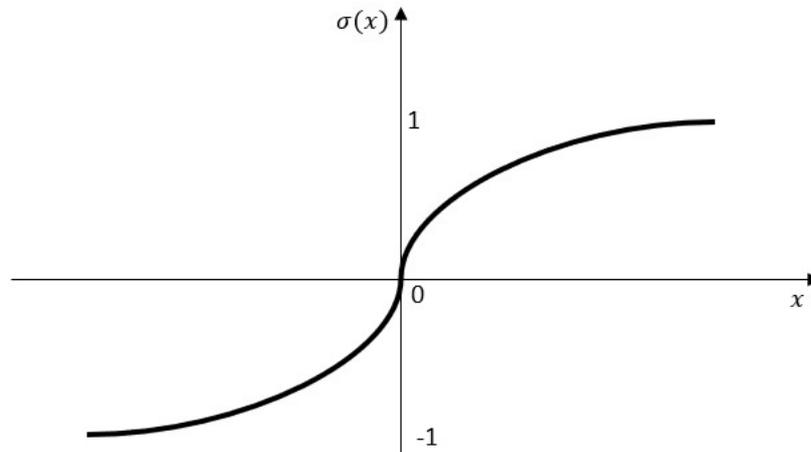


Figure 4.8 Plot of Hyperbolic activation function

4.3.4 ReLU function

The ReLU function is an activation function that performs a threshold operation on each input element such that any value less than zero is set to zero and all other values remain constant. It is computationally efficient and simple. This is because it has a linear and non-saturated form that converges six times faster than most other activation functions [93], making it commonly used for deep learning networks. The ReLU function is given by the expression;

$$\sigma(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (4.17)$$

$$\sigma'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (4.18)$$

And the graphical representation is shown in Figure 4.9

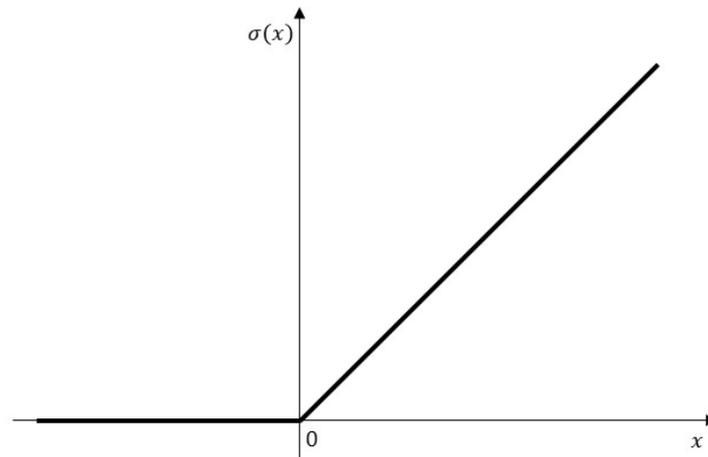


Figure 4.9 Plot of ReLU activation function

4.4 Deep learning models

As deep learning becomes increasingly popular, many different models exist. However, some of the more commonly used versions include convolution neural network (CNN), deep-belief network (DBN), and recurrent neural network (RNN) [1,7].

4.4.1 Recurrent neural network

In the recurrent neural network, unlike other feed-forward neural networks, the information path progresses through a series of mathematical operations performed on the nodes with feedback loops. Hence, the output will be dependent on the order of the input signal. This implies that this type of network is suitable for handling sequential input or time-series data. This network type is used for applications that require time-series data like stock market price prediction, text prediction, speech recognition [162, 163]. In these examples, it is not only the order of the data sequence that matters; the future values are also significantly correlated to the recent past values. Hence, for an RNN, the current output is determined by the current input and the previous output. Figure 4.10 shows the structure of an RNN; it is a chain of repeating cells performing their function in the time dimension. Where x , h , \tanh , $t - 1$, t and $t + 1$ are the input signal, output signal, activation function, past time, current time, and future time respectively. The number of cells or layers of an RNN is determined by the time step's length

under consideration [160]. The RNN uses the same backpropagation method as other feedforward networks for adjusting the weights of the network's internal nodes based on its error during training. However, in this case, the process is referred to as backpropagation through time (BPTT) since the series of layers in a feedforward network has been swapped with a series of similar cells in a timeline.

The issue with RNN is that it is affected by vanishing gradient [93, 160]. Vanishing gradient is a phenomenon where if a network is significantly deep, a situation arises where the partial derivative of the error with respect to the weight during backpropagation in certain layers is so small that it does not affect the previous weights. This problem was resolved by long short-term memory (LSTM), a variation of the RNN. This is achieved by utilising memory blocks in their recurrent connection to store the network's temporal state. Meaning, the network does not forget its initial inputs when new ones are available. Hence, for an LSTM network, the current output is determined by the current input, the previous output, and the previous state. The structure of the cell of an LSTM network is shown in Figure 4.11. Each cell comprises the forget gate, the input gate, and the output gate.

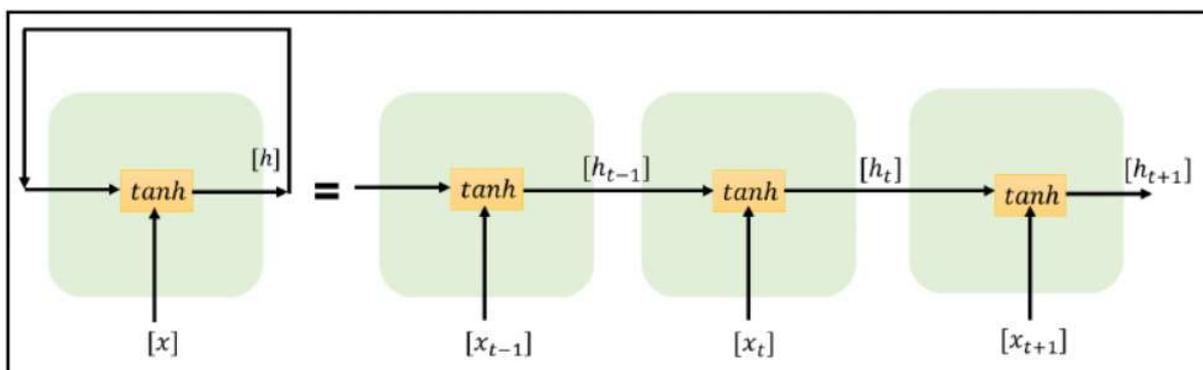


Figure 4.10 RNN structure [160]

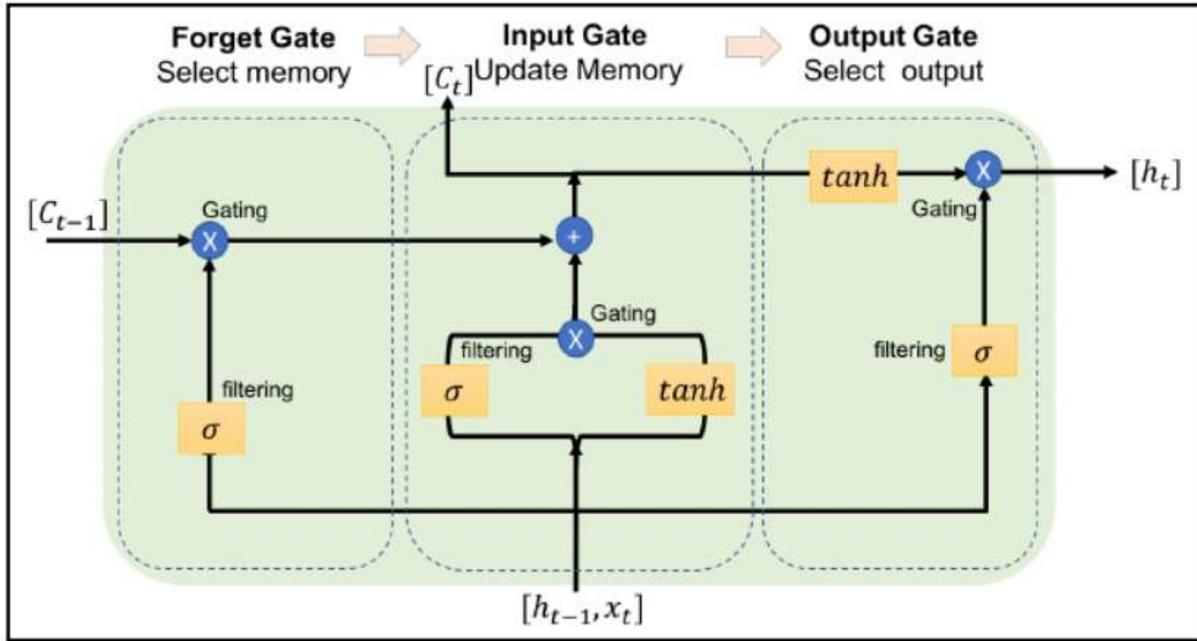


Figure 4.11 An LSTM network with three gates [160]

The forget gate, also known as the select memory gate, is responsible for deciding which part of the previous output should feedback to influence the current input. The input gate, also known as the update memory gate, is responsible for updating the current memory value by adding the previous output to the current input. It uses an activation function, like the \tanh to select which input value to update. The output gate, also known as the select gate, is responsible for making three decisions: what part of the current cell state will be hidden, what part will be feedback back to the forget gate, and what part will be set as the output.

The learning process of an LSTM is calculated at every time step t between 1 and τ , and the neuron parameter at layer l is given by the following equations [88];

$$a_1^{(t)} = b_1 + W_1 \cdot C_1^{(t-1)} + U_1 \cdot x^{(t)} \quad (4.19)$$

$$C_l^{(t)} = \tanh(a_1^t) \quad (4.20)$$

$$a_l^{(t)} = b_l + W_l \cdot C_l^{(t-1)} + U_l \cdot C_{l-1}^{(t)} \quad (4.21)$$

$$h^{(t)} = b_N + W_N \cdot C_N^{(t-1)} + U_N \cdot C_N^{(t)} \quad (4.22)$$

Where a is the input value of a given layer at a particular time step, b is the bias, W is the weight, C is the cell state, U is the recurrent connection, x is the input, and N is the total number of layers.

4.4.2 Deep-belief Network

Deep-belief networks are neural networks formed by stacking restricted Boltzmann machine (RBM), one on top of another. The RBM is a simple neural network with just two layers, namely, hidden and visible layers. They are termed ‘restricted’ because they do not have any intra-layer connections, as shown in Figure 4.12.

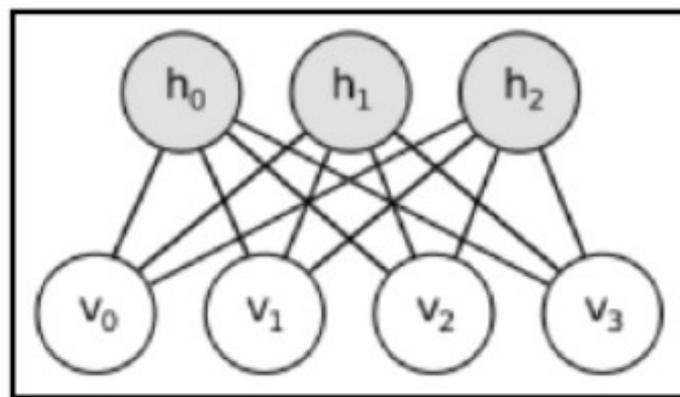


Figure 4.12 Basic RBN structure [160]

The RBM is an energy-based model; its energy function E produces a scalar value, which specifies the model's probability in that configuration.

$$E(v, h) = -a^T v - b^T h - v^T W h \quad (4.23)$$

Where W is the weight associated with the connections between the units in the visible layer v and the hidden layer h , a is the bias of the units in the visible layer, and b is the biases of the units in the hidden layer. The probability P between each pair of the visible and hidden vector is given by [162];

$$P(v, h) = \frac{e^{-E(v, h)}}{S} \quad (4.24)$$

Where S is a partition function and is defined as the summation of $e^{-E(v, h)}$ over every possible configuration. Figure 4.13 shows a DBN with three hidden layers and one visible layer.

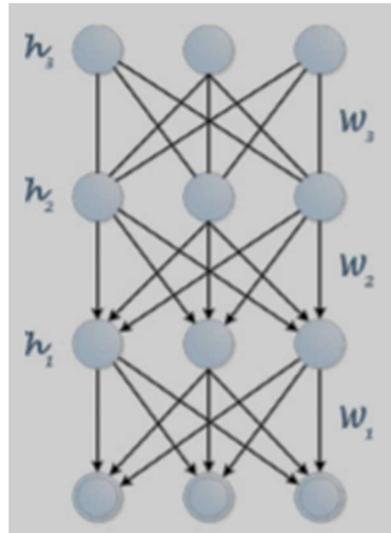


Figure 4.13 DBN structure [163].

4.4.3 Convolution neural network

A convolutional neural network (CNN or convNet) is a feedforward neural network that can be effectively used in visual imagery analysis and classification type problems. The CNN revolution was set in motion by Alex Krizhevsky, who won first place in the 2012 ImageNet competition with his CNN called AlexNet [93]. AlexNet achieved a classification error reduction of 11 %, building on previous works like Yann LeCun's LeNet, which was mainly used for character recognition tasks [164]. The CNN comprises four fundamental layers: convolution layer, activation layer, pooling layer, and fully connected layer. Typically, a functional CNN consist of groups of convolution, activation, and pooling (Conv-ReLU-Pool) layers stacked together, followed by the fully connected layers, as shown in Figure 4.14.

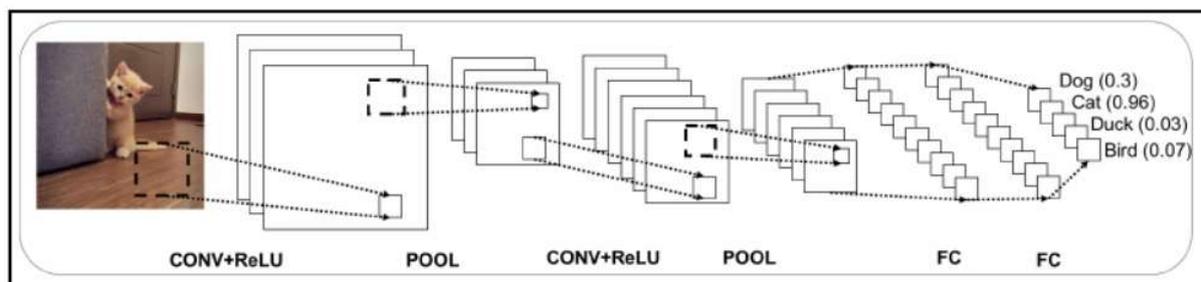


Figure 4.14 A typical CNN with two stacked set of convolution, activation, and pooling layers and two fully connected layers [160]

The convolution layer performs a dot product multiplication of the input image and a randomised weight matrix. The weight matrix is technically referred to as the convolution filter.

The action of the filter is shown in Figure 4.15.

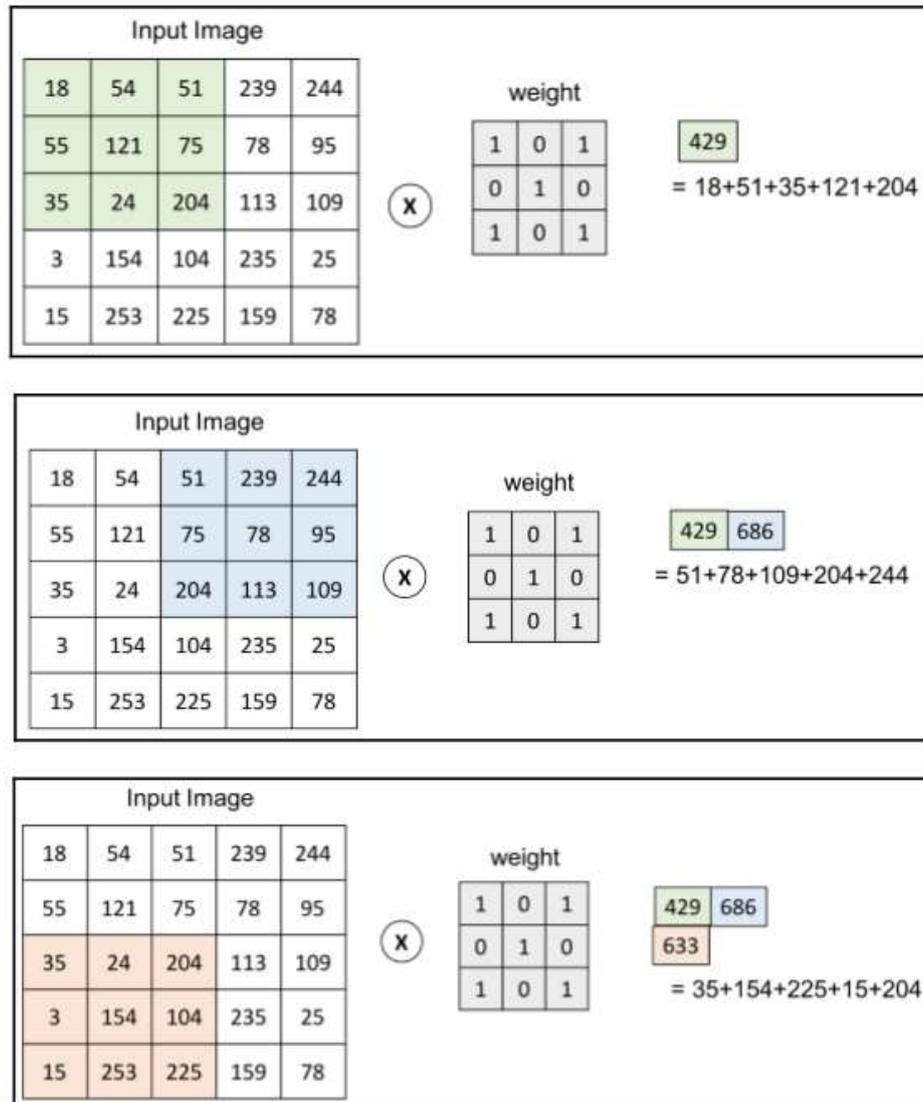


Figure 4.15 Convolution example with a stride of 2 [160]

The filter performs convolution across the input image from left to right and up to down. The number of pixels the filter moves at a time during convolution is called the stride. Convolution results in a matrix of smaller size compared to the original input image. Typically, the larger the stride, the smaller the result. In order to regain the original image size after convolution, a technique of introducing zeros to replace the missing cells is used; this is illustrated in Figure 4.16.

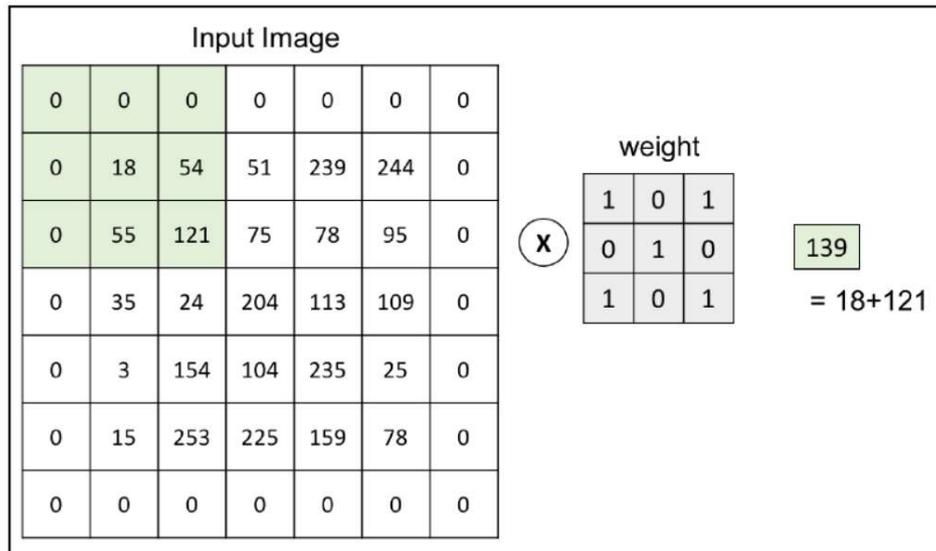


Figure 4.16 Zero padding for convolution [160]

The filter size is typically smaller than the input image size, and the weight is adjusted in training using stochastic gradient descent (SGD) with momentum technique. Mathematically, the SGD with momentum is given by [102];

$$L = \frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^n 1\{y^{(i)} = j\} \log \frac{e^{W_j^T x^{(i)}}}{\sum_{l=1}^n e^{W_l^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{j=1}^n W_j^2 \quad (4.25)$$

$$\nabla_W L(W; x^{(i)}, y^{(i)}) = \frac{1}{m} \sum_{i=1}^m [x^{(i)} \{1(y^{(i)} = j) - p(y^{(i)} = j | x^{(i)}; W)\}] + \lambda W_j \quad (4.26)$$

$$v \leftarrow \varepsilon v - \alpha \nabla_W L(W; x^{(i)}, y^{(i)}) \quad (4.27)$$

$$W_{j+1} \leftarrow W_j + v \quad (4.28)$$

The loss function L is first evaluated from the deviation between the actual class and the predicted class given initial random weights W . the term $1\{y^{(i)} = j\}$ returns the value one when a predicted class of the i th term input is true for j class; otherwise, it returns a zero. The hyperparameter λ is a regularization parameter used to penalise wrong weights and therefore help to prevent overfitting [165, 166]. Secondly, the gradient of the loss function $\nabla_W L$ is evaluated with respect to the weight as a function of the input x and output y . The expression $p(y^{(i)} = j | x^{(i)}; W)$ is a probabilistic SoftMax function of the i th training epoch out of a total

of m training epochs. Thirdly, the hyperparameters momentum ε and learning rate α are used to update \leftarrow the velocity v . Finally, the velocity is used to update the weight.

The pooling layer is directly after the ReLU activation function in the Conv-ReLU-Pool stack and generates feature maps by down sampling the input image. The number of features is determined by the number of input neurons in the layer it precedes. Max-pooling is the most commonly used in CNN's as it shows better performance over other types like mean-pooling. Max-pooling involves taking the maximum value in the filter area, while mean-pooling takes the average value in the filter area. Figure 4.17 shows a 2×2 max pooling with a stride of 2, and Figure 4.18 shows a 3×3 max pooling with a stride of 1 and overlapping.

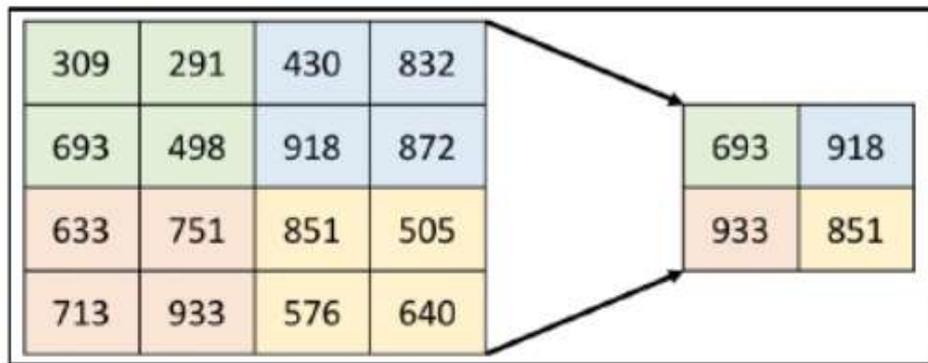


Figure 4.17 2×2 max-pooling without overlapping [160]

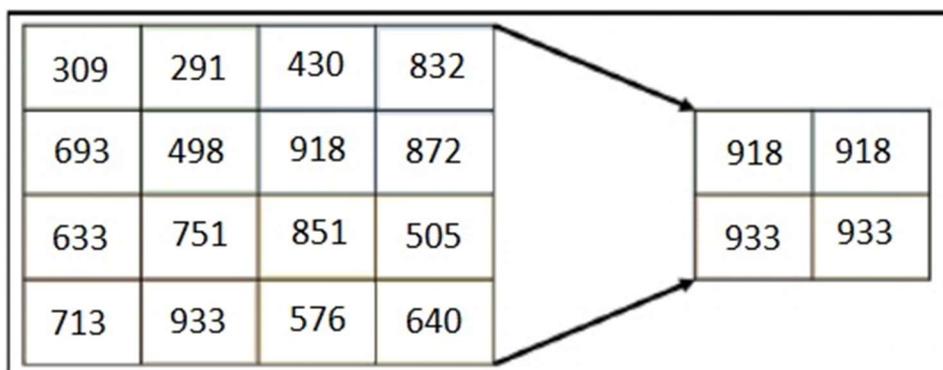


Figure 4.18 3×3 max pooling with overlapping [160]

The fully connected layer makes up the last few layers of CNN with a softmax activation function, which is used to output the prediction class in terms of probability. The class with the highest probability wins the vote and is therefore taken as the predicted class. A technique termed ‘dropout’ is also implemented in this layer. Dropout involves setting a threshold

probability, such that any inputs with a probability less than that threshold are set to zero. This is achieved by the random break-up of the co-adaptations typically generated in a standard backpropagation neural network. This has been found to improve robust feature learning and reduce overfitting in CNNs [93]. On the other hand, the use of dropout will result in the model spending more time in the training phase as it takes more time to converge.

4.5 The deep convolution neural network algorithm

A deep convolution neural network (DCNN) is a CNN with multiple hidden layers (convolution, activation, pooling, and fully connected layers). Its invention, along with the development of large database visualisation systems (ImageNet, which consists of more than 20000 categories of numerous images), enables the capability for a rich and wide-ranging feature summary from a broad spectrum of images [167]. These summarised features make an accurate pattern recognition technique that typically performs better than other classification techniques for a diverse set of image data [167]. This implies that DCNN supports inter-domain transfer learning and its summarised features are general, even when the quantity of available data is limited [168]. This research adapts the DCNN obtained from the ImageNet pre-trained network called AlexNet. It is a deep learning network with twenty layers (eighteen hidden layers), as shown in Table 4.1.

Table 4.1 DCNN layers

Layer	Description	Neurons	Dimension
1	Input	2	227X227X3
2 – 4	Conv-ReLU-Pool 1	96	11x11x3
5 - 7	Conv-ReLU-Pool 2	256	5x5x48
8 – 10	Conv-ReLU-Pool 3	384	3x3x256
11 - 13	Conv-ReLU-Pool 4	384	3x3x192
14 - 16	Conv-ReLU-Pool 5	256	3x3x192
17	Fully connected 1	4096	-

18	Fully connected 2	4096	-
19	Fully connected 3	2	-
20	Output	2	

Layer 1 represents the input layer. It is designed such that it can only accept image files. It has two neurons with sizes set as $227 \times 227 \times 3$. The number of neurons is set by the number of prediction classes.

Layer 2 to layer 9 represents the hidden layers, which is the engine room of the algorithm. It consists of five stacks of Conv-ReLU-Pool layers. Each of these stacks performs convolution, applies the ReLU activation function, and then performs maximum pooling in that order to each input before sending it to the next layer. The ReLU activation is used because it has been proven to make DCNNs at least six times more efficient at the training stage and reduce operational overhead [93]. A 3×3 pooling with a stride of 2 is used for summarising the feature in this network. This down sampling technique reduces overfitting and the error rate by at least 0.3 % [93, 161]. Figure 4.19 portrays a summarised structure of the DCNN.

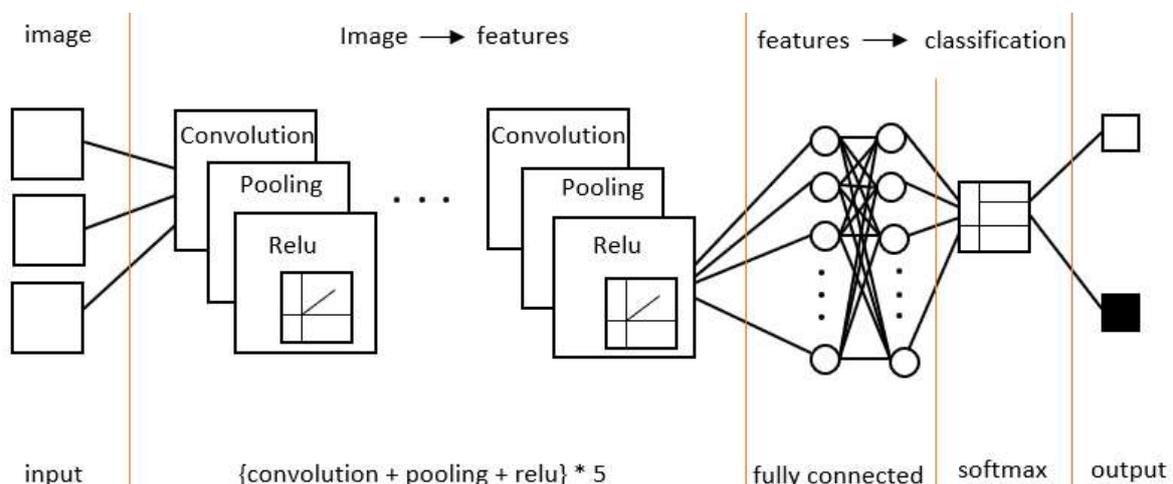


Figure 4.19 DCNN architecture

Channel-wise normalization is performed in advance of the ReLU activation layer as it helps feature vector generalization among the input data. The response-normalised output y for every individual input element x and channel window size n is given by the expression;

$$y = \frac{x}{\left(1 + \frac{s}{10000n}\right)^{\frac{3}{4}}} \quad (4.29)$$

Where s is the sum of the squares of the elements in the normalized window. The constant terms were set during training using a validation set with a window channel size of 5. Layer 17 to layer 19 represents the fully connected layers. They perform supervised learning based on the identified classes, on the features extracted in the Conv-ReLU-Pool layers. Layer 17 and layer 18 have 4096 neurons; each performs arbitrary dropouts by setting to zero all input elements with a probability of 0.5 or less.

Layer 20 represents the output layer. This is where the final classification result is presented based on the accuracy of the DCNN algorithm. The output has two neurons, which is defined by the number of prediction classes of the network. The network utilises the cross-entropy function for a 1-of-2 coding scheme for error evaluation, which is expressed as;

$$E(\theta) = - \sum_{i=1}^n \sum_{j=1}^2 t_{ij} \ln y_j(x_i, \theta) \quad (4.30)$$

Where t_{ij} shows that the i th sample belongs to the j th class, $y_j(x_i, \theta)$ represents the output for sample i , n indicates the number of observations, and θ represents the parameter vector.

Preliminary analysis of the time-series data collected from the experiment and the results were first published in the proceeding of the Euspen international conference and exhibition [2], with more information provided in chapter 5. The results shows that using the numeric data directly as an input to the deep learning network gives poor results, only achieving 51 % accuracy [2]. This was done by arranging the time-series data into mini-batches [169] , after being sorted for even distribution of the mini-batches. Another approach used was the conversion of the time-series data to image files such that the image serve as input to the deep learning network using the advantage of image convolution for the analysis [170] . This approach of utilising image files has been shown to give good results [93]. It was able to

achieve an accuracy of 94 % [2] when used with the image file of the vibration data, which is very promising. Hence, it is the approach that will be employed in this thesis.

4.6 Chapter summary

Deep learning is an aspect of AI that uses a complex algorithm with multiple hidden layers to train a network. It learns the hierarchical structures within a dataset and performs complex mapping of an input space to a predetermine output space without an expert agent's intervention. This is also how it differs from conventional machine learning, which will require an expert agent to perform feature extraction. There are many different types of deep learning networks like the recurrent neural network, deep-belief network, and deep convolution neural network. The proposed deep learning model for this research is the deep convolution neural network (DCNN). It has an input layer, five convolution layers, five ReLU layers, five max-pooling layers, three fully connected layers, and an output layer (see section 4.5). A stack of the convolution, ReLU, and max-pooling layers is responsible for the feature extraction, while the fully connected layer is responsible for classification. The next chapter delves into the experimentation and analysis of the dataset from the experiments.

Chapter 5 Experiments and analysis of the dataset

This chapter presents an insight into the experimental setup and procedures for this project and the corresponding data collected. It starts by providing an overview of the design of the experiment as it applies to this research. Next, it discusses particular experiments that generate each dataset. This is followed by an evaluation of a test dataset with the proposed algorithms. This chapter checks the viability of the proposed strategies in terms of their strengths and weaknesses regarding classification problems. It also assesses the practicality with which the proposed strategies are adaptable to condition monitoring of ball screws assembly. Materials used for this chapter include journals, conference papers, datasheets, and textbooks.

5.1 Design of experiment

As with every research project, the design of experiment is instrumental in achieving the research aims. This is because a poorly designed experiment will lead to erroneous results and, consequently, wrong conclusions. Hence it is crucial to carefully and properly plan research experiments to fit the requirement. The design of experiment for this research will be discussed under three sections, namely: pre-test, test set-up, and test procedure.

5.1.1 Pre-test

The pre-test activities involve all the preparatory activities that precede setting up the experiment. It includes selecting the appropriate hardware and software to be used for the test.

5.1.1.1 CNC machine

The CNC machine used in this study is a five-axis gantry-type milling machine (Figure 5.1). A computer-aided design (CAD) representation of the machine without guarding is provided in Figure 5.2. The five-axis machine comprises three linear axes: X, Y, Z, and two rotary axes, namely B and C. The machine tool spindle, on which the tool is fitted, is mounted on the rotary

B-axis, and all axes move to position the tool. The maximum feed rate of the X-axis, Y-axis, and Z-axis are 75000 mm/min, 75000 mm/min, 70000 mm/min, respectively. The travel spans along the same axis are 2500 mm, 1200 mm, 700 mm, respectively. The maximum rotational speed of the spindle is 3200 rev/min.

The experiments are all performed on the Y-axis of the machine. This is the “gantry” axis, which is a system of two independent closed-loop ball screw drives. The left side ball screw is Y1, while Y2 is on the right (Figure 5.2). This axis was chosen for this work because the performance of the two ball screws can be compared during a single test without introducing additional influencing factors; for a given test, each screw moves the same nominal distance, with the same loading, the same acceleration, and jerk (rate of change of acceleration), the same nominal temperature conditions, and the same overall life cycle. Both Y1 and Y2 have a fixed ball screw and a rotating ballnut. The nominal pitch for the ball screw is 32 mm in each case. To enable comparison between “good” and “worn” ball screws, Y1 is in a good state, while Y2 has been damaged from accelerated wear through increased friction. The damage on ball screw Y2 was inadvertently generated when a fault meant that the normal lubricant flow to its ballnut was restricted.



Figure 5.1 The 5-axis gantry-type CNC machine used in this study

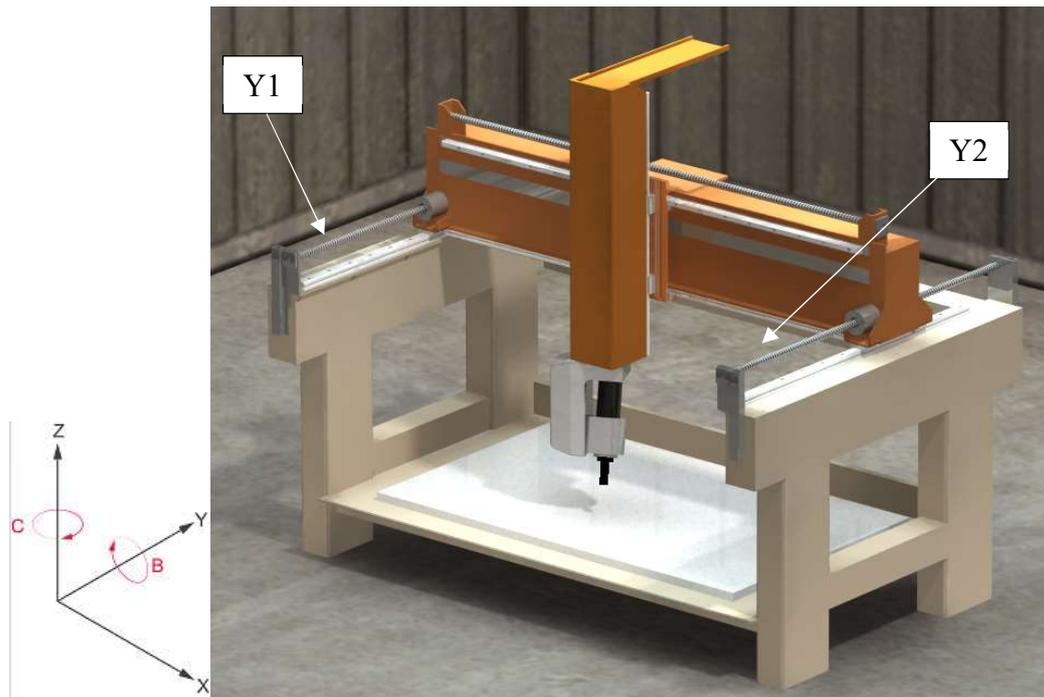


Figure 5.2 A CAD model of the 5-axis gantry-type CNC machine

5.1.1.2 Accelerometer

An accelerometer is an electromechanical instrument used for measuring vibration. It does this by measuring proper acceleration and then converts it to electrical signals, which are representative of the vibration in the body being monitored. Proper acceleration refers to the rate at which the velocity of a body changes with respect to its real-time rest frame. This is unlike conventional acceleration, otherwise known as coordinate acceleration, which is the rate at which the velocity of a body changes in a fixed coordinate system. Accelerometers are typically calibrated by comparison to a known acceleration level. For this research, a piezotronics accelerometer sensor, model 629A11, is used (see Figure 5.3). It is a triaxial sensor with a response frequency range of 2.4 to 2000 Hz. This falls within the range of vibration signal, which makes this sensor suitable for use in this application. The size is 31.1 mm x 38.1 mm x 20.8 mm (length x width x height), ideal for mounting on a relatively small surface area.



Figure 5.3 Tri-axial accelerometer used for model calibration and testing

5.1.1.3 Acoustic emission sensor

Acoustic emission (AE) sensors are sensors used in non-destructive testing applications to detect and measure high-frequency signals or ultrasonic stress waves released from localised sources when a material deforms or vibrates under the influence of an active force. For this research, a piezoelectric type with an integrated amplifier (model VS150-RIC) is used. The size is 28.6 mm x 31.5 mm (diameter x height), suitable for mounting on a relatively small surface area. The AE sensor has a large frequency response range of 100 kHz to 450 kHz. This makes it suitable for detecting possible high-frequency ultrasonic stress waves that could be released from the ball screws in motion. Figure 5.4 shows the AE sensor kit, which consists of the sensor, a mounting holder, and an analogue to digital converter (ADC), and Figure 5.5 shows the AE sensor mounted on a ball screw assembly.

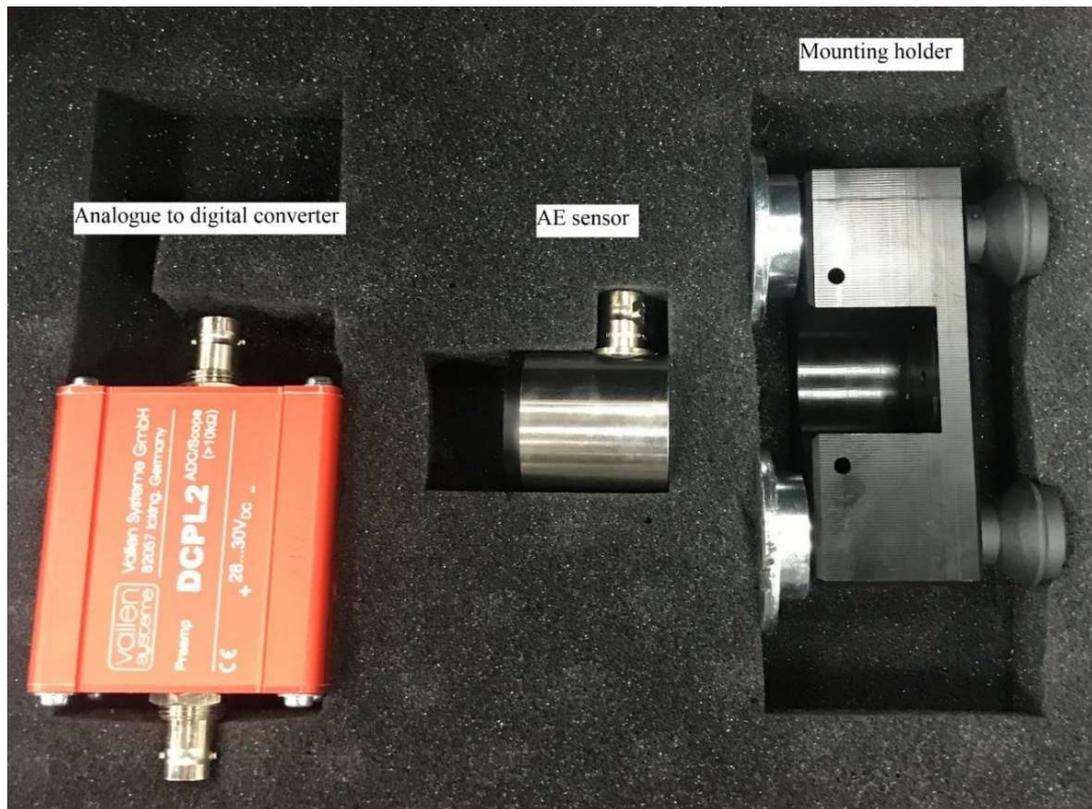


Figure 5.4 Acoustic emission sensor kit

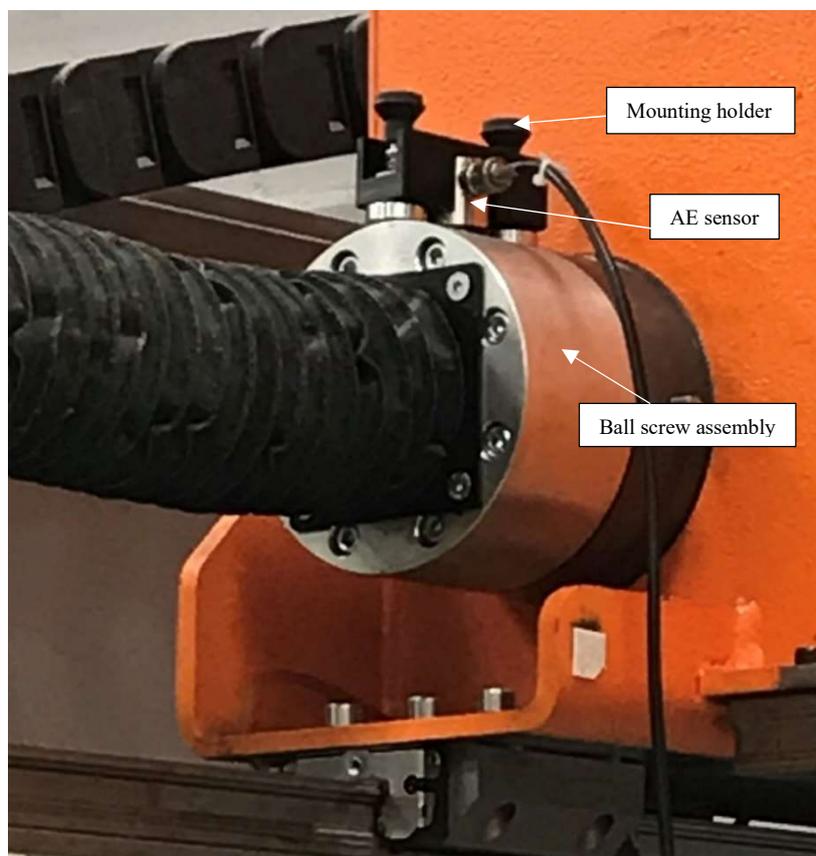


Figure 5.5 AE sensor mounted on a ball screw assembly during a test

5.1.1.4 Data acquisition kit

Data acquisition is an essential aspect of conducting experiments as, without it, the experiment is useless. It is the process of converting sampled analogue waveforms that are a measure of real-world conditions into digital numeric values that can be worked on using a personal computer. The type of data acquisition kit employed will largely depend on its compatibility with the sensor used. In this research, a national instrument (NI) data acquisition kit (NI compact data acquisition (NI-cDAQ) – 9174 and NI – 9234) is used to capture the accelerometer data. The NI-cDAQ – 9174 is a compact four-slot universal serial bus (USB) chassis that houses the NI – 9234. The NI – 9234 is a four-channel dynamic signal input module that performs condition monitoring functionalities like frequency analysis and order tracking. It can also perform signal coupling, signal conditioning, and anti-aliasing filtration for automatic sample rate adjustment. Figure 5.6 shows how the data acquisition equipment setup.

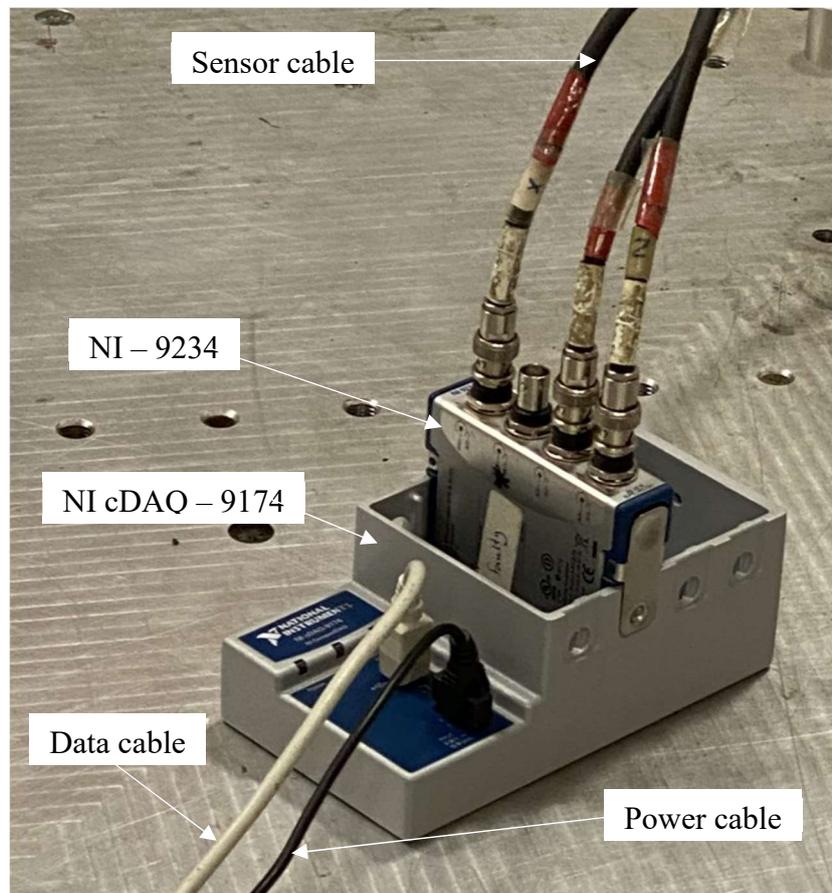


Figure 5.6 Data acquisition kit as wired for the experiment

5.1.1.5 Laboratory Virtual Instrumentation Engineering Workbench

The Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) is a NI-developed, graphical programming language for representing applications and processes. It is a high-level and system-level design platform for test, measurement, and control applications that demand quick access to hardware and data insights. It offers a visual approach to analysing every aspect of an application, from the hardware configuration to measured data. It does this by creating intuitive flowchart-like blocks called virtual instruments (VI), which are integrated within the graphical framework to form a complete design [121]. VIs in LabVIEW are modular, which implies that each VI can run independently as long as data flows into it. This characteristic of VIs makes it possible for a LabVIEW program to perform multiple operations in parallel as task execution is based on the flow of data rather than the sequence of lines of code [121]. In this research, LabVIEW is used for accelerometer data visualisation and logging.

Figure 5.7 shows the front panel of the LabVIEW software.

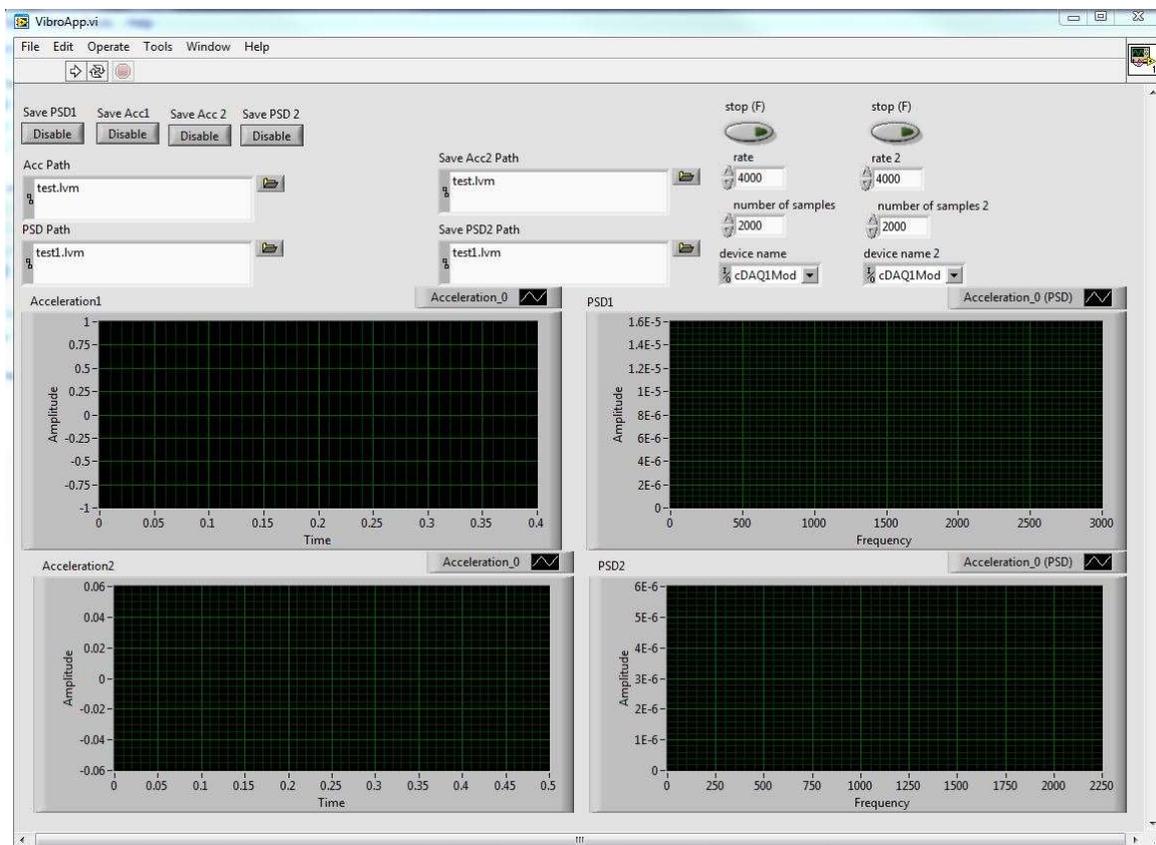


Figure 5.7 Front panel of LabVIEW software

This LabVIEW VI was created by the researcher for the efficiency of acquisition and checking data quality using NI hardware. However, the main analysis is performed in MATLAB because of the greater flexibility of MATLAB.

5.1.1.6 Temperature sensor

A temperature sensor is a device that can be used to the temperature of a surface or body. This research uses a DS18B20 digital thermometer [171]. It is used to monitor the change in the temperature of the machine structure and the environment. These sensors have been previously mounted on several locations on the CNC machine under investigation [81]. Figure 5.8 shows a short section of the DS18B20 temperature loom attached to the frame of the CNC machine to monitor any general increase in temperature during the series of experiments.



Figure 5.8 Three DS18B20 temperature sensors mounted on the CNC machine frame.

Similar sensors are located within the ball screw drive to provide direct information on any heating of the ball screw and ball nut but are currently inaccessible for photographing. The DS18B20 temperature sensors communicate over a 1-wire bus to a central microprocessor, thus allowing the sensors mounted on the machine to be wired using only a single

communication wire plus two power wires in series. This is a significant advantage over other sensor types, which require between two and four wires per sensor. The operating temperature of the sensor is between $-55\text{ }^{\circ}\text{C}$ and $+125\text{ }^{\circ}\text{C}$, which comfortably accommodates the expected changes in temperature on the machine of $10\text{ }^{\circ}\text{C}$ ambient and up to $50\text{ }^{\circ}\text{C}$ on machine heat sources, such as motors.

A software application called WinTcal (version 2.4.0) is used to display and log the temperature data. WinTcal is an in-house developed software by the Centre for Precision Technology (CPT) research group of the University of Huddersfield. Figure 5.9 shows a picture of the WinTcal display. The software can identify all the temperature sensors on the CNC machine tool and capture the temperature of the contact surface. It logs these temperatures in a file specified by the user. The logged data can be displayed in real-time as a line graph and instantaneously beside the pre-set sensor name. All subsequent analysis was carried out in MATLAB.

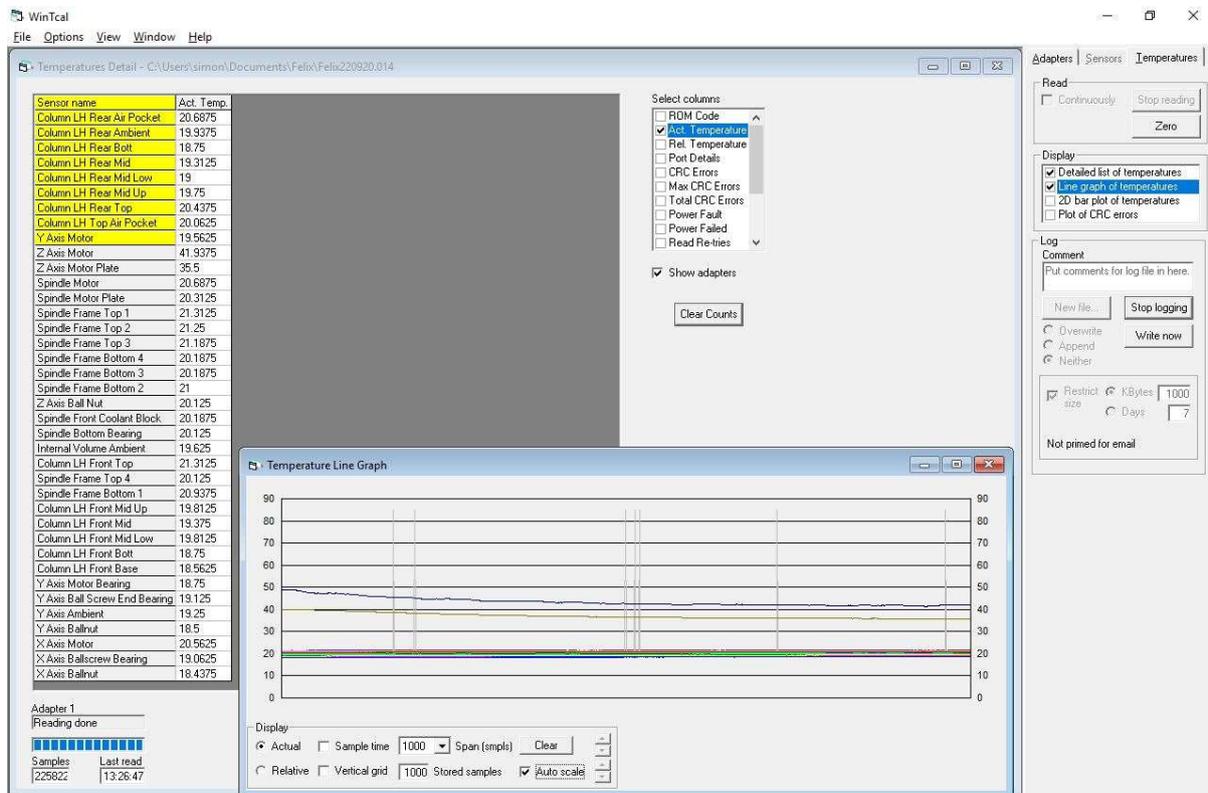


Figure 5.9 WinTcal interface

5.1.1.7 Personal computer

Personal computers (PC) are used to facilitate data logging and processing. It is also where the mentioned software is installed. For this research, three PCs are used, PC-1 for monitoring the experiments (on which LabVIEW is installed), PC-2 for monitoring temperature (on which WinTcal is installed), and PC-3 for data processing and running of simulations. Table 5.1 shows the specifications of the PCs.

Table 5.1 PC specifications

	PC-1	PC-2	PC-3
Processor speed	2.8 GHz	3.1 GHz	3.6 GHz
RAM	12 GB	12 GB	16 GB
Processor	Intel core i5	AMD	Intel core i7
Operating system	Windows 7	Windows 7	Windows 10
System type	64-bit OS	64-bit OS	64-bit OS

Ideally, all data would be captured using a single software application on a single PC to ensure common timing and synchronisation. However, for this PhD project, using separate software and PCs is acceptable since the temperature change is relatively slow, so synchronisation is only needed on a per-experiment basis.

5.1.2 Test setup

The test setup describes all the activities involved in setting up the hardware and software required for the experiment. These activities are list below in the sequence they were carried out;

1. Turn on the CNC machine energise all drives, and perform the homing sequence to ensure the correct coordinate frame is used.
2. Position the ram of the CNC machine such that it is mid-way from either ball screw.

This was done to ensure that its weight is evenly supported on both ball screws, not

giving either ball screw more load, which could lead to forced degradation on that ball screw during the test.

3. Attach the sensor to the machine tool ball screw. The sensor is attached to the rotating ballnut of the machine tool via magnets. This is as close as physically possible to the ball screw the sensor can get based on the machine tool structure. The sensor is positioned such that its X-axis aligns with the direction of motion of the CNC machine tool axis, as illustrated in Figure 5.10. This is done to minimise the noise on the data by eliminating the influence of yaw error.

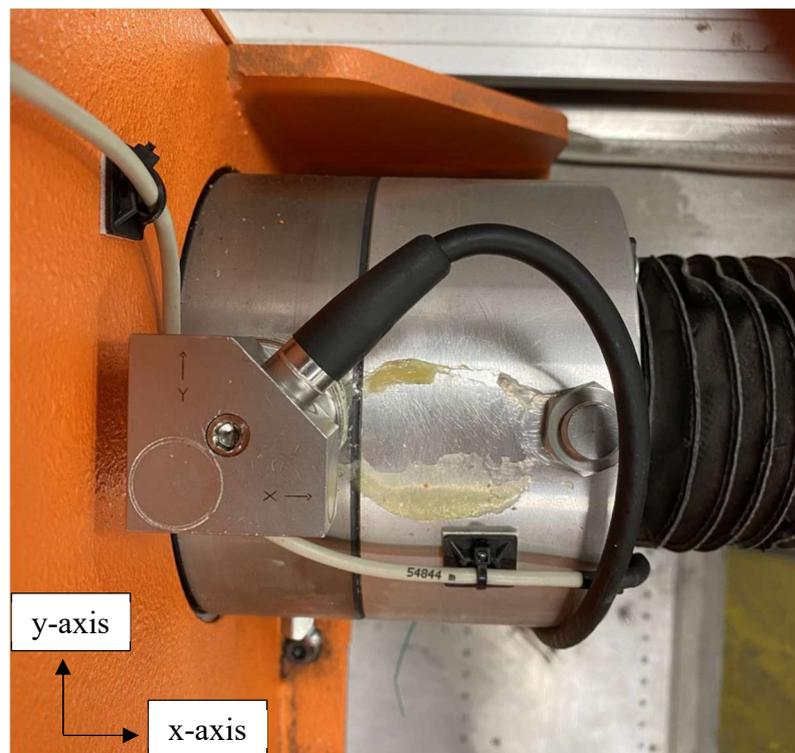


Figure 5.10 Accelerometer on ballnut with axes aligned with the machine axes (plan view)

4. Set up the program on the CNC machine control panel that will run the machine tool's gantry at different feed rates, from front to back through the entire length of the ball screw (about 1500 mm). The data captured during a run over the entire ball screw length is an entry in a dataset. The different speeds chosen for the test are 10000 mm/min, 1000 mm/min, 9000 mm/min, 2500 mm/min, 7500 mm/min, 3000 mm/min, and

5000 mm/min, in that order. The reason for this is to maintain a steady temperature rise in the ball screw.

5. Run the WinTcal for several hours (at least 8 hours) before the test. This is done to check that the temperature on the CNC machine components has saturated (without trends) before the test starts to eliminate the possibility of spurious data.
6. Setup the data acquisition kit.
7. Setup LabVIEW for data capture.

5.1.3 Test procedure

The test involves the movement (at different feed rates) of the axis along the span of the ball screw, Y1 and Y2, as shown in Figure 5.2. Each movement is performed in both the positive and negative directions of travel to identify any differences. The considerations for the test are expressed as follows:

1. The feed rates were selected quasi-randomly but within the normal operating range of the machine during operation.
2. The five-axis production machine (see Figure 5.1) used for the test was selected because it has a gantry of two closed-loop horizontally aligned ball screw drive mechanisms. Y1 is in good condition, and Y2 is noisy during operation. On further investigation, it was discovered that the lubrication system to Y2 was faulty, which had stopped the flow of lubrication to Y2. This is likely to have resulted to accelerated wear due to movement of the horizontal axis and the absence of lubrication in the ballnut for a long period of time. The sensor is then placed on the ball screw assembly as physically close to the source of vibration as possible.
3. The temperature data of the ball screws and surrounding atmosphere (ambient) was logged to monitor the thermal condition of the ball screws before, during, and after the

test. In order to remove bias in the datasets from the heating of the axis, the order of the feed rates in the experiments is also randomised.

The test procedure includes the following steps;

1. Log the ambient temperature and the temperature of the ballnuts.
2. Perform any initial thermal perturbation as specified for each test.
3. Run the LabVIEW ready for data capture.
4. Set the CNC to the start position; this coincides with the 0.0 mm point on the Y-axis (rear).
5. Set the feed rate of the CNC machine tool axis to 10000 mm/min.
6. Move the axis to -1200 mm (front) then back to rear fifteen (15) times and capture the data from the sensor. Include a 3 s dwell at each extreme of travel to allow for easy detection of reversal of the direction in the signal during signal processing.
7. Change the CNC machine axis speed to 1000 mm/min, 9000 mm/min, 2500 mm/min, 7500 mm/min, 3000 mm/min, and 5000 mm/min in that order. For each speed, repeat steps 5 and 6.

This procedure generates a dataset of 420 samples for the ‘good’ and ‘worn’ ball screw, giving a good sample size to train machine learning and deep learning networks.

5.2 Description of conditions for each dataset

The datasets used in this thesis were obtained from different instants of the experiment conducted in the university of Huddersfield CNC machine workshop. They were nine experiments/tests resulting in nine datasets, dataset_1 (baseline test dataset) was used to test and develop the algorithms used in this thesis, and the rest were used for validation. Table 5.2 shows the summary of the experimental activities.

Table 5.2 Summary of experimental activities

Dataset/ Test	Sensor	Initial thermal condition	Sensor alignment	Set of feed rates
1	Accelerometer	Room temperature	Aligned	Centuple
2	Accelerometer	Room temperature	Misaligned	Centuple
3	Accelerometer	5°C thermal perturbation	Aligned	Centuple
4	Accelerometer	10°C thermal perturbation	Aligned	Centuple
5	Accelerometer	15°C thermal perturbation	Aligned	Centuple
6	Accelerometer	20°C thermal perturbation	Aligned	Centuple
7	Accelerometer	25°C thermal perturbation	Aligned	Centuple
8	Accelerometer	Room temperature	Aligned	Non-centuple
9	Acoustic emission	Room temperature	Inapplicable	Centuple

The following subsections describe the conditions under which each dataset was obtained and the rationale for designing the test in each case. The analysis of the data sets follows in Section 5.3

5.2.1 Baseline test dataset

This dataset is obtained from experiment_1, which is performed as described in section 5.1.3, except that there is no thermal perturbation for this test. The temperature profile of the ballnut during the test is presented in Figure 5.11. Where A represents the ambient temperature, letters B, C, D, E, F, G, and H represent the temperature profile during the 10000 mm/min, 1000 mm/min, 9000 mm/min, 2500 mm/min, 7500 mm/min, 3000 mm/min, and 5000 mm/min feed rate respectively. A simpler version of the same profile is presented in Figure 5.12. Each bar indicates the final temperature at the particular feed rate and the ambient temperature indicated by the 0 mm/min feed rate. The profile shows that the temperature of both ball screws steadily rises from 24.2 °C to 26.8 °C during the experiment after an initial 1 °C rise during the 10000 mm/min feed rate test.

The offset between the temperature sensors in Y1 and Y2 is due to the uncertainty of measurement and the fact that they are physically separated from each other, so will have a slight difference in environmental condition. This is indicated in the 0.3 °C temperature difference between Y1 and Y2, which is inconsequential as the temperature trends are identical.

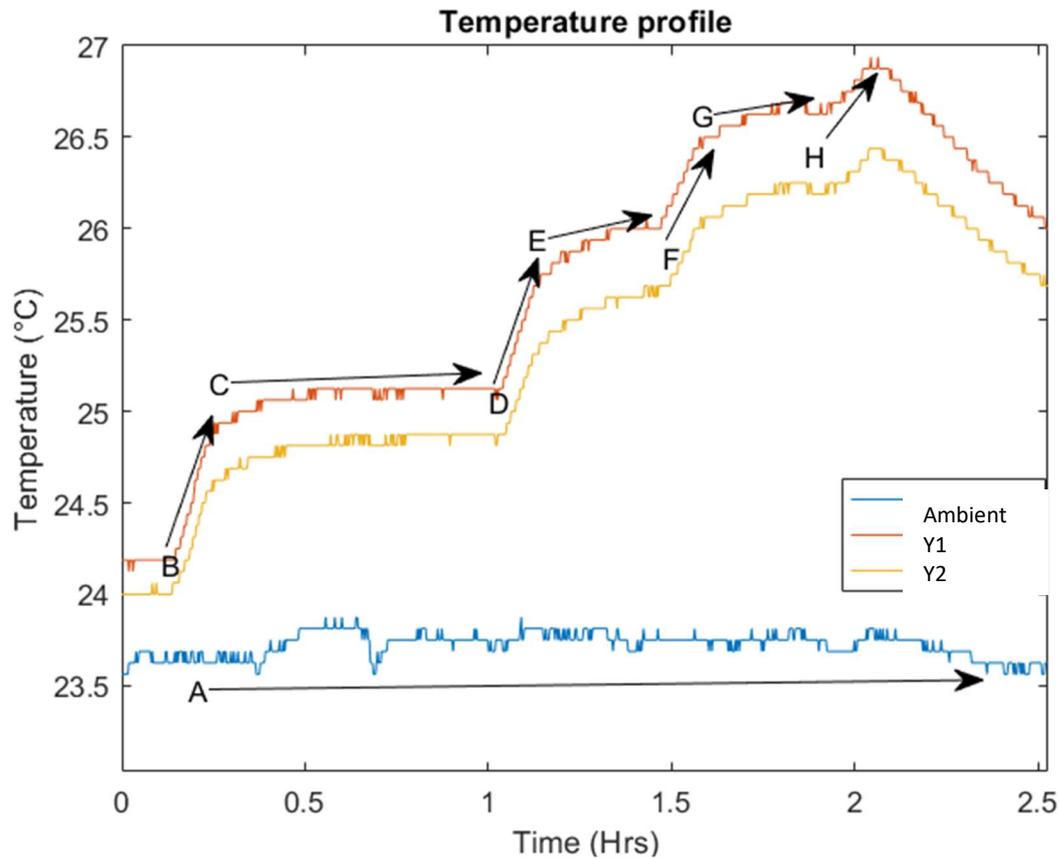


Figure 5.11 Temperature profile of the ballnut for the entire duration of baseline test

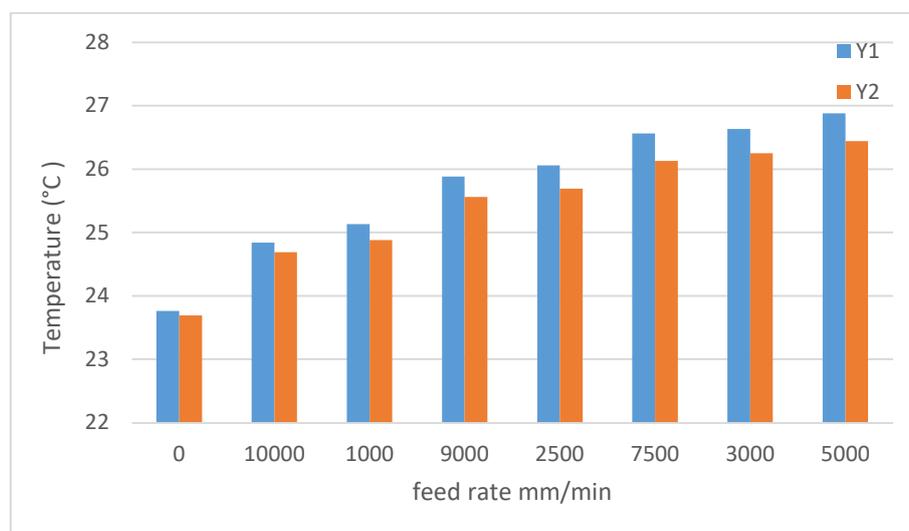


Figure 5.12 Temperature of the ballnut after each feed rate during baseline test

5.2.2 Misaligned sensor axis test dataset

This dataset is obtained from experiment_2, which is performed as described in section 5.1.3, except that there is no thermal perturbation for this test, and the sensor is mounted as shown in Figure 5.13. In this instance, the sensor axis does not align with the direction of travel of the machine axis. This was done to investigate the effect of misalignment of the axes of the sensor with those of the machine, specifically in the direction of travel of the axis. The temperature profile of the ballnut during the test is similar to that of experiment_1.

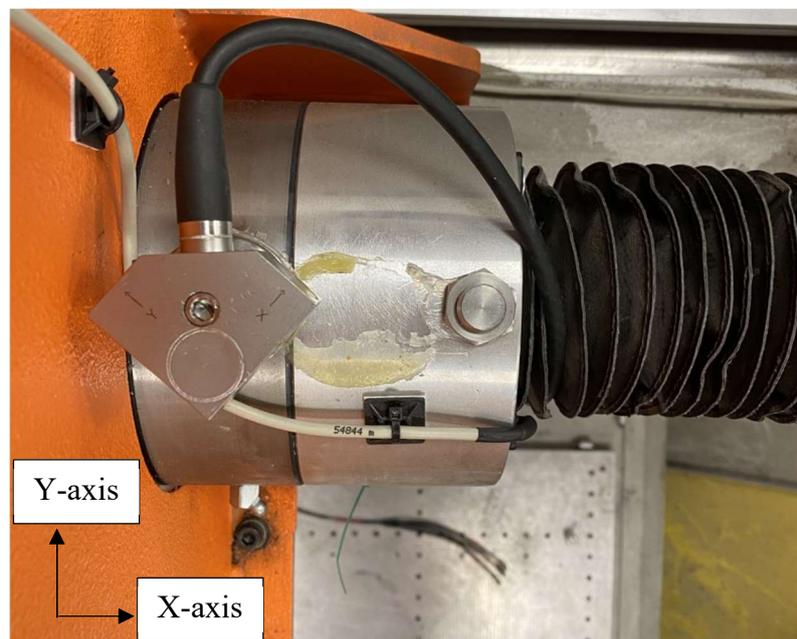


Figure 5.13 Accelerometer on ballnut with sensor X- and Y- axes not aligned with the machine axes (Plan view)

5.2.3 Thermal perturbation test datasets

These datasets are obtained from experiment_3 to 7, which are performed as described in section 5.1.3. The value of the thermal perturbation introduced in the ball screw for these experiments is as stated in Table 5.2. This was achieved by running the Y-axis at a feed rate of 40000 mm/min until the temperature rise, for the set thermal perturbation, above the ambient temperature is reached. The introduction of the thermal perturbation is to assess the impact of the possible associated thermal error on the accuracy of the proposed methodologies. Figure 5.14 shows the temperature of the ballnut after each feed rate for the 5 °C thermal perturbation

test. The chart shows that the temperature of the ball screws was initially raised from about 21 °C to 26 °C before starting the experiment. During the experiment, the temperature fell slightly (during the 10000 mm/min and 1000 mm/min feed rate test) until it stabilised at about 24 °C (at the 9000 mm/min feed rate test) for the rest of the experiment. The temperature charts for the 10 °C, 15 °C, 20 °C and 25 °C thermal perturbation test are shown in appendix V.

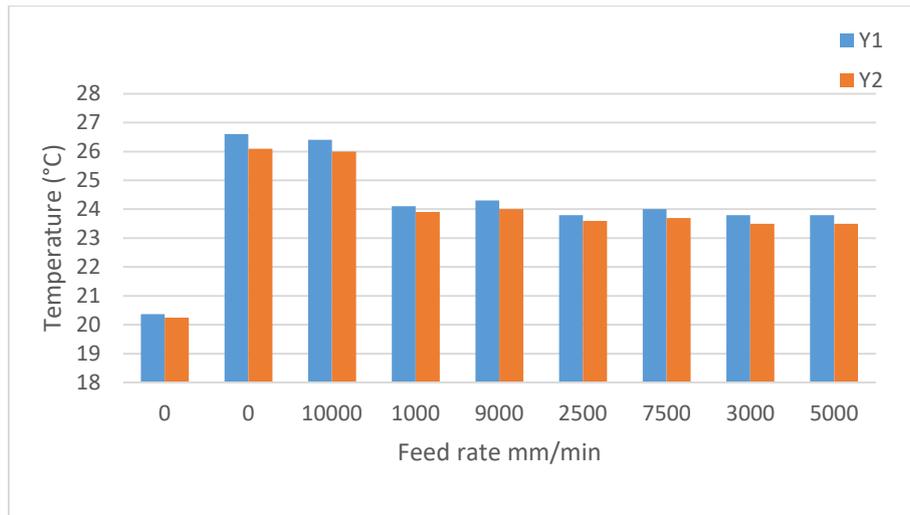


Figure 5.14 Temperature of the ballnut after each feed rate during the 5 °C thermal perturbation test

5.2.4 Non-centuple feed rates test dataset

The natural tendency is for experiments to have values set to multiples of the decimal system in which we normally work. ISO 230-2:2014 [172] notes that choosing a step size that is a multiple of the pitch of the ball screw for positional targets will result in aliasing. It solves this problem by requiring that each step size be quasi-random within the nominal step size. This test randomises the feed rates in a similar way.

This dataset is obtained from experiment_8, which is performed as described in section 5.1.3, except that there is no thermal perturbation for this test and the feed rate is set to 10126 mm/min, 1932 mm/min, 9213 mm/min, 2674 mm/min, 7507 mm/min, 3658 mm/min, and 5819 mm/min. These non-centuple feed rates were quasi-randomly selected (within the neighbourhood of the original feed rates) to test for any possible harmonic error that could

persist undetected because its frequency is a multiple of the decimal feed rate. The temperature chart of the ballnut during the test is shown in Figure 5.15. It shows that the temperature steadily rises from about 20 °C to 24 °C during the experiment.

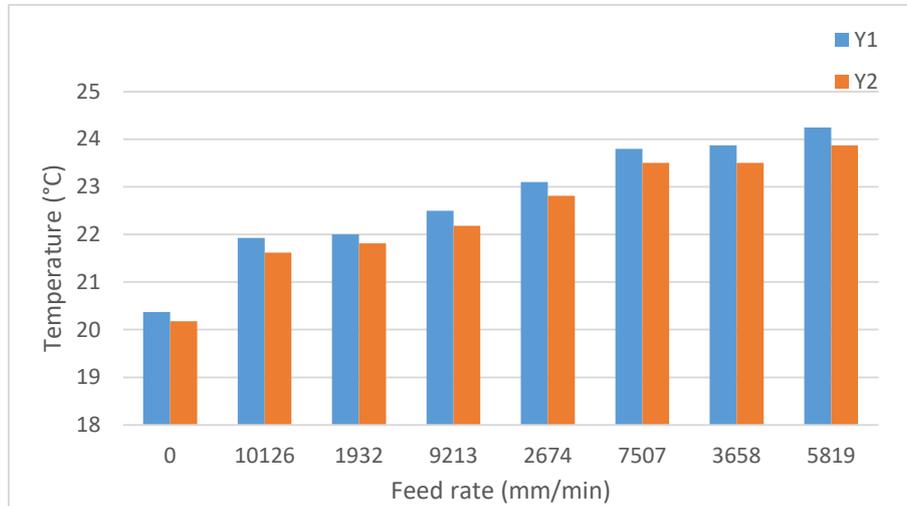


Figure 5.15 Temperature of the ballnut after each feed rate during the non-centuple feed rates test

5.2.5 AE sensor test dataset

This dataset is obtained from experiment_9, which is performed as described in section 5.1.3, except that there is no thermal perturbation for this test, and an AE sensor is used instead of the accelerometer. The temperature chart of the ballnut during the test is shown in Figure 5.16 and is similar to that of experiment_1. The AE sensor is mounted, as indicated in Figure 5.5.

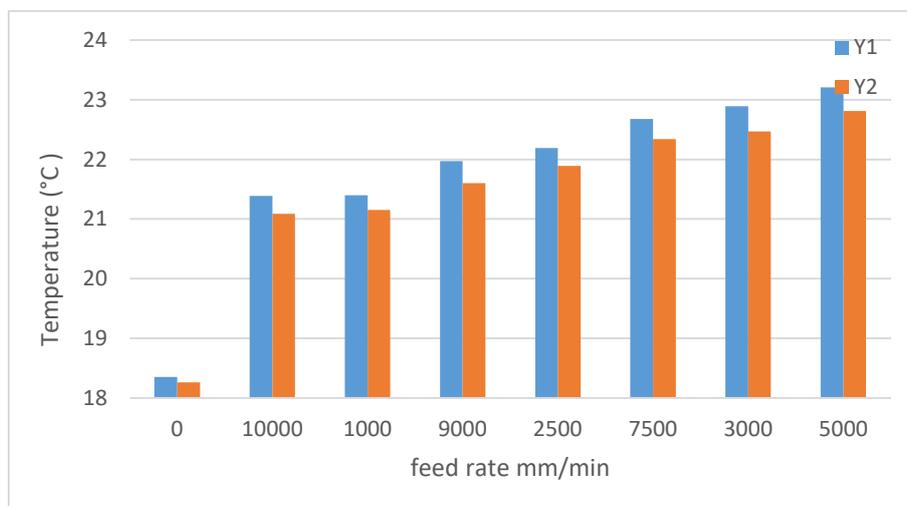


Figure 5.16 Temperature of the ballnut after each feed rate during the AE sensor test

5.3 Analysis of datasets

5.3.1 Temperature sensor data

The different datasets collected during the experiments include the temperature, vibration, and acoustic emission measurements. The temperature data has been presented for each test in section 5.2 above. The purpose of this data is to monitor the temperature response as the ball screw goes through its duty cycle during the tests, such that if abnormal behaviour is detected in the sensor data, then the corresponding temperature could be analysed. Also, a conscious effort was made to keep the temperature nominally constant by minimising the amount of heat generated from the chosen cycles and reducing any potential for thermally-induced trends to affect the interpretation of results. This was achieved by alternating between cycles with high and low feed rates during the test experiments, rather than building sequentially from low to high feed rate, which would likely have induced heat into the ball screw through friction, causing a temperature-induced bias in the later results. The temperature over the whole test duration was monitored and checked for changes before proceeding to the next stage. The temperature data for these tests show that both ballnuts rose steadily and consistently for the experiments that started from ambient (experiments 1, 2, 8, and 9). In contrast, experiments 3, 4, 5, 6, and 7 fell steadily and consistently after the initial forced temperature rise (thermal perturbation). This indicates that there is no significant temperature signature that indicates if a ball screw is good or worn for the given condition. Hence the temperature information was not included as inputs to the proposed strategies for applying artificial intelligence.

5.3.2 Accelerometer data

Samples of the sensor data from the good and worn ball screws at two different feed rates (1000 mm/min and 10000 mm/min) are shown in Figure 5.17, and their corresponding FFT spectrums are shown in Figure 5.18.

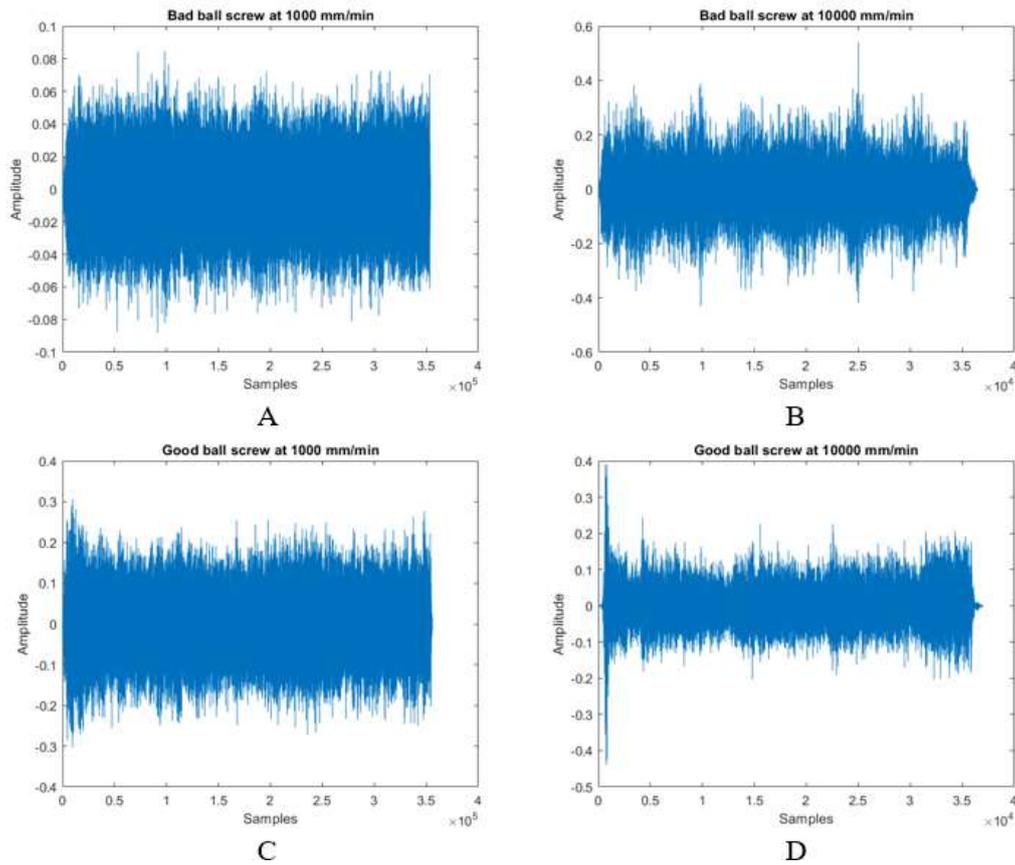


Figure 5.17 Accelerometer data of worn and good ball screw at 1000 and 10000 mm/min

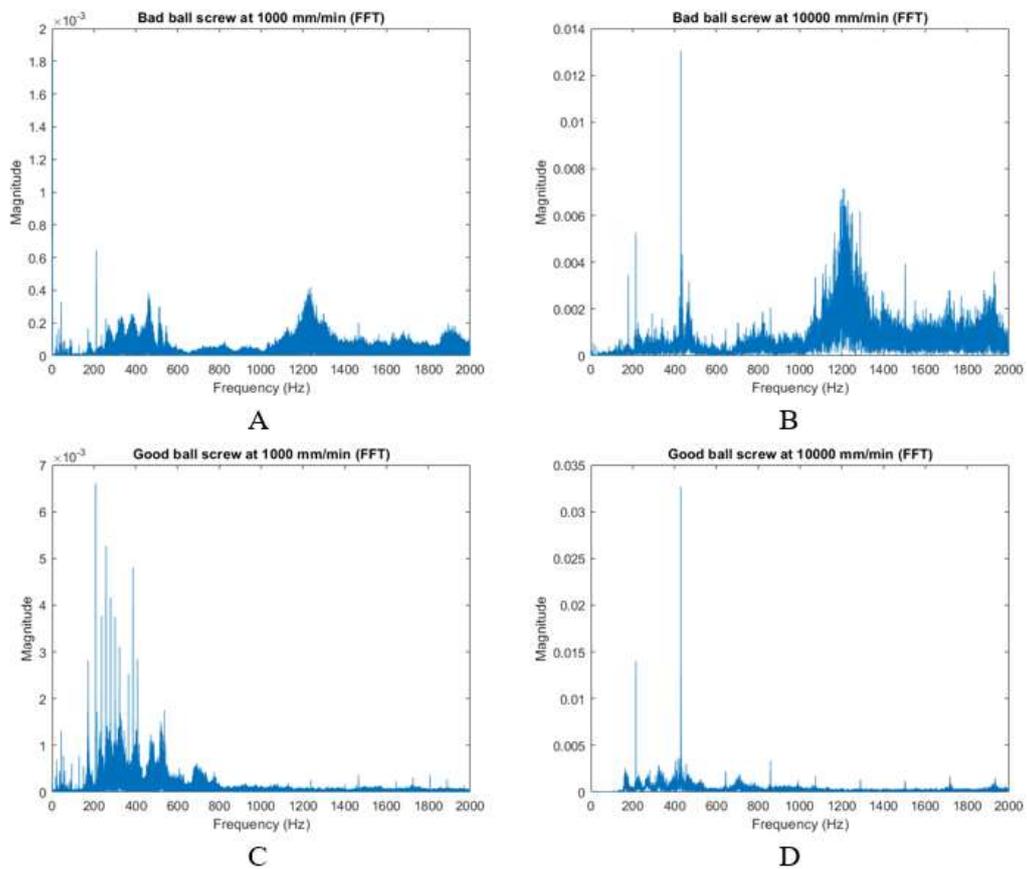


Figure 5.18 FFT of accelerometer data of worn/good ball screw at 1000 and 10000 mm/min

This gives a first glance at the vibration signal in the time and frequency domain, respectively, with both axes (of the plot) set to auto-scale to allow the shape of the features to be seen. The FFT spectrum shows that there is more high-frequency content in the worn ball screw data. The same FFT spectrum is presented in Figure 5.19, with the vertical axis all fixed to the same value to allow the magnitudes to be directly compared. It shows that the frequency content in the 1000 mm/min feed rate data is not as prominent as it initially appeared in Figure 5.18.

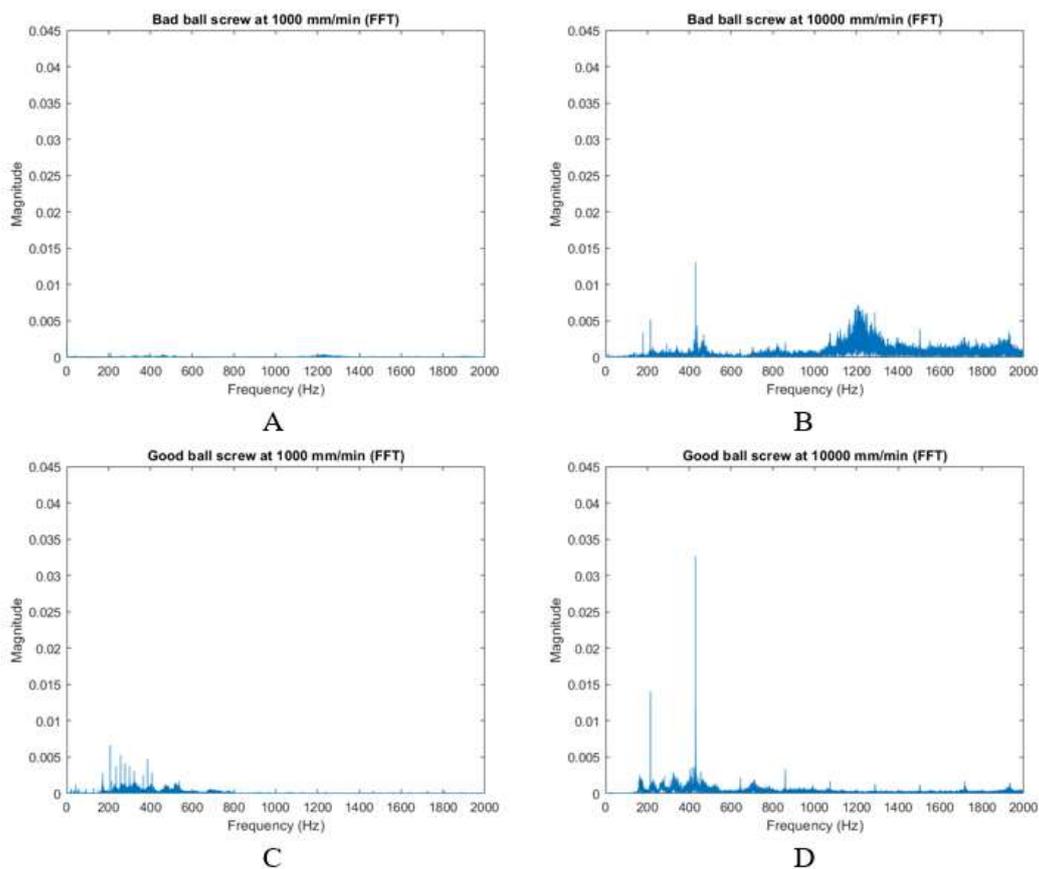


Figure 5.19 FFT of accelerometer data of worn and good ball screw with fixed axes

Further analysis of the collected data is presented in Figure 5.20 and Figure 5.21 for both the good and the worn ball screw, respectively, at all the test feed rates. In this case, the graph's axes are fixed for all the FFT spectrums for both ball screws data. It can be seen that the spectrum is not the same across all feed rates in terms of both frequency and magnitude. In the good ball screw, the amount, magnitude, and frequency of the significant peaks across the different feed rates are not consistent. For example, the 10000 mm/min feed rate data has two

significant peaks, one at 200 Hz and the other just above 400 Hz, while the 9000 mm/min feed rate data has one significant peak at 400 Hz.

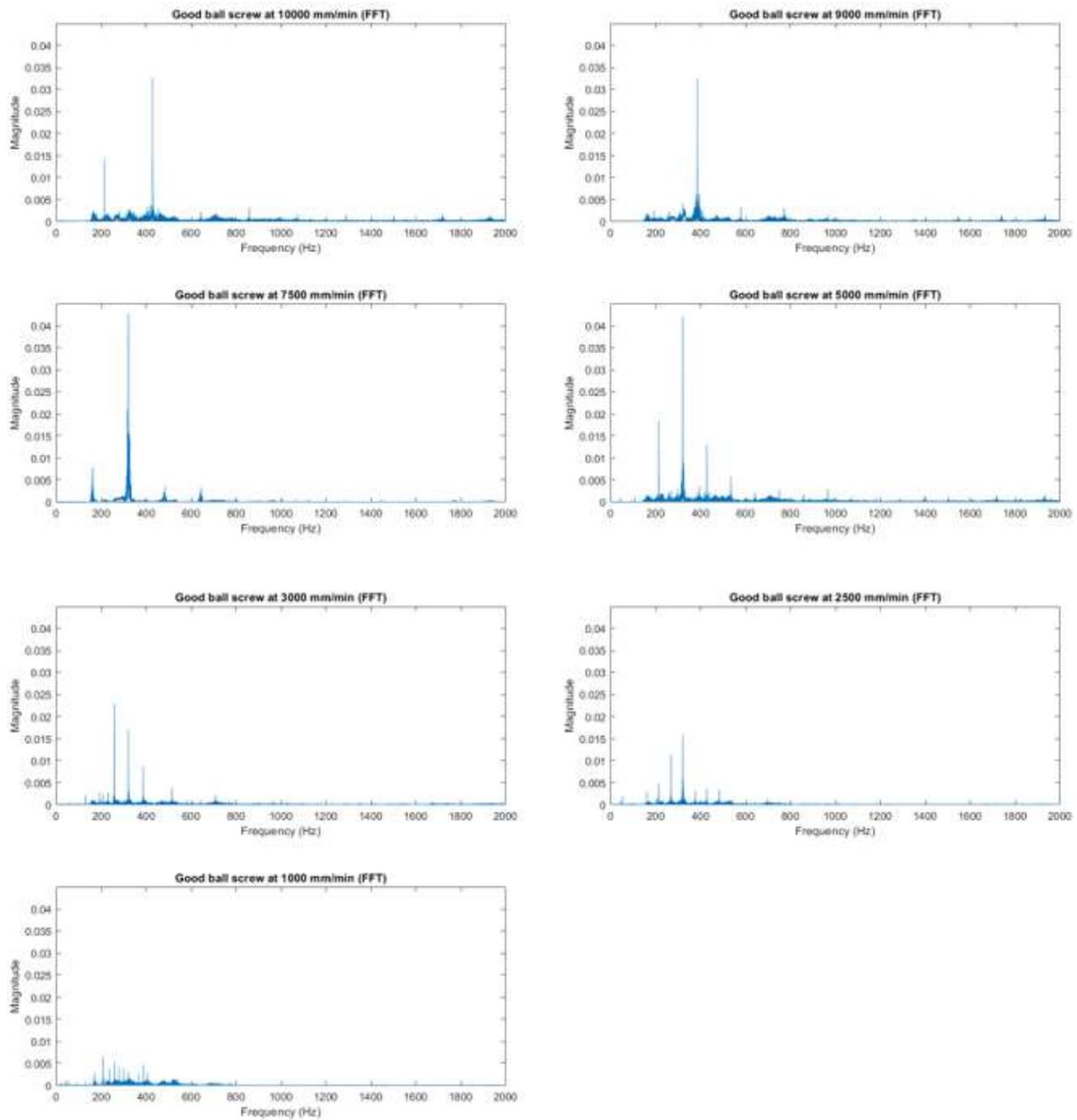


Figure 5.20 FFT of vibration signals at different feed rates for good ball screw.

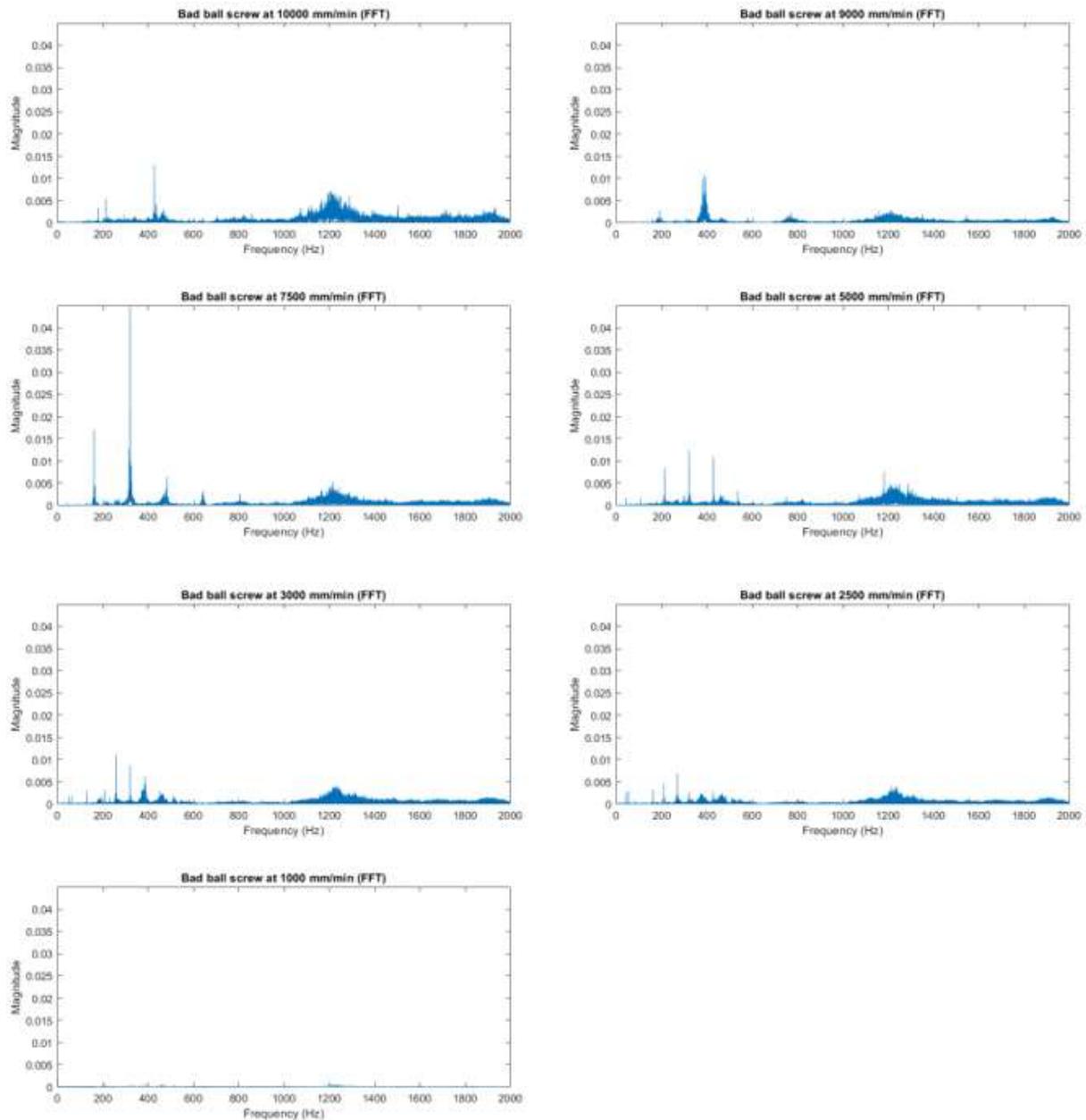


Figure 5.21 FFT of vibration signals at different feed rates for worn ball screw

The situation persists even when the frequency axis is normalized to reflect the cyclic nature of the signal (Figure 5.22). This can be a problem when using classical methods for assessing ball screw conditions, as it will be difficult, or maybe impossible, to set a baseline or threshold on the good data. For this reason, this research has proposed machine learning and deep learning strategies for evaluating the condition of CNC machine tool ball screw since they are more suited to identifying patterns within random datasets.

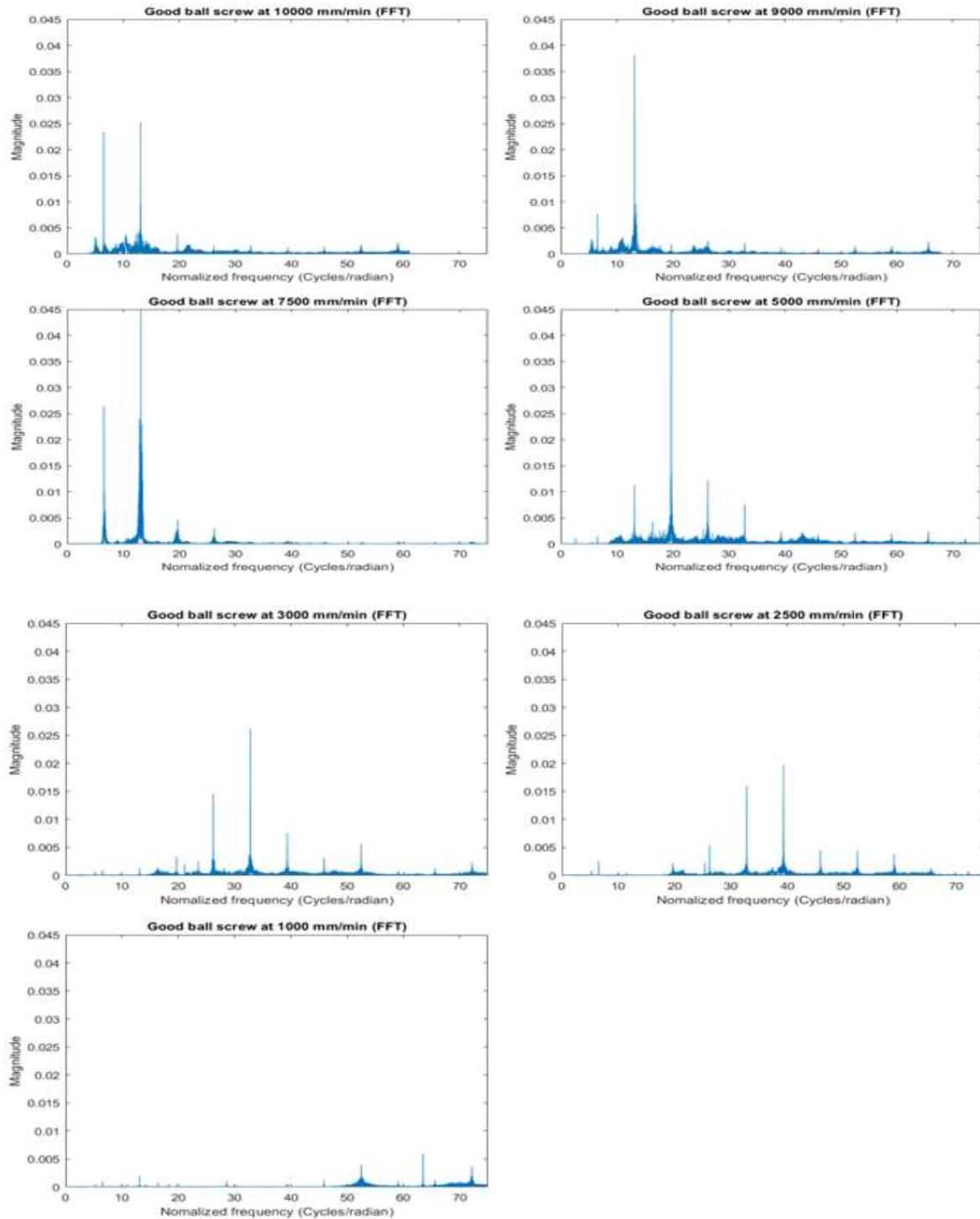


Figure 5.22 FFT of vibration signals at different feed rates for good ball screw with normalized frequency

5.3.3 AE sensor data

The AE sensor data from the good and worn ball screws at two different feed rates (1000 mm/min and 10000 mm/min) are shown in Figure 5.23, and the corresponding FFTs shown in Figure 5.24. This gives a first glance at the AE signal in the time and frequency

domain, respectively, with both axes set to auto-scale. A zoomed-in version of the FFT diagram is presented in Figure 5.25 that shows the first 2 kHz frequency range for the AE sensor data of a good ball screw at 10000 mm/min feed rate. The FFT diagrams show that the AE sensor is not able to detect frequency components in the low-frequency range. This is probably because the sensor's frequency response is in the frequency range of 100 kHz to 450 kHz. The FFT diagrams also show that there is no significant high frequency detected. The AE sensor data will therefore not be investigated any further in this research since its frequency response is higher and far removed from the frequency of vibration of the good and worn ball screws under investigation. This is not to say that these sensors cannot be used to detect other faults. Further research is recommended to identify the most suitable sensors for different types of faults to apply the analysis methods in this thesis.

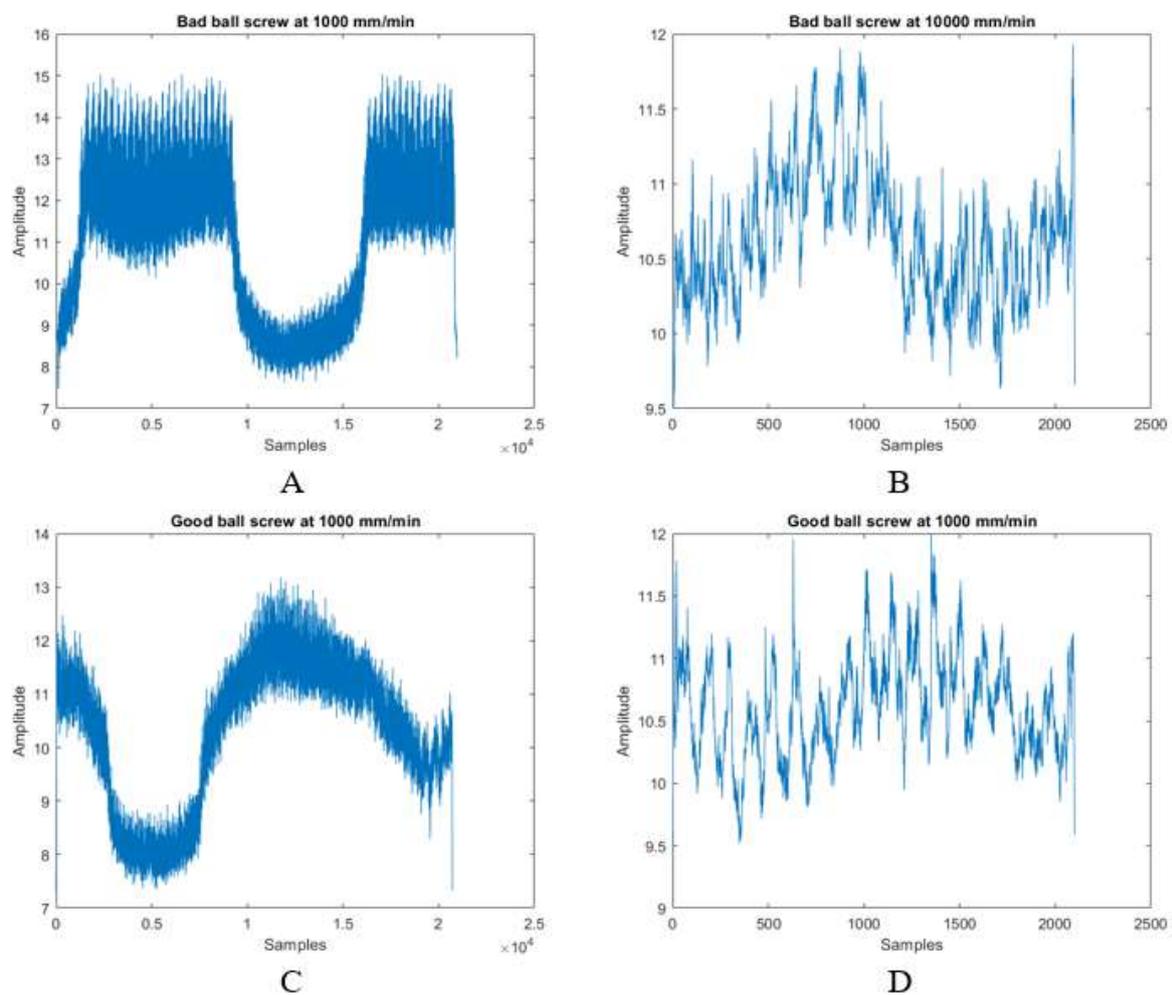


Figure 5.23 AE sensor data of worn and good ball screw at 1000/10000 mm/min

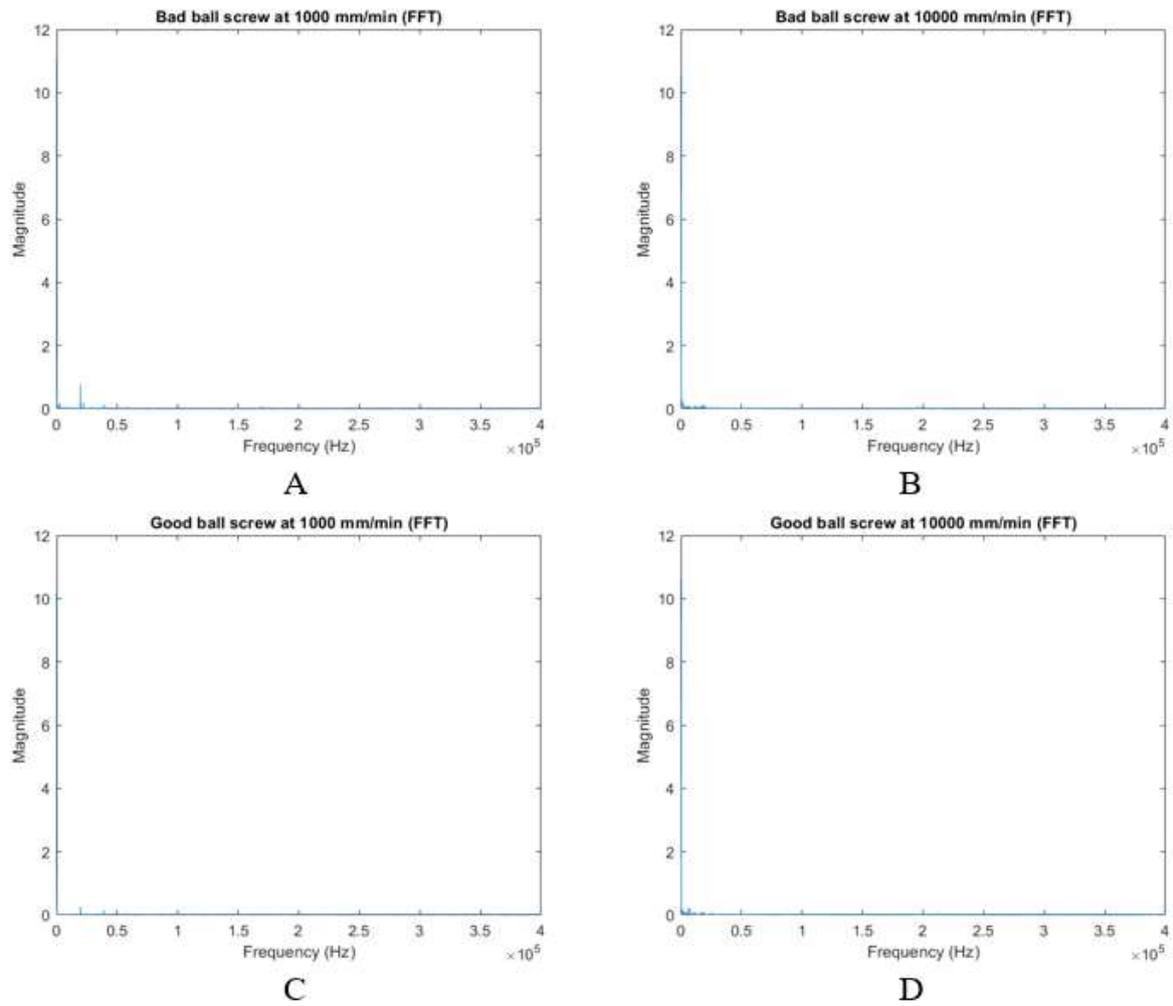


Figure 5.24 FFT of AE sensor data of worn and good ball screw at 1000/10000 mm/min

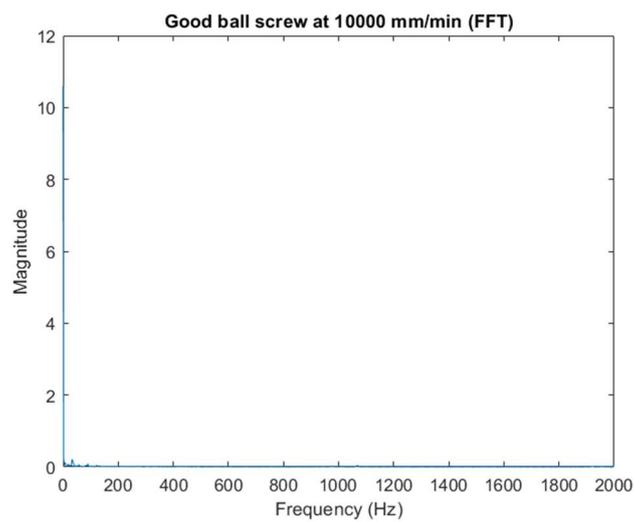


Figure 5.25 FFT of AE sensor data of good ball screw at 10000 mm/min for the first 2 kHz frequency range

5.4 Chapter summary

The datasets used in this research were obtained from experiments carried out at the CPT machine tool workshop at the University of Huddersfield. In all, a total of nine datasets were acquired from nine experiments for use in this thesis. Each experiment varied in set-up to accommodate the different parameters and factors being tested. Experiment_1 was the baseline test, and every other test was altered slightly from this baseline. Experiment_2 had the sensor axes misaligned with those of the machine tool. Experiments 3 to 7 had initial thermal perturbations of between 5 °C and 25 °C. This was done to investigate the influence of external disturbance on the system. Experiment _8 had quasi-random feed rates, similar to the initial feed rate of experiment_1 to avoid any aliasing of data. Finally, experiment_9 used an AE sensor in place of the vibration sensor to investigate whether this would provide additional data for analysing the particular fault. The next chapter delves into the discussion and validation of the obtained results of the PhD research.

Chapter 6 Results and validations

This chapter presents the results and validation of the proposed concepts and algorithms used in this research. It starts by selecting an optimum wavelet transform used for data processing by examining all the available mother wavelets in MATLAB using a brute force approach. Next, the validation of the wavelet and FFT features by comparing against other more commonly used but less robust feature types. This is followed by the implementation of the proposed machine learning and deep learning approaches. Then the validation of the deep learning algorithm using different datasets.

6.1 Selection of optimal wavelet

In order to achieve the proposed process of machine learning described in section 5.4, wavelet transform was used. There are many different types of wavelet transform, as shown in Table 3.1. The number after the wavelet name represents the number of vanishing moments [173, 174]. It means that db1 is a Debauchies wavelet with one vanishing moment and so on. The vanishing moment is significant because it indicates the points of a signal where the wavelets coefficients are zero. This occurs in the interval where the signal exhibits behaviours consistent with a polynomial of degree N , and the wavelet has N vanishing moments. All the wavelets are assessed on their first-level detail coefficients. To justify the choice of wavelet to use, each wavelet transform is used to process dataset_1 before training and testing with machine learning, and the results are shown in Table 6.1. The results in this table are the accuracy that each machine learning algorithm has achieved after training and testing when a particular wavelet transform is applied for data processing before feature extraction. The wavelet transform used is the coefficient is the first level detail coefficient (D1), and the extracted features were: mean, maximum, minimum, root mean square (RMS), kurtosis, and skewness.

After extracting the features, 85 % were used to train the network and 15 % to test the trained network.

Table 6.1 Wavelet validation results

Wavelet features	Fine Tree	Medium Tree	Linear SVM	Quad SVM	Cubic SVM	Coarse Gaussian SVM	Fine KNN	Medium KNN	Cubic KNN
bior11	95	95	95	95	98	74	99	96	95
bior13	95	95	95	95	98	70	99	97	95
bior15	96	96	96	97	97	72	98	96	94
bior22	98	98	98	92	95	69	100	100	100
bior24	99	99	99	96	97	67	100	99	99
bior26	98	98	98	95	97	69	100	100	99
bior28	98	98	98	93	97	70	100	100	100
bior31	97	97	97	94	97	71	98	96	94
bior33	99	99	99	94	97	67	97	95	92
bior35	97	97	97	93	96	73	97	96	94
coif1	96	96	96	92	97	66	100	100	100
coif2	97	97	97	96	96	70	100	100	99
coif3	98	98	98	90	97	71	99	99	99
coif4	99	99	99	91	96	68	100	100	99
coif5	99	99	99	96	97	69	100	100	99
db1	97	97	97	95	96	72	97	95	94
db2	99	99	99	88	97	71	99	99	99
db3	98	98	98	99	97	67	100	99	99
db4	99	99	99	93	93	68	100	100	100
db5	100	100	100	94	96	67	100	99	99
db6	98	98	98	97	97	66	100	99	99
db7	96	96	96	89	96	69	100	100	99
db8	96	96	96	92	97	68	100	100	99
db9	98	98	98	91	97	70	100	98	98
db10	97	97	97	89	97	68	100	100	100
Dmey	98	98	98	95	98	69	100	99	98

fk14	99	99	99	94	98	71	100	99	99
fk18	97	97	97	95	96	73	100	99	100
fk22	98	98	98	89	96	70	100	99	98
fk4	99	99	99	98	95	69	100	100	99
fk6	99	99	99	98	99	71	100	100	100
fk8	98	98	98	88	99	69	100	100	99
Haar	98	98	98	97	98	72	99	97	95
rbio11	98	98	98	96	97	75	99	97	97
rbio13	98	98	98	91	97	72	97	95	93
rbio15	97	97	97	94	96	74	98	96	95
rbio22	98	98	98	94	97	69	100	100	100
rbio24	97	97	97	91	97	67	100	100	100
rbio26	98	98	98	95	96	70	100	100	100
rbio28	99	99	99	95	97	67	100	100	100
rbio31	97	97	97	95	96	72	98	97	94
rbio33	96	96	96	96	97	69	98	97	96
rbio35	96	96	96	95	97	75	99	96	96
sym2	99	99	99	91	97	72	100	100	99
sym3	99	99	99	93	96	69	100	99	98
sym4	99	99	99	94	97	71	99	100	99
sym5	98	98	98	97	99	72	100	99	98
sym6	98	98	98	88	97	71	99	99	99
sym7	99	99	99	94	98	71	100	100	99
sym8	99	99	99	96	93	65	100	99	99

Table 6.2 shows the best-performing wavelet features that consistently perform across all the tested machine learning algorithms in the top 50 percentile. The results show that wavelet performance is independent of their family or its vanishing moments for this particular problem domain. So unlike other papers [75, 156], it is not recommended to use one without proper assessment arbitrarily. The same analysis can also be used to generate the worst-performing

wavelet features, as shown in Table 6.3. The criteria for this are if there are at least four results with accuracy less than the threshold. These are highlighted in red.

Table 6.2 Best performing wavelet features

Wavelet features	Fine Tree	Medium Tree	Linear SVM	Quad SVM	Cubic SVM	Coarse Gaussian SVM	Fine KNN	Medium KNN	Cubic KNN
fk14	99	99	99	94	98	71	100	99	99
fk6	99	99	99	98	99	71	100	100	100
haar	98	98	98	97	98	72	99	97	95
rbio11	98	98	98	96	97	75	99	97	97
sym4	99	99	99	94	97	71	99	100	99
sym5	98	98	98	97	99	72	100	99	98
sym7	99	99	99	94	98	71	100	100	99

Table 6.3 Worst performing wavelet features

Wavelet features	Fine Tree	Medium Tree	Linear SVM	Quad SVM	Cubic SVM	Coarse Gaussian SVM	Fine KNN	Medium KNN	Cubic KNN
bior11	95	95	95	95	98	74	99	96	95
bior13	95	95	95	95	98	70	99	97	95
bior15	96	96	96	97	97	72	98	96	94
bior33	99	99	99	94	97	67	97	95	92
coif1	96	96	96	92	97	66	100	100	100
db8	96	96	96	92	97	68	100	100	99

Also, by visual inspection, it can be observed that the accuracy score for entries in Table 6.3 is mostly equal to or above 94 % except for the coarse gaussian machine learning algorithm and a few others. This implies that most of the wavelet transforms shown in Table 6.1 will satisfy the proposed machine learning algorithm. However, wavelets shown in Table 6.2 are most highly recommended because they show consistently better performance. For this research,

sym5 is selected even though any of the other candidates from Table 6.2 are equally viable. This selection is because the symlet wavelet family has three entries in Table 6.2, which implies that it might have more stability than the other tested wavelet families. Among the three symlet entries, sym5 performs better than the other two in coarse Gaussian SVM, where they all performed poorly. On the other hand, rbio11 can be selected since it outperforms the rest in coarse Gaussian SVM, where all seven entries performed poorly.

Similarly, an analysis was performed to determine the level of decomposition of sym5 that would give the best overall feature performance. This was done by assessing the first three levels of detail coefficients of sym5 wavelet transforms. The result is shown in Table 6.4 and presented in a bar chart (Figure 6.1) for more straightforward visualisation. It is clear that the detail coefficient from level 1 is more robust than that of levels 2 and 3 since it consistently has a higher accuracy for all of the tested algorithms. This could be because more of the lower frequency content is lost as the level of decomposition increases. Although the lower frequency content is dominated by a lower signal-to-noise ratio, it also contains useful signals. So, at decomposition levels 2 and 3, more of the useful signals will have been lost than level 1.

Table 6.4 Accuracy of level 1,2 and 3 detail coefficients of sym5

ML Algorithm	D1 Wavelet-FFT	D2 Wavelet-FFT	D3 Wavelet-FFT
	Features	Features	Features
Fine Tree	98.5	97.7	98.8
Medium Tree	98.5	97.7	98.8
Linear SVM	97.9	70.5	86.1
Quad SVM	99.2	98.7	99.8
Cubic SVM	98.5	91.4	99.2
Coarse Gaussian	98.5	70.9	69.5
Fine KNN	99.2	100.0	99.8
Medium KNN	98.5	99.8	99.4
Cubic KNN	98.8	99.8	99.2

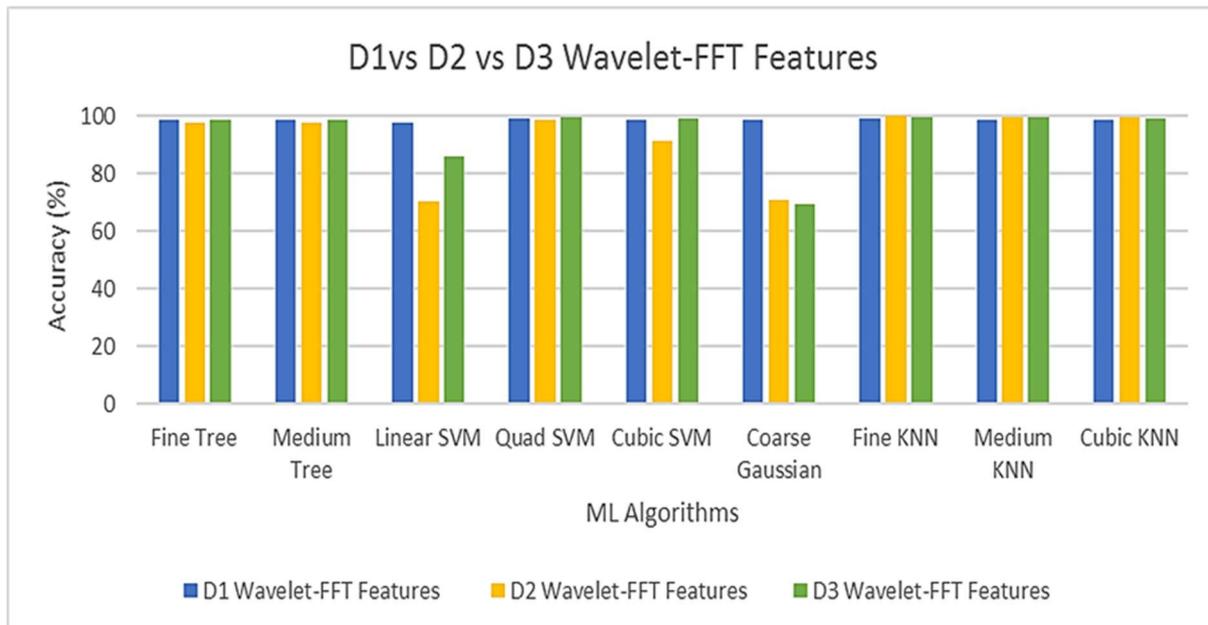


Figure 6.1 Bar chart of the accuracy of levels 1,2 and 3 detail coefficients of sym5

6.2 Validation of wavelet and FFT features

Another set of features was extracted using direct statistics (kurtosis, mean, mode, skewness, variance) from the original raw data to assess the performance of the wavelet and FFT features. These features are used with several machine learning algorithms (see Table 6.5) to classify the healthy and worn ball screws. In the machine learning algorithms, 85 % of the extracted features were used to train the network and 15 % to test the trained network. The results are presented in Table 6.5, comparing wavelet and FFT features against traditional statistical features corresponding to different machine learning algorithms.

Table 6.5 Machine learning classification accuracy

Machine Learning Algorithms	Statistical Features Accuracy (%)	Wavelet and FFT Features Accuracy (%)
Fine Tree	94.8	98.5
Medium Tree	94.8	98.5
Linear SVM	87.6	97.9
Quadratic SVM	93.9	99.2
Cubic SVM	94.8	98.5
Coarse Gaussian SVM	88.2	98.5

Fine KNN	96.6	99.2
Medium KNN	92.4	98.5
Cubic KNN	91.8	98.8

It can be observed from Table 6.5 that higher classification accuracies were achieved with wavelet and FFT features (above 97 %) compared to the statistical features, which were mostly below 95 %. This result is reflected in the scattered diagram when two variables from both feature sets are plotted against each other (for example, kurtosis vs mean). It can be observed that the two variables have a lot less discrimination between them in Figure 6.2 compared to Figure 6.3, hence leading to better separation in the latter than the former.

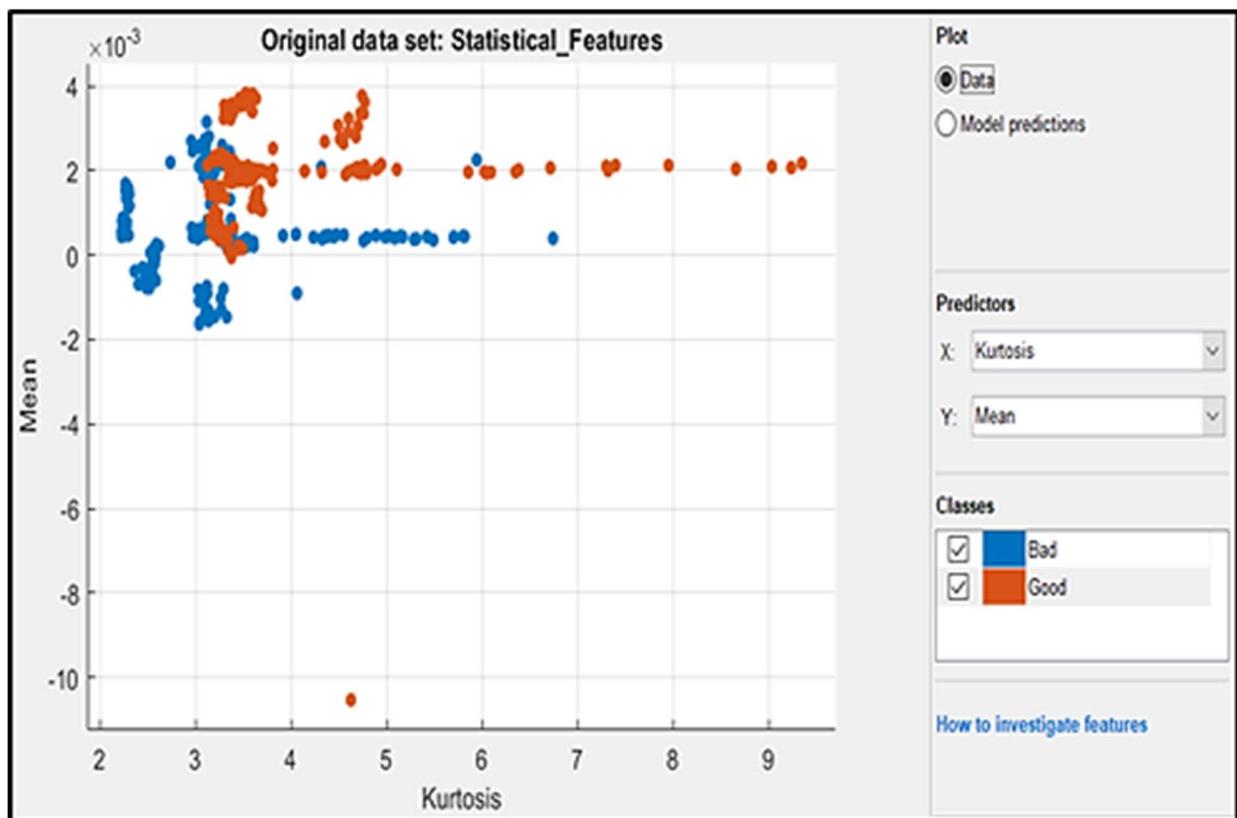


Figure 6.2 Wavelet and FFT features scatter plot of Kurtosis vs mean

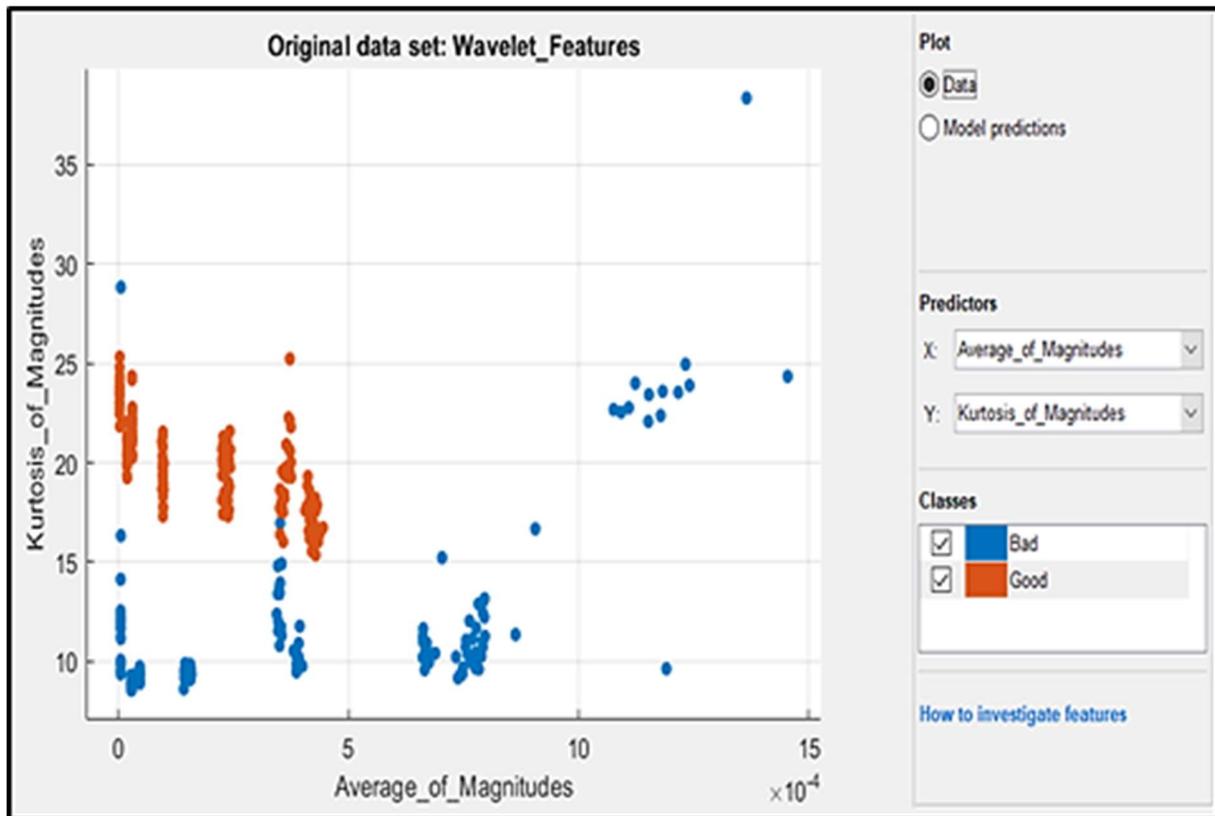


Figure 6.3 Wavelet and FFT features scatter plot of Kurtosis of magnitude vs Average of magnitude

Further analysis is done by comparing the first level decomposition of wavelet and FFT features (D1 wavelet-FFT features) with features derived from traditional methods (statistical features), wavelets alone (D1 Coefficient features), and FFT alone (FFT features). The result is presented in Table 6.6. The same result is presented in a bar chart for easy visualisation in Figure 6.4. It can be observed that the wavelet-FFT features remain stable or robust across different machine learning algorithms, unlike the rest. Statistical features perform reasonably well, as in most cases outperform the other two. Wavelets alone appear to be the least robust, with FFT alone just outperforming them. Figure 6.4 clearly shows the need to complement wavelets with FFT features for this application to develop an algorithm that will perform better than the traditional statistical method.

Table 6.6 Classification accuracy from multiple feature type

ML Algorithm	Statistical Features	D1 Wavelet-FFT Features	D1 Coefficients Features	FFT Features
Fine Tree	94.8	98.5	86.7	93.3
Medium Tree	94.8	98.5	86.7	93.3
Linear SVM	87.6	97.9	62.9	71.4
Quad SVM	93.9	99.2	62.9	85.3
Cubic SVM	94.8	98.5	52.8	82.0
Coarse Gaussian	88.2	98.5	62.9	83.2
Fine KNN	96.6	99.2	66.7	100.0
Medium KNN	92.4	98.5	70.5	100.0
Cubic KNN	91.8	98.8	71.4	100.0

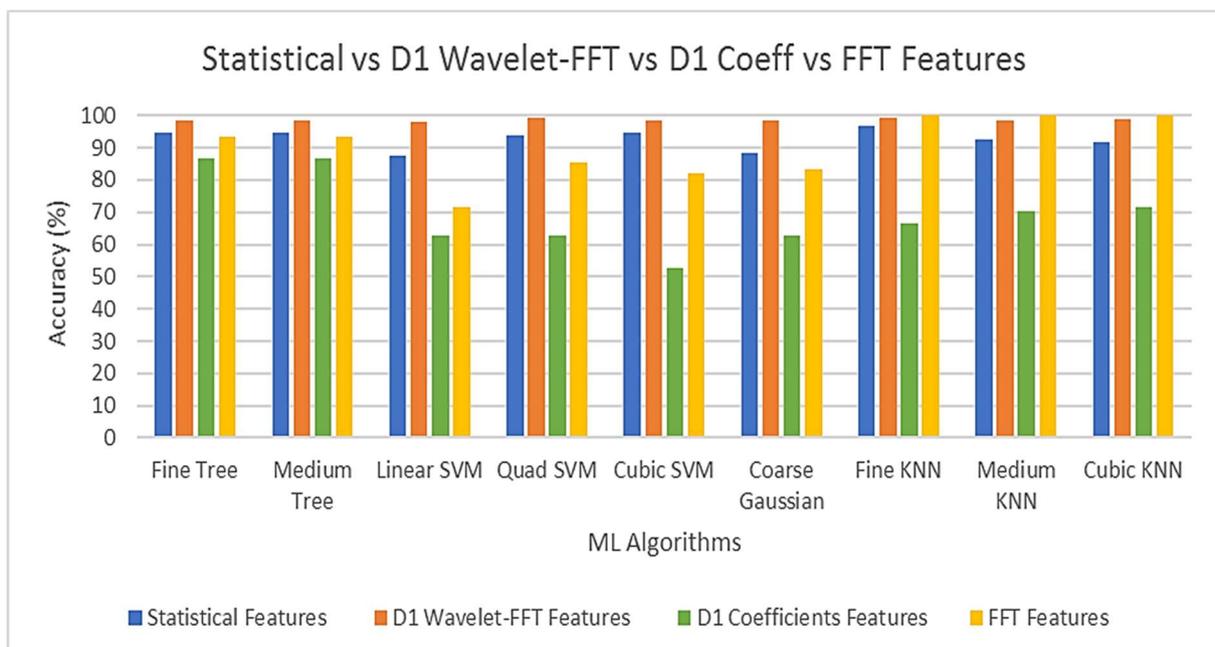


Figure 6.4 Bar chart of classification accuracy from multiple feature type

6.3 Application of machine learning

In order to apply machine learning, the data is first processed with wavelet and FFT, and then useful features are extracted for the machine learning process. The wavelet transform used is

the sym5 wavelet, and the coefficient is the first level detail coefficient (D1). The extracted features were: mean, maximum, minimum, root mean square (RMS), kurtosis, and skewness. After extracting the features, 85 % were used to train the network and 15 % to test the trained network; a split was recommended to ensure good quality in the validation phase [84]. This is set up using the classification learner application in MATLAB [175]. The results obtained are shown in Table 6.7, using dataset_1. On the one hand, the former shows the results when only the kurtosis and skewness features are used. In this instance, some machine learning algorithms perform very well (fine KNN and fine tree), while others perform not so well (coarse Gaussian SVM and quadratic SVM).

On the other hand, the latter shows the results when all the extracted features are used. It can be observed that the accuracy is a lot more consistent and around the range of 98 % to 99 %. This shows robustness across multiple machine learning algorithms. A comparison of the results from Table 6.7 is presented in Figure 6.5.

Table 6.7 Machine learning algorithms accuracy for Kurtosis and skewness features

Machine Learning Algorithms	Kurtosis and skewness Features Accuracy (%)	All Features Accuracy (%)
Fine Tree	99.0	98.5
Medium Tree	99.0	98.5
Linear SVM	90.5	97.9
Quadratic SVM	90.5	99.2
Cubic SVM	95.2	98.5
Coarse Gaussian SVM	89.5	98.5
Fine KNN	100.0	99.2
Medium KNN	96.2	98.5
Cubic KNN	95.2	98.8

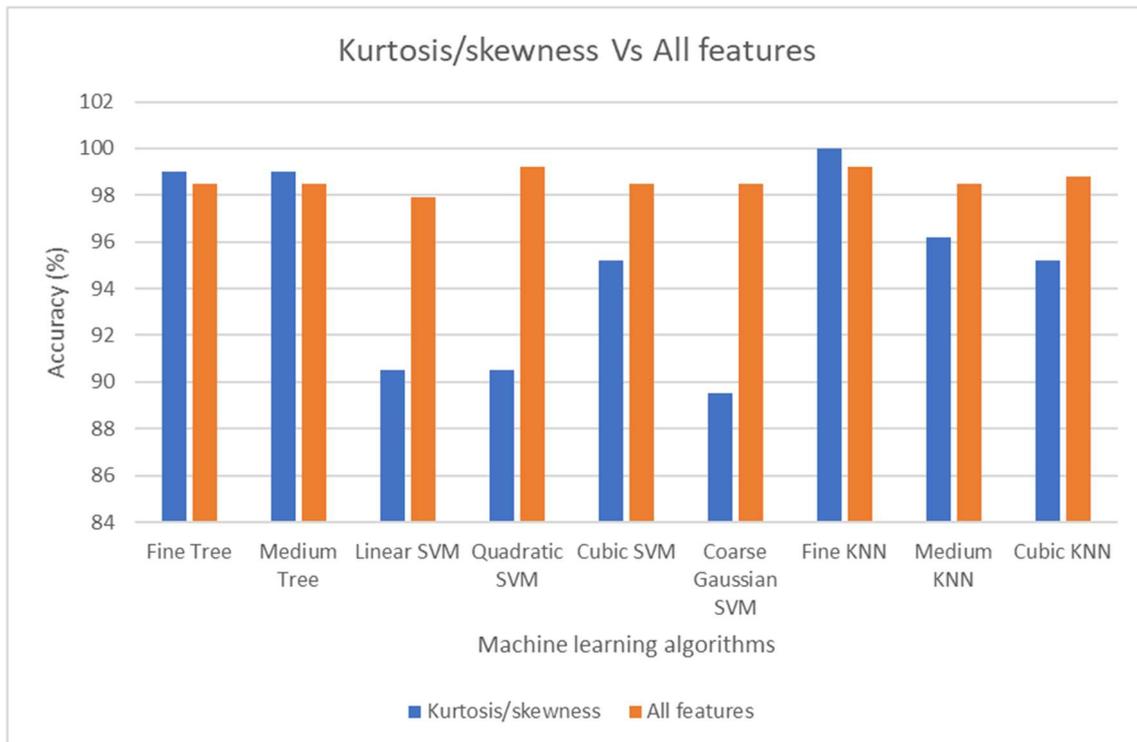


Figure 6.5 Comparison of results for kurtosis/skewness vs all features.

6.4 Application of the DCNN

A convolution neural network works by utilising feature maps, which are generated by filters in the convolution layers. Each filter is responsible for convolving a specific section on the feature map from a previous convolution layer or the input (in the first convolution layer). The first convolution layer is responsible for capturing edges, corners, and colours. The second convolution layer is responsible for capturing texture and mesh patterns. These patterns are combined and populated in the feature space by the subsequent convolution layer (fully connected) to form a complex invariance of the input image.

In order to apply time-series data to DCNN, it is required to present the data in such a format that it is acceptable by the DCNN. Also, it is necessary to optimise the format in which the data is presented to obtain the best possible result. The steps taken to achieve this is discussed in the next section.

6.4.1 Data processing for DCNN

Applying DCNN to dataset 1, the raw dataset is first converted to image files, which is the input format for DCNNs. A sample plot is shown in Figure 6.6 of the vibration data of the worn ball screw at 1000 mm/min feed rate. The plot includes the axes and grids that account for the magnitude and the feed rate as additional input features to the DCNN. Figure 6.7 shows the other feed rates (2500, 3000, 5000, 7500, 9000, 10000 mm/min) and how they compare to one another.

Another sample plot is shown in Figure 6.8, and it does not contain axes or grids. This is another possible way to format the input image data. This implies that the magnitude of vibration and the feed rate is not part of the input features. Figure 6.9 and Figure 6.10 show a contrast between vibration data at 10000 mm/min and 1000 mm/min with and without fixed grids and axes, respectively. Figure 6.11 and Figure 6.12 show the same contrast but at all the selected experimental feed rates. Both formats give similar results, with the former having better prediction capability (Appendices J to N). This shows that adding the grids and axes as part of the input feature does improve the results compared to when not included (this is discussed further in section 6.5).

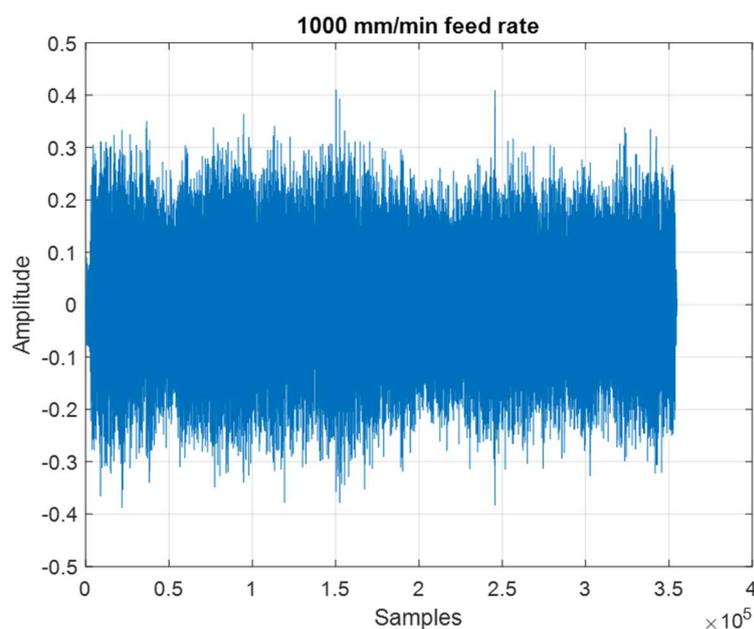


Figure 6.6 Vibration data at 1000 mm/min feed rate with axes and grids

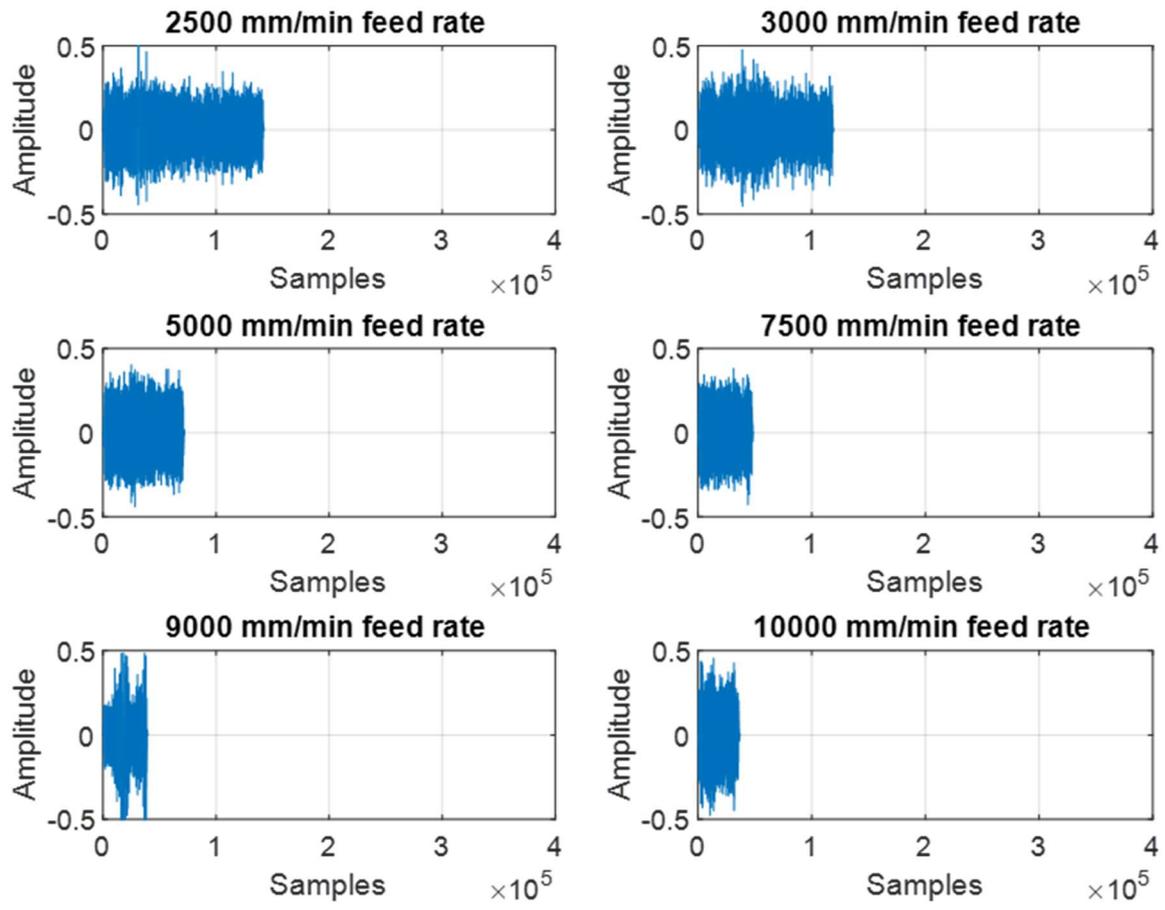


Figure 6.7 Vibration data at different feed rates with axes and grids

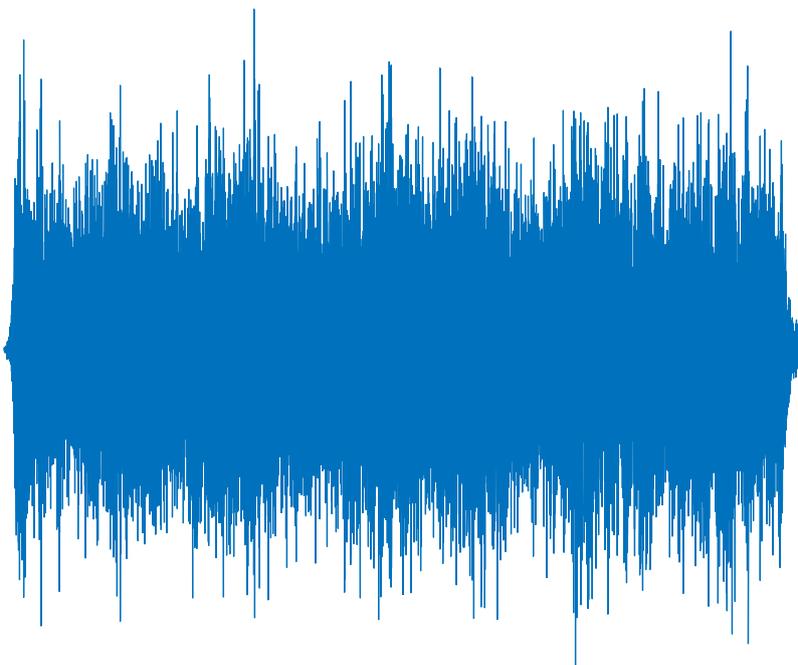


Figure 6.8 Sample plot of a vibration signal without axes and grids

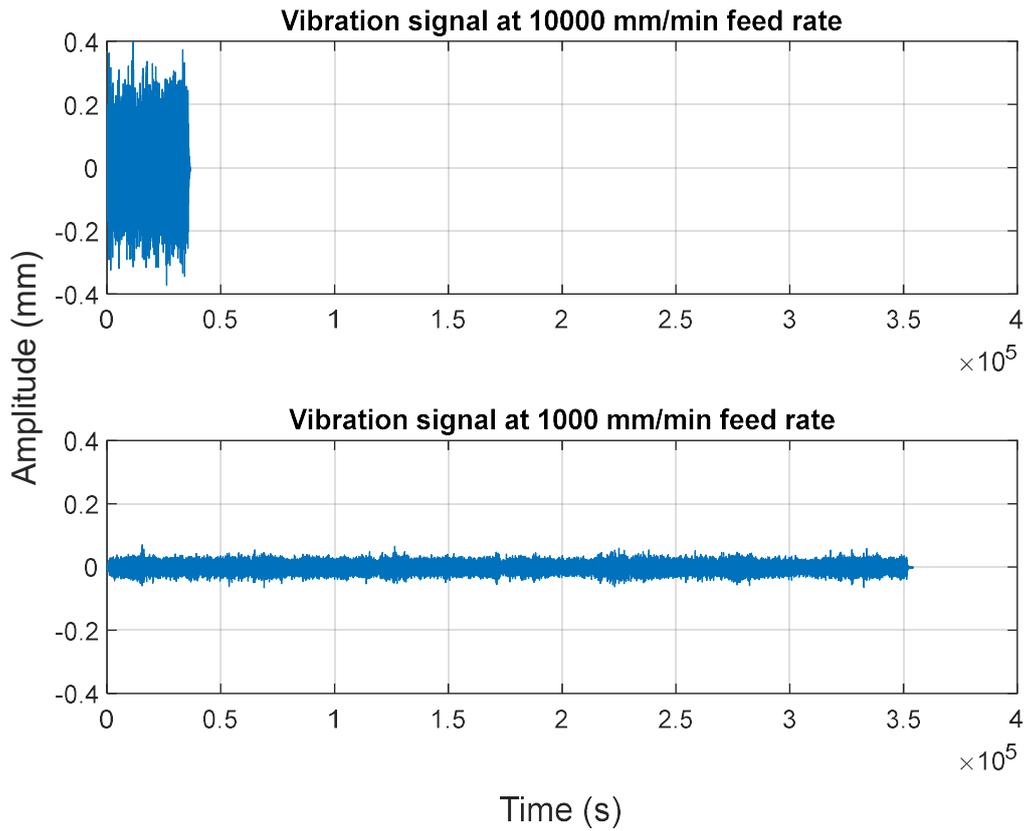


Figure 6.9 Vibration signal with fixed axes and grids

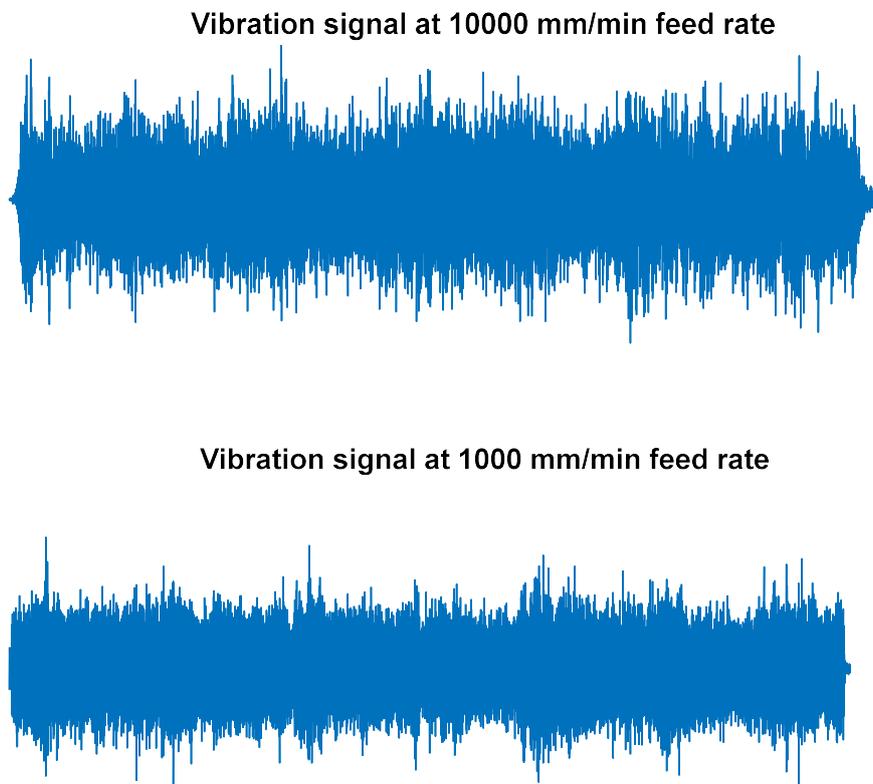


Figure 6.10 Vibration signal without axes and grids

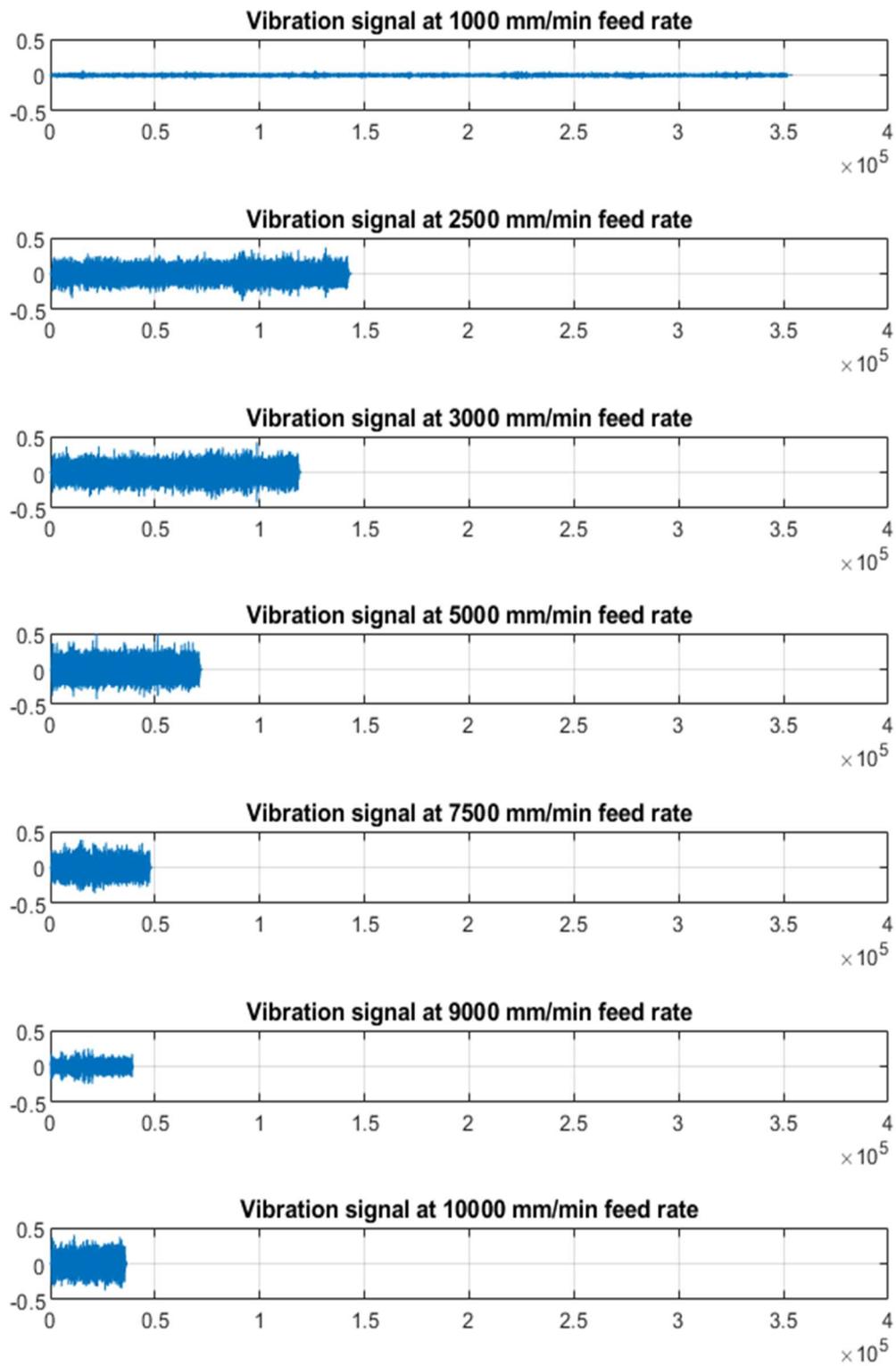


Figure 6.11 Vibration signal of all feed rates with fixed axes and grids

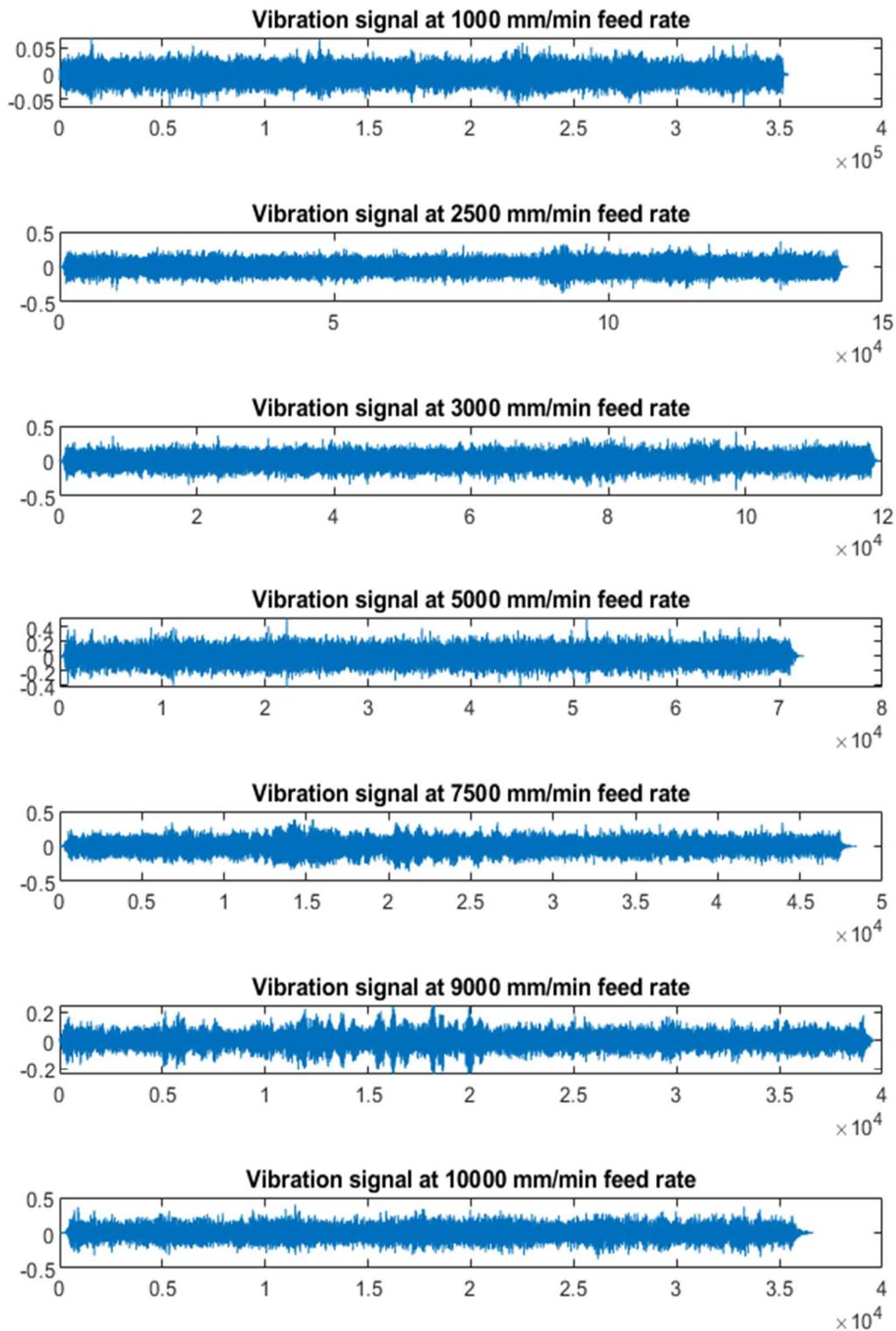


Figure 6.12 Vibration signal of all feed rates without fixed axes and grids

6.4.2 Results from training with DCNN

The convolution and fully connected layers of the DCNN algorithm are used to extract features internally. The first convolution layer initiates the feature extraction by detecting edges and

colours from various angles. Later convolution layers and the fully connected layer perform advanced combinations of the features learnt in the preceding layers. The last feature extraction activity, which is utilised for classification, is carried out in the third fully connected layer.

The DCNN network is trained with 80 % of the data, and 20 % is used for validation. This results in an accuracy of 94 % (see Figure 6.13) [2].

bad	38	4
good	1	41
	bad	good

Figure 6.13 DCNN confusion matrix showing 94 % accuracy

On further improvement of the DCNN algorithm (Appendix A) by making the partitioning into training and validation dataset more random, the DCNN is able to attain an accuracy level of 98.8 %. This is due to the fact that there is a lot more homogeneity across the different datasets. Less homogeneity across the datasets will result in the DCNN overfitting on the training set, which would affect the test set's accuracy. The training progress and the confusion matrix are shown in Figure 6.14 and Figure 6.15, respectively.

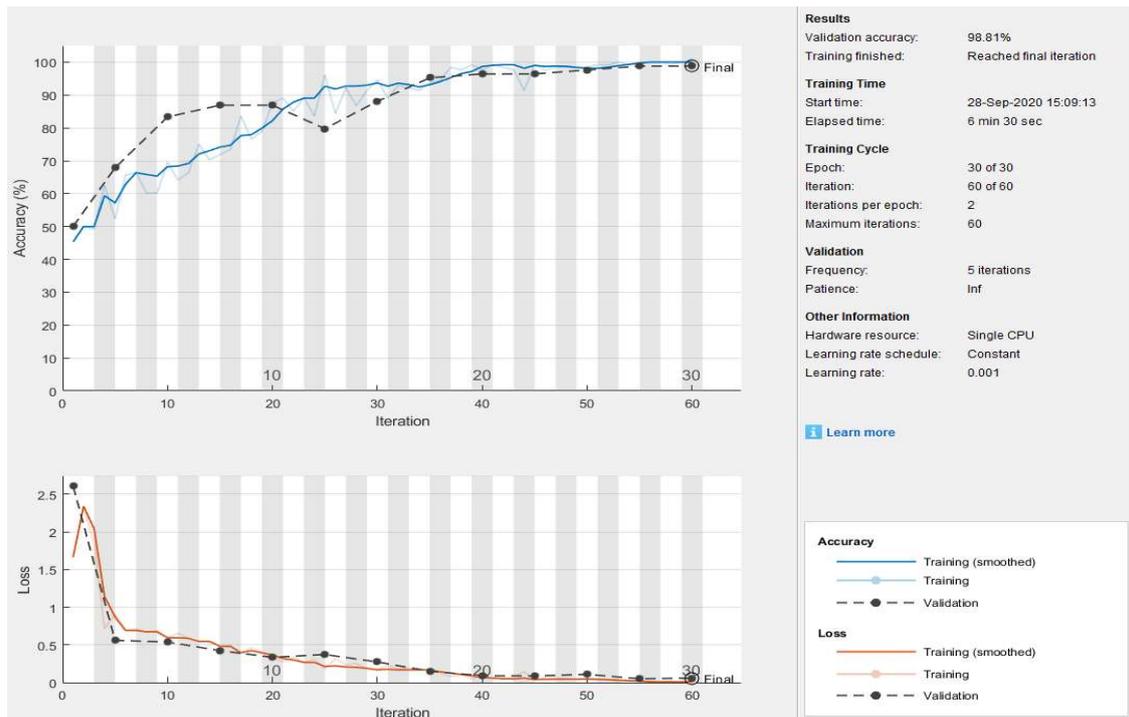


Figure 6.14 Training progress for raw data

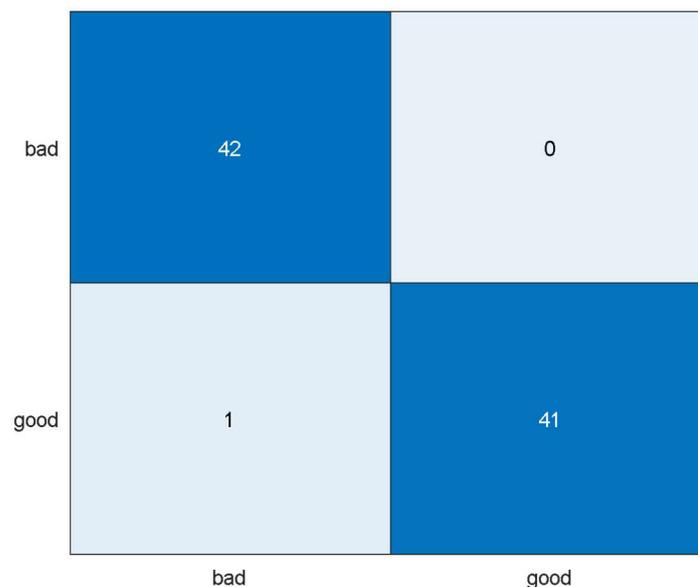


Figure 6.15 DCNN confusion matrix showing 98.8 % accuracy

The input to the DCNN could also be processed to remove the noise by presenting it in the frequency domain as a plot of the Fourier transform (Appendix O). For this, the normalised version of the FFT plot is used to train the DCNN network. This is able to achieve an accuracy of 100 %, as shown in the training progress (Figure 6.16) and confusion matrix (Figure 6.17). This shows an improved performance compared to when the raw data is used. It is also clear

from the training process (Figure 6.16) that the trained network converges quicker than when unprocessed data is used (Figure 6.14), indicating that the network does not over-train on the input data. Further tests will be conducted in section 6.5 using all the different datasets collected in section 5.2.



Figure 6.16 Training progress for processed data

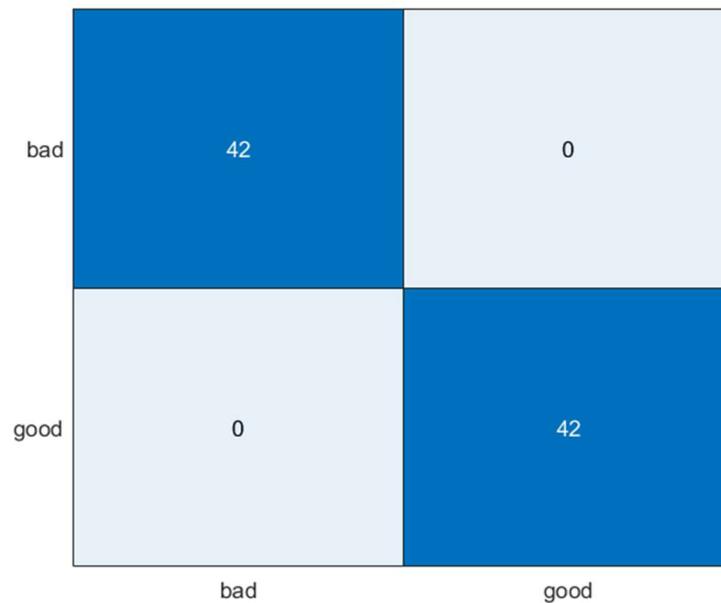


Figure 6.17 DCNN confusion matrix showing 100 % accuracy

6.5 Validation of DCNN

6.5.1 Comparing the different datasets

Validation of the DCNN is done by accessing the robustness (consistency) of the algorithm's performance on varied datasets. These datasets used were described in section 5.2, and they were developed under different conditions. These conditions include misaligned axes (dataset 2), the introduction of thermal perturbations (dataset 3 to dataset 7), and random feed rates (dataset 8). The reasons for these conditions are highlighted below;

1. Misalignment: to evaluate the effect of sensor orientation.
2. Thermal perturbation; since heat accounts for about 70 % of errors in a machine tool [176, 177], the ability of DCNN to perform accurately within this environment will demonstrate its robustness and practicality as heat is a necessary by-product of a machine tool in operation.
3. Random feed rates: to ensure that there is no harmonic effect on the results due to the feed rate being a perfect round number.

The results obtained from the datasets are shown in Table 6.8 (see appendices B to I for the training progress and confusion matrix).

Table 6.8 DCNN results from several datasets to classify good and worn ball screw

	Training set (%)	Test set (%)	Accuracy (%)	Number of misclassifications
Misaligned sensor axis test	90	10	100.0	0
	85	15	98.3	1
	80	20	98.8	1
	70	30	97.6	3
	60	40	100.0	0
5 °C thermal perturbation test	90	10	97.6	1
	85	15	98.3	1
	80	20	100.0	0
	70	30	98.4	2
	60	40	98.2	3
	90	10	100.0	0

10 °C	85	15	98.3	1
thermal	80	20	100.0	0
perturbation	70	30	97.6	3
test	60	40	99.4	1
	90	10	100.0	0
15 °C	85	15	100.0	0
thermal	80	20	97.6	2
perturbation	70	30	99.2	1
test	60	40	98.2	3
	90	10	100.0	0
20 °C	85	15	100.0	0
thermal	80	20	100.0	0
perturbation	70	30	99.2	1
test	60	40	97.6	4
	90	10	100.0	0
25 °C	85	15	100.0	0
thermal	80	20	98.8	1
perturbation	70	30	99.2	1
test	60	40	98.8	2
	90	10	100.0	0
Non-	85	15	98.3	1
centuple	80	20	100.0	0
feed rate	70	30	99.2	1
test	60	40	100.0	0

The training progress charts and the confusion matrices for the results in Table 6.8 are shown in appendices B to H. Also, the analysis done in Table 6.8 is repeated with the processed data (Appendix O), and the result obtained in all the above scenarios was 100 %. The training process and confusion matrix of all the scenarios are presented in appendices P to U.

6.5.2 Comparing the effect of fixed axes and grids

Furthermore, an analysis is done on the experiment data to assess the impact of using input data that are plotted on fixed axes and grids compared to those without fixed axes and grids. To do this, a fixed section of each input data is used to classify the data according to the feed rate. This implies seven output classes since there are seven different feed rates. The use of the section as opposed to the entire signal was to remove any indication of the feed rate from the

input data since the signal span is consistent across all the inputs. The sample plot at different feed rates of the data with and without grids and axes is shown in appendix J. The choice to assess machine tool feed rate (instead of ball screw condition) is to show that the DCNN algorithm is robust in terms of classification problems. A summary of the results is presented in Table 6.9 (see appendices K to N for the training progress, confusion matrix, and comparison of the results for different datasets).

Table 6.9 DCNN results to classify feed rates of different datasets

	Training set (%)	Test set (%)	Accuracy – fixed axes/grids (%)	Accuracy – no axes/grids (%)
Baseline test	90	10	97.6	80.9
	80	20	95.2	86.9
	70	30	91.2	86.5
	60	40	89.2	75.6
Misaligned sensor axis test	90	10	95.2	76.1
	80	20	97.6	88.1
	70	30	94.4	80.9
	60	40	90.4	77.3
5 °C thermal perturbation test	90	10	92.8	90.4
	80	20	89.2	82.14
	70	30	88.9	76.1
	60	40	89.8	73.2
Baseline testdatabase	90	10	91.7	86.5
	80	20	92.5	77.6
	70	30	92.5	88.3
	60	40	93.2	84.5

6.6 Chapter summary

The datasets collected from the experiments were used to develop, test, and validate the proposed methodologies. Dataset_1 was used to develop and test the proposed machine learning and DCNN methodologies. While all the other datasets were used to validate the

proposed models. Different aspects of the methodologies were validated. These include selecting the optimal wavelet, the feature extraction process, and the DCNN accuracy.

From the machine learning model, at least 98.0 % accuracy is obtained when assessed with some machine learning algorithms like decision tree, SVM, and KNN. From the DCNN model, the accuracy obtained when tested on the raw data with no data processing is 98.8 %. This is a positive result as it shows the potential of this model and its suitability for automation and online real-time applications. However, with processed data, an accuracy of 100 % is obtained. The next chapter discusses the research conclusions and possible future work that could be embarked on to build on the outcome of this research.

Chapter 7 Conclusions and future work

7.1 Summary of the research work

This research is focused on the condition monitoring of the ball screw of a CNC machine tool using artificial intelligence. The input data used is vibration signals, and the output is a binary classification of “good” and “worn.” This vibration signal was collected from the two ball screws on a gantry-type CNC machine - one is good, and the other is worn. This was as a result of faulty lubrication system which resulted in the restriction of the normal lubrication to one of the ball screws. During the data collection process, the gantry was made to travel along the Y-axis at different feed rates within the normal working condition of the machine tool. Both machine learning and deep learning were used to classify the condition of the ball screws. The machine learning approach implemented an optimised feature extraction technique, whereas the deep learning approach implemented convolution-type deep learning where the inputs are image data taken from measurement signals. The exploration of this use of the image data, rather than raw data signals, has not previously been used for this type of engineering application.

The validation process was initiated by selecting and confirming the choice of wavelet transform to be used in the data processing for the machine learning-based technique through MATLAB simulation (see section 6.1). This led to the comparison (based on machine learning algorithms) of the wavelet and FFT features against those based upon wavelet alone, FFT alone, and statistical features. The DCNN was validated using multiple datasets with varied conditions that included axes misalignment and the introduction of thermal perturbation of varied magnitude. It was also shown to perform well on noisy time-series data achieving greater than 95 % classification accuracy.

This research has shown that DCNN can accurately predict the condition of a ball screw assembly in a binary condition (good/bad) scenario using image data taken from measurement signals. It also showed that similar results could be obtained with a machine learning approach on the time-series data using wavelet and FFT for data processing before feature extraction. However, the DCNN approach has potential advantages because less overhead is required since there is no need for data processing and feature extraction to achieve similar results as the machine learning approach. There is also reduced potential for human error, bias or preconceptions, in the feature selection process and data processing technique which would result in low classification accuracy. This advantage would mean that using the DCNN approach would not require a domain expert. Therefore, the method, tested on a binary condition in this thesis, could perhaps be applied to different fault conditions with relatively little human intervention. DCNN can also be easily adapted to in-line, real-time applications that further enable automation. Finally, this research showed that thermal perturbations, noise, or harmonic frequencies do not adversely affect the accuracy of the DCNN model.

7.2 Conclusions

The conclusions reached in this thesis are limited to the problem area and data type that could be obtained during the project. That is:

1. Vibration data from an accelerometer.
2. Ball screw condition monitoring based on the given binary classification.

The conclusions may, however, also be used as a foundational hypothesis for similar or broader problem areas or datasets. It is recommended that a detailed analysis, similar to what has been done in this thesis, be carried out before applying any of its conclusions to other domains.

These conclusions, in order of importance, are:

1. While traditional condition monitoring has been achieved using statistical and analysis tools, it has been shown that, for the given fault condition, DCNN can be used to classify the binary (good/bad) classes from an accelerometer from images of the raw time-series alone, with better than 95 % accuracy.
2. Ensuring that the images used in DCNN training and testing are of consistent scale improves the classification accuracy. Adding a grid and fixing the maximum and minimum values of the axis of the plot of the vibration data preserves information that the DCNN uses in its learning. This was tested by taking a section of the vibration data to classify the feed rate of the machine tool. The results (Table 6.9) show that when the feature of relative size is included in the input, the obtained classification accuracy is on average 10 % better than when this feature is excluded.
3. When DCNN is applied to images in the frequency domain, by using images of the FFT, it can achieve slightly higher classification accuracy (sections 6.4 and 6.5). While the results are improved, this approach does lose some of the benefits of being able to work on data that does not have to be pre-processed. Nevertheless, it strongly indicates that the method will be more generally applicable to other faults or problem domains.
4. The DCNN proves to be robust in application, as the classification accuracy did not diminish progressively with reduced size of the training data. This is presented in Table 6.8, which shows that the results are consistent regardless of the size of the training dataset between 60 % to 90 %.
5. The machine learning methodology used in this thesis can give classification results with similar accuracy to that of deep learning (with raw data). This is presented in Table 6.8 and Table 6.9. Both methods are able to achieve accuracy levels of 97 % and above. However, the machine learning approach is susceptible to human error during the feature extraction stage. This stage requires a domain expert, and any error,

preconceptions, or bias introduced at this stage will affect the quality of the result (Table 6.7, Table 6.8 and Figure 6.5).

6. The optimal choice of wavelet for machine learning is dependent on the type of data. This research showed that many different wavelets are suited for processing the vibration data presented in this thesis. This includes fk4, fk6, Haar, rbio11, sym4, sym5, and sym7 (see Table 6.2). This is contrary to Anoh et al. [165] that biorthogonal wavelets are more suited for image processing while the orthogonal wavelet is more appropriate for processing continuous signals. It is apparent from Table 6.1 that many biorthogonal wavelets like bior22, bior24, bior28, rbio11, rbio26, and rbio28 compare favourably to their orthogonal counterparts in processing the vibration data from ball screws of machine tools. Hence the best approach to selecting a wavelet for processing a particular signal would be adequate analysis of the results from different wavelets on the signal.
7. Condition classification for this failure mode is unaffected by temperature. Table 6.8 shows that even with the introduction of thermal perturbation of 5 °C, 10 °C, 15 °C, 20 °C, and 25 °C in experiments 3, 4, 5, 6, and 7 respectively, the results are still consistent with those of experiment_1 (the baseline test).
8. There is no discernible change in the temperature of the ballnut during operation for this failure mode. The temperatures of the ballnuts were monitored during the experiments (see Figure 5.11, Figure 5.14 to Figure 5.16) to observe if there was any differential deviation in the temperature of the worn and good ball screws. These figures show that the deviation in the temperature of the ball screw is minimal and relatively constant as both temperature rises (experiments 1, 2, 8, and 9) and falls (experiments 3, 4, 5, 6, and 7) uniformly and consistently. The temperature differs by 0.3 °C and is negligible in the context of the measurement uncertainty.

9. The sensitivity of alignment of the axes of the measuring sensor (accelerometer) with respect to the machine axes has a negligible effect on the DCNN results. This is illustrated in experiment 1 (baseline test with aligned axes) and experiment 2 (the same test as experiment 1, but with misaligned axes). The results from both tests indicate that there is no significant change in accuracy when the axes are misaligned (98.8 % accuracy when training dataset to test dataset ratio is 4:1). This could be because the orientation of the sensor does not affect its functionality, even though the amplitude might be reduced since it is less sensitive in the direction of interest. This is important because when replacing a sensor, there is no need to create a new training dataset even if the exact alignment of the original sensor cannot be replicated. Nonetheless, it is recommended to keep the axes of the accelerometer aligned to those of the machine tool within ± 5 degrees, which is achievable without using any specialist equipment.

7.3 Contribution to knowledge

This research has shown that DCNN can be used on image representation of continuous time-base signals as a method for generating condition monitoring models. The method works with raw data and is further improved when the FFT spectrum is used. The thesis also proves that the methodology is further enhanced by including the size (magnitude) information as one of the input features, which is done by fixing the scale of the plot. Other researchers can apply this technique in other domains where time-series data constitute a substantial part of their analysis. It also provides an opportunity for a deeper investigation into whether other aspects of image conversion (such as colour maps) might enhance the system further.

Thermal effect influences on machine tools justifiably have widespread attention because of its impact on manufacturing. This research has shown that for this fault condition, the effect of temperature is negligible. Researchers in this domain in the future can therefore concentrate their efforts on other aspects of their research rather than repeating the process of investigating

the temperature effect during their tests. This will be a significant saving in terms of efficiency and cost of doing the testing. It is recommended that temperature still be monitored, but the full measurement regime could be reduced by up to a fifth with this knowledge.

This research also showed that, in this problem area, the axis of the vibration sensor does not necessarily have to be closely aligned with that of the machine. Hence, it is not required to specialised equipment for sensor-to-axis alignment. This is important if deployed in working machines; since a model trained with one sensor alignment has been shown to be viable for a misaligned sensor, there is no costly constraint when replacing damaged sensors. This will result in savings in the cost of maintenance.

7.4 Limitations of research work

Some of the limitations of this research are stated below:

1. This research was performed on a fixed ball screw so the results obtained may not apply for applications where a rotating ball screw is used.
2. Since the approach employed in this research is data driven, the results obtained are largely dependent on the quality of data collected. Poor data collection will give poor results. Therefore, work is required to quantify the sensitivity of the approach to the uncertainty of the input data.
3. DCNNs are quite slow to train and would require a GPU to improve its training time. Also increasing the hidden layers of the DCNN will increase its training time. The work in this thesis did not seek to optimise network performance for speed, which would be required for a real-time system.
4. There could be a class imbalance in the in the input data. This is a situation in machine and deep learning where the total of samples in a given class of data is significantly

different from that of another class. This could occur since the particular fault scenario is unknown and the vibration data is taken from the entire length of the ball screw.

5. This research concentrated on one element of a machine and, it is assumed, one fault on that subsystem. The DCNN approach for detecting multiple faults from a single ball screw has therefore not been validated. Further work would be needed to investigate the efficacy in detecting multiple faults at a time.

7.5 Further work

This research work has answered several questions; however, further work arising from the study should include:

1. Investigation of the efficacy of the DCNN and machine learning methods for different failure modes and types. This research was focused on creating and appraising the methods for identifying a worn ball screw assembly and a good one and tested on the rig available at the University. A ball screw assembly could be worn due to many different reasons. These include overloading, misalignment, lack of or improper lubrication, and the presence of dirt. All these factors could result in different failure modes of the ball screw, which need to be measured and investigated. This will lead to a possible classification database to allow for the identification of the actual fault scenario on a particular ball screw.
2. Similarly, the methods need to be applied on different machines to understand how much benchmarking is needed and how transferable the trained model is to other ball screws. An important question is whether the method can be applied to deviations from the benchmark rather than absolute vibration characteristics. This would make mass deployment on different machines much more efficient.

3. Investigate the effectiveness of the DCNN and machine learning methods to detect wear progression from initiation to failure. Further research should be done to investigate the degree of wear in a ball screw assembly and at what point the models can detect a change has occurred. This can be done by running a new ball screw until it breaks or is severely worn. The data from this test can be used to train the proposed models developed in this research to be able to predict wear onset.
4. Application of the DCNN technique to other problem domains. This is the most obvious next step, as this research has shown that the proposed models effectively classify worn and good ball screws from vibration data converted to images. This could be in similar problem domains like tool wear, gearbox fault analysis, surface detection, and other similar problem areas in engineering. It could also be applied to diverse problem domains that are quite different from those focused on in this thesis, like medicine, computer science, archaeology, and many more.

References

- [1] Alegeh, N. E., Shagluf, A., Longstaff, A. P. *et al.*, "Accuracy in detecting failure in ballscrew assessment towards machine tool servitization," (in English), *International Journal of Mechanical Engineering and Robotics Research*, pp. 667-673, 2019.
- [2] Alegeh, N., Shagluf, A., Longstaff, A. *et al.*, "Degradation monitoring of machine tool ballscrew using deep convolution neural network," 2020, no. Conference Proceedings, pp. 209-212.
- [3] Alegeh, N., Thottoli, M., Mian, N. S. *et al.*, "Feature extraction of time-series data using dwt and fft for ball screw condition monitoring," presented at the International Conference in Manufacturing Research ICMR, Derby, UK, 2021, Paper No.
- [4] Vandermerwe, S. and Rada, J., "Servitization of business: Adding value by adding services," *European management journal*, vol. 6, no. 4, pp. 314-324, 1988, doi: 10.1016/0263-2373(88)90033-3.
- [5] Baines, T., Lightfoot, H., and Smart, P., "Servitization within manufacturing: Exploring the provision of advanced services and their impact on vertical integration," *Journal of Manufacturing Technology Management*, vol. 22, no. 7, pp. 947-954, 2011, doi: 10.1108/17410381111160988.
- [6] Mei, H., Sher, P. J., Chen, C. *et al.*, "A preliminary study on manufacturing servitization in machine tool industry," in *Proceedings of PICMET '14 Conference: Portland International Center for Management of Engineering and Technology; Infrastructure and Service Integration*, 27-31 July 2014 2014, pp. 2343-2353.
- [7] Garza-Reyes, J. A., Eldridge, S., Barber, K. D. *et al.*, "Overall equipment effectiveness (oee) and process capability (pc) measures: A relationship analysis," *The International journal of quality & reliability management*, vol. 27, no. 1, pp. 48-62, 2010, doi: 10.1108/02656711011009308.
- [8] Selcuk, S., "Predictive maintenance, its implementation and latest trends," *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of engineering manufacture*, vol. 231, no. 9, pp. 1670-1679, 2017, doi: 10.1177/0954405415601640.
- [9] March, S. T. and Scudder, G. D., "Predictive maintenance: Strategic use of it in manufacturing organizations," *Information systems frontiers*, vol. 21, no. 2, pp. 327-341, 2019, doi: 10.1007/s10796-017-9749-z.
- [10] Peng, Y., Dong, M., and Zuo, M. J., "Current status of machine prognostics in condition-based maintenance: A review," *International journal of advanced manufacturing technology*, vol. 50, no. 1, pp. 297-313, 2010, doi: 10.1007/s00170-009-2482-0.
- [11] Moeuf, A., Pellerin, R., Lamouri, S. *et al.*, "The industrial management of smes in the era of industry 4.0," *International journal of production research*, vol. 56, no. 3, pp. 1118-1136, 2018, doi: 10.1080/00207543.2017.1372647.
- [12] Adu-Amankwa, K., Attia, A. K. A., Janardhanan, M. N. *et al.*, "A predictive maintenance cost model for cnc smes in the era of industry 4.0," *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 9, pp. 3567-3587, 2019/10/01 2019, doi: 10.1007/s00170-019-04094-2.
- [13] Goda, K. and Kijima, K., "Servitization of machine-tool trading company to value orchestration platform: Visualization of japanese integral-modular strategie," *Journal of Service Science and Management*, vol. 08, pp. 349-364, 01/01 2015, doi: 10.4236/jssm.2015.83037.
- [14] Neely, A., "The servitization of manufacturing: An analysis of global trends," *14th EurOMA Conference*, 01/01 2007.

- [15] Finne, M., Brax, S., and Holmström, J., "Reversed servitization paths: A case analysis of two manufacturers," *Service business*, vol. 7, no. 4, pp. 513-537, 2013, doi: 10.1007/s11628-013-0182-1.
- [16] Brax, S. and Jonsson, K., "Developing integrated solution offerings for remote diagnostics: A comparative case study of two manufacturers," *International Journal of Operations & Production Management*, vol. 29, pp. 539-560, 04/24 2009, doi: 10.1108/01443570910953621.
- [17] Willoughby, P., Verma, M., Longstaff, A. P. *et al.*, "A holistic approach to quantifying and controlling the accuracy, performance and availability of machine tools," Springer, 2010.
- [18] Holmström, J., Brax, S., and Ala-Risku, T., "Comparing provider-customer constellations of visibility-based service," *Journal of Service Management*, vol. 21, pp. 675-692, 10/12 2010, doi: 10.1108/09564231011079093.
- [19] Laine, T., Paranko, J., and Suomala, P., "Downstream shift at a machinery manufacturer: The case of the remote technologies," *Management Research Review*, vol. 33, no. 10, pp. 980-993, 2010, doi: 10.1108/01409171011083987.
- [20] Akehurst, G., "What do we really know about services?," *Service Business*, vol. 2, no. 1, pp. 1-15, 2008.
- [21] Cohen, M. A., Agrawal, N., and Agrawal, V., "Winning in the aftermarket," vol. 84, no. 5, p. 129
- [22] Brax, S., "A manufacturer becoming service provider – challenges and a paradox," *Managing service quality*, vol. 15, no. 2, pp. 142-155, 2005, doi: 10.1108/09604520510585334.
- [23] Howells, J., "Innovation & services: New conceptual frameworks," 01/01 2000.
- [24] Baines, T., Lightfoot, H., Benedettini, O. *et al.*, "The servitization of manufacturing: A review of literature and reflection on future challenges," *Journal of Manufacturing Technology Management*, vol. 20, pp. 547-567, 06/05 2009, doi: 10.1108/17410380910960984.
- [25] Nordin, F., Kindström, D., Kowalkowski, C. *et al.*, "The risks of providing services," *Journal of service management*, vol. 22, no. 3, pp. 390-408, 2011, doi: 10.1108/09564231111136881.
- [26] Matthyssens, P. and Vandenbempt, K., "Service addition as business market strategy: Identification of transition trajectories," *Journal of service management*, vol. 21, no. 5, pp. 693-714, 2010, doi: 10.1108/09564231011079101.
- [27] Xu, A. Y., Bhatnagar, J., Bednarz, G. *et al.*, "Failure modes and effects analysis (fmea) for gamma knife radiosurgery," *Journal of applied clinical medical physics*, vol. 18, no. 6, pp. 152-168, 2017, doi: 10.1002/acm2.12205.
- [28] Vinodh, S. and Santhosh, D., "Application of fmea to an automotive leaf spring manufacturing organization," *The TQM Journal*, vol. 24, no. 3, pp. 260-274, 2012, doi: 10.1108/17542731211226772.
- [29] Yang, Z., Xu, B., Chen, F. *et al.*, "A new failure mode and effects analysis model of cnc machine tool using fuzzy theory," in *The 2010 IEEE International Conference on Information and Automation*, 20-23 June 2010 2010, pp. 582-587, doi: 10.1109/ICINFA.2010.5512403.
- [30] Lo, H.-W., Liou, J. J. H., Huang, C.-N. *et al.*, "A novel failure mode and effect analysis model for machine tool risk analysis," *Reliability Engineering & System Safety*, vol. 183, pp. 173-183, 2019/03/01/ 2019, doi: <https://doi.org/10.1016/j.res.2018.11.018>.
- [31] Patil, R. and Kothavale, B., "Failure modes and effects analysis (fmea) of computerized numerical control (cnc) turning center," *International Review of Mechanical Engineering (IREME)*, vol. 12, p. 78, 01/31 2018, doi: 10.15866/ireme.v12i1.14156.

- [32] Patil, R. B., Kothavale, B. S., and Joshi, S. G., "Reliability analysis of cnc turning center based on the assessment of trends in maintenance data: A case study," *The International journal of quality & reliability management*, vol. 34, no. 9, pp. 1616-1638, 2017, doi: 10.1108/IJQRM-08-2016-0126.
- [33] Qiao, W. and Lu, D., "A survey on wind turbine condition monitoring and fault diagnosis-part ii: Signals and signal processing methods," *IEEE transactions on industrial electronics (1982)*, vol. 62, no. 10, pp. 6546-6557, 2015, doi: 10.1109/TIE.2015.2422394.
- [34] García Márquez, F. P., Tobias, A. M., Pinar Pérez, J. M. *et al.*, "Condition monitoring of wind turbines: Techniques and methods," *Renewable energy*, vol. 46, pp. 169-178, 2012, doi: 10.1016/j.renene.2012.03.003.
- [35] Mehrjou, M. R., Mariun, N., Hamiruce Marhaban, M. *et al.*, "Rotor fault condition monitoring techniques for squirrel-cage induction machine—a review," *Mechanical systems and signal processing*, vol. 25, no. 8, pp. 2827-2848, 2011, doi: 10.1016/j.ymssp.2011.05.007.
- [36] Yang, W., Tavner, P. J., and Wilkinson, M. R., "Condition monitoring and fault diagnosis of a wind turbine synchronous generator drive train," *Renewable Power Generation, IET*, vol. 3, pp. 1-11, 04/01 2009, doi: 10.1049/iet-rpg:20080006.
- [37] Peng, Z. K., Chu, F. L., and Tse, P. W., "Singularity analysis of the vibration signals by means of wavelet modulus maximal method," *Mechanical systems and signal processing*, vol. 21, no. 2, pp. 780-794, 2007, doi: 10.1016/j.ymssp.2005.12.005.
- [38] Caselitz, P. and Giebhardt, J., "Rotor condition monitoring for improved operational safety of offshore wind energy converters," *Journal of Solar Energy Engineering-transactions of The Asme - J SOL ENERGY ENG*, vol. 127, 05/01 2005, doi: 10.1115/1.1850485.
- [39] Liang, B., Iwnicki, S. D., and Ball, A. D., "Asymmetrical stator and rotor faulty detection using vibration, phase current and transient speed analysis," *Mechanical systems and signal processing*, vol. 17, no. 4, pp. 857-869, 2003, doi: 10.1006/mssp.2002.1513.
- [40] Bouno, T., Yuji, T., Hamada, T. *et al.*, "Failure forecast diagnosis of small wind turbine using acoustic emission sensor," *KIEE Int. Trans. Electrical Machinery and Energy Conversion Systems*, vol. 5, pp. 78-83, 01/01 2005.
- [41] Qu, Y., Bechhoefer, E., He, D. *et al.*, "A new acoustic emission sensor based gear fault detection approach," *International journal of prognostics and health management*, vol. 4, no. 3, 2020, doi: 10.36001/ijphm.2013.v4i3.2141.
- [42] Widodo, A., Kim, E. Y., Son, J.-D. *et al.*, "Fault diagnosis of low speed bearing based on relevance vector machine and support vector machine," *Expert systems with applications*, vol. 36, no. 3, pp. 7252-7261, 2009, doi: 10.1016/j.eswa.2008.09.033.
- [43] Krause, T. and Ostermann, J., "Damage detection for wind turbine rotor blades using airborne sound," *Structural control and health monitoring*, vol. 27, no. 5, pp. n/a-n/a, 2020, doi: 10.1002/stc.2520.
- [44] An, Y., Gu, D., Lee, J. *et al.*, "Development and application of intensified envelope analysis for the condition monitoring system using acoustic emission signal," *Journal of mechanical science and technology*, vol. 28, no. 11, pp. 4431-4439, 2014, doi: 10.1007/s12206-014-1011-8.
- [45] Guo, P., Infield, D., and Yang, X., "Wind turbine generator condition-monitoring using temperature trend analysis," *IEEE transactions on sustainable energy*, vol. 3, no. 1, pp. 124-133, 2012, doi: 10.1109/TSTE.2011.2163430.

- [46] Touret, T., Changenet, C., Ville, F. *et al.*, "On the use of temperature for online condition monitoring of geared systems – a review," *Mechanical systems and signal processing*, vol. 101, pp. 197-210, 2018, doi: 10.1016/j.ymsp.2017.07.044.
- [47] Sheng, S., "Investigation of oil conditioning, real-time monitoring and oil sample analysis for wind turbine gearboxes (presentation)," 01/01 2011.
- [48] Zhu, J., Yoon, J. M., He, D. *et al.*, "Online particle-contaminated lubrication oil condition monitoring and remaining useful life prediction for wind turbines," *Wind energy (Chichester, England)*, vol. 18, no. 6, pp. 1131-1149, 2015, doi: 10.1002/we.1746.
- [49] Lu, D., Gong, X., and Qiao, W., "Current-based diagnosis for gear tooth breaks in wind turbine gearboxes," 2012, no. Conference Proceedings: IEEE, pp. 3780-3786, doi: 10.1109/ECCE.2012.6342293.
- [50] Gong, X., Qiao, W., and Zhou, W., "Incipient bearing fault detection via wind generator stator current and wavelet filter," 2010, no. Conference Proceedings: IEEE, pp. 2615-2620, doi: 10.1109/IECON.2010.5675135.
- [51] Mohanty, A. R., *Machinery condition monitoring: Principles and practices* (no. Book, Whole). Boca Raton, Florida: CRC Press, 2015.
- [52] Rao, J. S., *Vibratory condition monitoring of machines* (no. Book, Whole). Pangbourne: Alpha Science International Ltd, 2000.
- [53] Davies, A. and Institution of Production, E., *Management guide to condition monitoring in manufacture* (no. Book, Whole). London: Institution of Production Engineers, 1990.
- [54] Randall, R. B. and Books24x, I., *Vibration-based condition monitoring: Industrial, aerospace, and automotive applications*, 1. Aufl. ed. (no. Book, Whole). Chichester: Wiley, 2011.
- [55] Bhuiyan, M. S. H., Choudhury, I. A., Dahari, M. *et al.*, "Application of acoustic emission sensor to investigate the frequency of tool wear and plastic deformation in tool condition monitoring," *Measurement : journal of the International Measurement Confederation*, vol. 92, pp. 208-217, 2016, doi: 10.1016/j.measurement.2016.06.006.
- [56] Arun, A., Rameshkumar, K., Unnikrishnan, D. *et al.*, "Tool condition monitoring of cylindrical grinding process using acoustic emission sensor," *Materials today : proceedings*, vol. 5, no. 5, pp. 11888-11899, 2018, doi: 10.1016/j.matpr.2018.02.162.
- [57] Nasir, V., Cool, J., and Sassani, F., "Acoustic emission monitoring of sawing process: Artificial intelligence approach for optimal sensory feature selection," *International journal of advanced manufacturing technology*, vol. 102, no. 9-12, pp. 4179-4197, 2019, doi: 10.1007/s00170-019-03526-3.
- [58] Feng, G.-H. and Pan, Y.-L., "Establishing a cost-effective sensing system and signal processing method to diagnose preload levels of ball screws," *Mechanical Systems and Signal Processing*, vol. 28, pp. 78-88, 2012, doi: 10.1016/j.ymsp.2011.10.004.
- [59] Li, P., Jia, X., Feng, J. *et al.*, "Prognosability study of ball screw degradation using systematic methodology," *Mechanical systems and signal processing*, vol. 109, pp. 45-57, 2018, doi: 10.1016/j.ymsp.2018.02.046.
- [60] Giberti, H., Carnevale, M., and Boccione, M., "A power recirculating test rig for ball screws: A new perspective for endurance tests," *Machines (Basel)*, vol. 8, no. 2, p. 18, 2020, doi: 10.3390/machines8020018.
- [61] Wu, C.-H. and Kung, Y.-T., "Thermal analysis for the feed drive system of a cnc machine center," *International journal of machine tools & manufacture*, vol. 43, no. 15, pp. 1521-1528, 2003, doi: 10.1016/j.ijmachtools.2003.08.008.
- [62] Gowid, S., Dixon, R., and Ghani, S., "A novel robust automated fft-based segmentation and features selection algorithm for acoustic emission condition based monitoring

- systems," *Applied Acoustics*, vol. 88, pp. 66-74, 2015, doi: 10.1016/j.apacoust.2014.08.007.
- [63] Barnes_Industry_Inc. "Ball screw." <http://www.barnesballscrew.com> (accessed 2018).
- [64] Watterson, J. M., *Rotating equipment: Maintenance and troubleshooting* (no. Book, Whole). New York, NY: Momentum Press, Engineering, 2019.
- [65] Amanov, A. and Karimbaev, R., "Improvement in frictional and fatigue performances of aisi 4150h steel by dual ultrasonic nanocrystal surface modification for ball screw applications," *Tribology International*, vol. 161, p. 107092, 2021/09/01/ 2021, doi: <https://doi.org/10.1016/j.triboint.2021.107092>.
- [66] McEvily, A. J., *Metal failures: Mechanisms, analysis, prevention*, 2nd ed. (no. Book, Whole). Hoboken, N.J: John Wiley & Sons, Inc, 2013.
- [67] Sofronas, A., *Case histories in vibration analysis and metal fatigue for the practicing engineer* (no. Book, Whole). Hoboken, N.J: Wiley, 2012.
- [68] Smith, G. T., *Machine tool metrology: An industrial handbook*, 1st 2016. ed. (no. Book, Whole). Cham: Springer International Publishing, 2016.
- [69] Kou, G., Chao, X., Peng, Y. *et al.*, "Machine learning methods for systemic risk analysis in financial sectors," *Technological and economic development of economy*, vol. 25, no. 5, pp. 716-742, 2019, doi: 10.3846/tede.2019.8740.
- [70] Sun, S., Wei, Y., Tsui, K.-L. *et al.*, "Forecasting tourist arrivals with machine learning and internet search index," *Tourism management (1982)*, vol. 70, pp. 1-10, 2019, doi: 10.1016/j.tourman.2018.07.010.
- [71] Reig, B., Heacock, L., Geras, K. J. *et al.*, "Machine learning in breast mri," *Journal of magnetic resonance imaging*, 2019, doi: 10.1002/jmri.26852.
- [72] Keskin, H., Grunwald, S., and Harris, W. G., "Digital mapping of soil carbon fractions with machine learning," *Geoderma*, vol. 339, pp. 40-58, 2019, doi: 10.1016/j.geoderma.2018.12.037.
- [73] Herremans, D., Martens, D., and Sörensen, K., "Dance hit song prediction," *Journal of new music research*, vol. 43, no. 3, pp. 291-302, 2014, doi: 10.1080/09298215.2014.881888.
- [74] Whang, T., Lammbrau, M., and Joo, H.-m., "Detecting patterns in north korean military provocations: What machine-learning tells us," *International Relations of the Asia-Pacific*, vol. 18, no. 2, pp. 193-220, 2018, doi: 10.1093/irap/lcw016.
- [75] Wang, G., Wang, G., Cui, Y. *et al.*, "On line tool wear monitoring based on auto associative neural network," *Journal of Intelligent Manufacturing*, vol. 24, no. 6, pp. 1085-1094, 2013, doi: 10.1007/s10845-012-0636-7.
- [76] Özel, T. and Karpat, Y., "Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks," *International journal of machine tools & manufacture*, vol. 45, no. 4-5, pp. 467-479, 2005, doi: 10.1016/j.ijmachtools.2004.09.007.
- [77] Lim, G.-M., Bae, D.-M., and Kim, J.-H., "Fault diagnosis of rotating machine by thermography method on support vector machine," *Journal of Mechanical Science and Technology*, vol. 28, no. 8, pp. 2947-2952, 2014, doi: 10.1007/s12206-014-0701-6.
- [78] Muralidharan, V., Ravikumar, S., and Kangasabapathy, H., "Condition monitoring of self aligning carrying idler (sai) in belt-conveyor system using statistical features and decision tree algorithm," *Measurement*, vol. 58, pp. 274-279, 2014, doi: 10.1016/j.measurement.2014.08.047.
- [79] Abdulshahed, A. M., Longstaff, A. P., and Fletcher, S., "The application of anfis prediction models for thermal error compensation on cnc machine tools," *Applied soft computing*, vol. 27, pp. 158-168, 2015, doi: 10.1016/j.asoc.2014.11.012.

- [80] Gunerkar, R. S. and Jalan, A. K., "Classification of ball bearing faults using vibro-acoustic sensor data fusion," *Experimental techniques (Westport, Conn.)*, vol. 43, no. 5, pp. 635-643, 2019, doi: 10.1007/s40799-019-00324-0.
- [81] Abdulshahed, A., "The application of ann and anfis prediction models for thermal error compensation on cnc machine tools," Dissertation/Thesis, 2015.
- [82] Chandrasekaran, M., Muralidhar, M., Krishna, C. M. *et al.*, "Application of soft computing techniques in machining performance prediction and optimization: A literature review," *International journal of advanced manufacturing technology*, vol. 46, no. 5-8, pp. 445-464, 2009, doi: 10.1007/s00170-009-2104-x.
- [83] Dureja, J. S., Gupta, V. K., Sharma, V. S. *et al.*, "A review of empirical modeling techniques to optimize machining parameters for hard turning applications," *Proceedings of the Institution of Mechanical Engineers. Part B, Journal of engineering manufacture*, vol. 230, no. 3, pp. 389-404, 2014, doi: 10.1177/0954405414558731.
- [84] Salimiasl, A. and Özdemir, A., "Analyzing the performance of artificial neural network (ann)-, fuzzy logic (fl)-, and least square (ls)-based models for online tool condition monitoring," *International journal of advanced manufacturing technology*, vol. 87, no. 1-4, pp. 1145-1158, 2016, doi: 10.1007/s00170-016-8548-x.
- [85] Gebhardt, M., Mayr, J., Furrer, N. *et al.*, "High precision grey-box model for compensation of thermal errors on five-axis machines," *CIRP annals*, vol. 63, no. 1, pp. 509-512, 2014, doi: 10.1016/j.cirp.2014.03.029.
- [86] Gill, S. S., Singh, R., Singh, J. *et al.*, "Adaptive neuro-fuzzy inference system modeling of cryogenically treated aisi m2 hss turning tool for estimation of flank wear," *Expert systems with applications*, vol. 39, no. 4, pp. 4171-4180, 2012, doi: 10.1016/j.eswa.2011.09.117.
- [87] Hellsen, R. H. A., Angelis, G. Z., van de Molengraft, M. J. G. *et al.*, "Grey-box modeling of friction: An experimental case-study," *European Journal of Control*, vol. 6, no. 3, pp. 258-267, 2000, doi: 10.1016/S0947-3580(00)71134-4.
- [88] Shi, H., Xu, M., and Li, R., "Deep learning for household load forecasting—a novel pooling deep rnn," *IEEE transactions on smart grid*, vol. 9, no. 5, pp. 5271-5280, 2018, doi: 10.1109/tsg.2017.2686012.
- [89] McCulloch, W. S. and Pitts, W., "A logical calculus of the ideas immanent in nervous activity," *Bulletin of mathematical biology*, vol. 52, no. 1-2, pp. 99-115, 1990, doi: 10.1007/BF02459570.
- [90] Hinton, G. E., Osindero, S., and Teh, Y.-W., "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527-1554, 2006/07/01 2006, doi: 10.1162/neco.2006.18.7.1527.
- [91] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature (London)*, vol. 521, no. 7553, pp. 436-444, 2015, doi: 10.1038/nature14539.
- [92] Hinton, G., Deng, L., Yu, D. *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," vol. 29, no. 6, pp. 82-97doi: 10.1109/msp.2012.2205597.
- [93] Krizhevsky, A., Sutskever, I., and Hinton, G., "Imagenet classification with deep convolutional neural networks," vol. 60, no. 6, pp. 84-90doi: 10.1145/3065386.
- [94] Graves, A., Mohamed, A.-r., and Hinton, G., "Speech recognition with deep recurrent neural networks," 2013 2013, no. Conference Proceedings: IEEE, pp. 6645-6649, doi: 10.1109/ICASSP.2013.6638947.
- [95] Pascanu, R., Gulcehre, C., Cho, K. *et al.*, "How to construct deep recurrent neural networks," in *In Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*, 2014.

- [96] Farabet, C., Couprie, C., Najman, L. *et al.*, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915-1929, 2013, doi: 10.1109/TPAMI.2012.231.
- [97] Tompson, J., Jain, A., LeCun, Y. *et al.*, "Joint training of a convolutional network and a graphical model for human pose estimation," presented at the Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, Montreal, Canada, 2014, Paper No.
- [98] Mikolov, T., Deoras, A., Povey, D. *et al.*, "Strategies for training large scale neural network language models," 2011 2011, no. Conference Proceedings: IEEE, pp. 196-201, doi: 10.1109/ASRU.2011.6163930.
- [99] Sainath, T. N., Mohamed, A.-r., Kingsbury, B. *et al.*, "Deep convolutional neural networks for lvcsr," 2013 2013, no. Conference Proceedings: IEEE, pp. 8614-8618, doi: 10.1109/ICASSP.2013.6639347.
- [100] Gibson, E., Li, W., Sudre, C. *et al.*, "Niftynet: A deep-learning platform for medical imaging," *Computer methods and programs in biomedicine*, vol. 158, pp. 113-122, 2018, doi: 10.1016/j.cmpb.2018.01.025.
- [101] Wang, G., Li, W., Zuluaga, M. A. *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE transactions on medical imaging*, vol. 37, no. 7, pp. 1562-1573, 2018, doi: 10.1109/tmi.2018.2791721.
- [102] Cha, Y.-J., Choi, W., and Büyüköztürk, O., "Deep learning-based crack damage detection using convolutional neural networks: Deep learning-based crack damage detection using cnns," *Computer-aided civil and infrastructure engineering*, vol. 32, no. 5, pp. 361-378, 2017, doi: 10.1111/mice.12263.
- [103] Collobert, R., Weston, J., Bottou, L. *et al.*, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2493-2537, 2011.
- [104] Jean, S., Cho, K., Memisevic, R. *et al.*, "On using very large target vocabulary for neural machine translation," vol. 1, 12/05 2014, doi: 10.3115/v1/P15-1001.
- [105] Sutskever, I., Vinyals, O., and Le, Q. V., "Sequence to sequence learning with neural networks," presented at the Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Montreal, Canada, 2014, Paper No.
- [106] Jianqiang, Z., Xiaolin, G., and Xuejun, Z., "Deep convolution neural networks for twitter sentiment analysis," *IEEE access*, vol. 6, pp. 23253-23260, 2018, doi: 10.1109/access.2017.2776930.
- [107] Tang, D., Wei, F., Qin, B. *et al.*, "Coooolll: A deep learning system for twitter sentiment classification," pp. 208-212, 01/01 2014, doi: 10.3115/v1/S14-2033.
- [108] Yang, W., Tavner, P. J., Crabtree, C. J. *et al.*, "Cost-effective condition monitoring for wind turbines," *IEEE transactions on industrial electronics (1982)*, vol. 57, no. 1, pp. 263-271, 2010, doi: 10.1109/tie.2009.2032202.
- [109] Charbkaew, N., Suwanasri, T., Bunyagul, T. *et al.*, "Vibration signal analysis for condition monitoring of puffer-type high-voltage circuit breakers using wavelet transform," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 7, no. 1, pp. 13-22, 2012, doi: 10.1002/tee.21690.
- [110] Sapena-Bano, A., Pineda-Sanchez, M., Puche-Panadero, R. *et al.*, "Fault diagnosis of rotating electrical machines in transient regime using a single stator current's fft," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 11, pp. 3137-3146, 2015, doi: 10.1109/TIM.2015.2444240.
- [111] Nopiah, M. M. P. S. A. M. Z. N. S. M. B. Z. M., "Evaluating ultrasound signals of carbon steel fatigue testing using signal analysis approaches," *Journal of Central South University: English Edition*, vol. 21, no. 1, pp. 232-241, 2014, doi: 10.1007/s11771-014-1934-3.

- [112] Yadav, S. K., Tyagi, K., Shah, B. *et al.*, "Audio signature-based condition monitoring of internal combustion engine using fft and correlation approach," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 4, pp. 1217-1226, 2011, doi: 10.1109/TIM.2010.2082750.
- [113] Kalvoda, T. and Hwang, Y.-R., "A cutter tool monitoring in machining process using hilbert–huang transform," *International Journal of Machine Tools and Manufacture*, vol. 50, no. 5, pp. 495-501, 2010, doi: 10.1016/j.ijmachtools.2010.01.006.
- [114] Emerson Raja, J., "Tool condition monitoring using competitive neural network and hilbert-huang transform," *Asian Journal of Scientific Research*, vol. 6, pp. 703-714, 01/01 2013.
- [115] Ocak, H., "Automatic detection of epileptic seizures in eeg using discrete wavelet transform and approximate entropy," *Expert systems with applications*, vol. 36, no. 2, pp. 2027-2036, 2009, doi: 10.1016/j.eswa.2007.12.065.
- [116] Nussbaumer, H. J., *Fast fourier transform and convolution algorithms*, 2nd corr. and updat ed. (no. Book, Whole). New York;Berlin;: Springer-Verlag, 1982.
- [117] Koornwinder, T. H., *Wavelets: An elementary treatment of theory and applications* (no. Book, Whole). New Jersey: World Scientific, 1995.
- [118] Subasi, A., "Eeg signal classification using wavelet feature extraction and a mixture of expert model," *Expert systems with applications*, vol. 32, no. 4, pp. 1084-1093, 2007, doi: 10.1016/j.eswa.2006.02.005.
- [119] Nayak, D. R., Dash, R., and Majhi, B., "Brain mr image classification using two-dimensional discrete wavelet transform and adaboost with random forests," *Neurocomputing (Amsterdam)*, vol. 177, pp. 188-197, 2016, doi: 10.1016/j.neucom.2015.11.034.
- [120] El-Dahshan, E.-S. A., Hosny, T., and Salem, A.-B. M., "Hybrid intelligent techniques for mri brain images classification," *Digital signal processing*, vol. 20, no. 2, pp. 433-441, 2010, doi: 10.1016/j.dsp.2009.07.002.
- [121] Kehtarnavaz, N., *Digital signal processing system design: Labview-based hybrid programming*, 2nd ed. (no. Book, Whole). London;Amsterdam;: Academic Press/Elsevier, 2008.
- [122] Gao, R. X. and Yan, R., *Wavelets: Theory and applications for manufacturing*, 1 ed. (no. Book, Whole). London;New York;: Springer, 2010.
- [123] Sifuzzaman, M., Islam, M., and Ali, M., "Application of wavelet transform and its advantages compared to fourier transform," *J. Phys. Sci*, vol. 13, 01/01 2009.
- [124] Zhang, Z., Wang, Y., and Wang, K., "Fault diagnosis and prognosis using wavelet packet decomposition, fourier transform and artificial neural network," *Journal of intelligent manufacturing*, vol. 24, no. 6, pp. 1213-1227, 2013, doi: 10.1007/s10845-012-0657-2.
- [125] Duan, F., Dai, L., Chang, W. *et al.*, "Ssemg-based identification of hand motion commands using wavelet neural network combined with discrete wavelet transform," *IEEE transactions on industrial electronics (1982)*, vol. 63, no. 3, pp. 1923-1934, 2016, doi: 10.1109/tie.2015.2497212.
- [126] Lei, Y., Yang, B., Jiang, X. *et al.*, "Applications of machine learning to machine fault diagnosis: A review and roadmap," (in English), *Mechanical systems and signal processing*, vol. 138, p. 106587, 2020, doi: 10.1016/j.ymsp.2019.106587.
- [127] Han, J. and Kamber, M., *Data mining: Concepts and techniques*, 2nd;3; ed. (no. Book, Whole). London;Amsterdam;: Elsevier, 2006.
- [128] Rokach, L. and Maimon, O. Z., *Data mining with decision trees: Theory and applications* (no. Book, Whole). Singapore: World Scientific, 2008.

- [129] Bohanec, M. and Bratko, I., "Trading accuracy for simplicity in decision trees," *Machine learning*, vol. 15, no. 3, pp. 223-250, 1994, doi: 10.1007/BF00993345.
- [130] Vapnik, V. N., *The nature of statistical learning theory*. Springer-Verlag, 1995.
- [131] Cortes, C. and Vapnik, V., "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1023/a:1022627411411.
- [132] Steeb, W. H., *The nonlinear workbook: Chaos, fractals, cellular automata, neural networks, genetic algorithms, gene expression programming, support vector machine, wavelets, hidden markov models, fuzzy logic with c++, java and symbolic++ programs*, 3rd ed. (no. Book, Whole). Hackensack, NJ: World Scientific, 2005.
- [133] Cristianini, N., Shawe-Taylor, J., and Books24x, I., *An introduction to support vector machines: And other kernel-based learning methods* (no. Book, Whole). Cambridge [England];New York;: Cambridge University Press, 2000.
- [134] Yang, H., Huang, K., King, I. *et al.*, "Localized support vector regression for time series prediction," *Neurocomputing*, vol. 72, no. 10, pp. 2659-2669, 2009, doi: 10.1016/j.neucom.2008.09.014.
- [135] Rajasekaran, S., Gayathri, S., and Lee, T. L., "Support vector regression methodology for storm surge predictions," *Ocean Engineering*, vol. 35, no. 16, pp. 1578-1587, 2008, doi: 10.1016/j.oceaneng.2008.08.004.
- [136] Wu, K.-P. and Wang, S.-D., "Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space," *Pattern recognition*, vol. 42, no. 5, pp. 710-717, 2009, doi: 10.1016/j.patcog.2008.08.030.
- [137] Hastie, T., Tibshirani, R., and Friedman, J. H., *The elements of statistical learning: Data mining, inference, and prediction*, 2nd ed. (no. Book, Whole). New York: Springer, 2009.
- [138] Webb, A. R., *Statistical pattern recognition*, 2nd ed. (no. Book, Whole). New York: Wiley, 2002.
- [139] Picton, P., *Neural networks*, 2nd ed. (no. Book, Whole). Basingstoke: Palgrave, 2000.
- [140] Maren, A. J., Harston, C. T., and Pap, R. M., *Handbook of neural computing applications*. Academic Press, Inc., 1990.
- [141] van Cranenburgh, S. and Alwosheel, A., "An artificial neural network based approach to investigate travellers' decision rules," *Transportation Research Part C*, vol. 98, pp. 152-166, 2019, doi: 10.1016/j.trc.2018.11.014.
- [142] Priddy, K. L. and Keller, P. E., *Artificial neural networks: An introduction* (no. Book, Whole). Bellingham, Wash: SPIE Press, 2005.
- [143] Callan, R., *The essence of neural networks* (no. Book, Whole). London: Prentice Hall Europe, 1999.
- [144] Christ, M., Braun, N., Neuffer, J. *et al.*, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package)," *Neurocomputing (Amsterdam)*, vol. 307, pp. 72-77, 2018, doi: 10.1016/j.neucom.2018.03.067.
- [145] Liu, T. I., Kumagai, A., Wang, Y. C. *et al.*, "On-line monitoring of boring tools for control of boring operations," *Robotics and computer-integrated manufacturing*, vol. 26, no. 3, pp. 230-239, 2010, doi: 10.1016/j.rcim.2009.11.002.
- [146] Shanmugam, R. and Chattamvelli, R., *Statistics for scientists and engineers* (no. Book, Whole). Hoboken, New Jersey: John Wiley & Sons, Incorporated, 2015.
- [147] Wang, P., Meng, Q., Zhao, J. *et al.*, "Prediction of machine tool condition using support vector machine," *Journal of physics. Conference series*, vol. 305, p. 12113, 2011, doi: 10.1088/1742-6596/305/1/012113.
- [148] Bland, J. M. and Altman, D. G., "Statistics notes: Measurement error," *BMJ*, vol. 312, no. 7047, p. 1654, 1996, doi: 10.1136/bmj.312.7047.1654.

- [149] Xiang, J., Maue, E., Fujiwara, H. *et al.*, "Delineation of epileptogenic zones with high frequency magnetic source imaging based on kurtosis and skewness," *Epilepsy Research*, vol. 172, p. 106602, 2021/05/01/ 2021, doi: <https://doi.org/10.1016/j.eplepsyres.2021.106602>.
- [150] Enzweiler Marques, J. M. and Benasciutti, D., "More on variance of fatigue damage in non-gaussian random loadings – effect of skewness and kurtosis," *Procedia Structural Integrity*, vol. 25, pp. 101-111, 2020/01/01/ 2020, doi: <https://doi.org/10.1016/j.prostr.2020.04.014>.
- [151] Brandt, A. and Books24x, I., *Noise and vibration analysis: Signal analysis and experimental procedures*, 1. Aufl. ed. (no. Book, Whole). Oxford: Wiley-Blackwell, 2011.
- [152] Liu, T.-I. and Jolley, B., "Tool condition monitoring (tcm) using neural networks," *The International Journal of Advanced Manufacturing Technology*, vol. 78, no. 9, pp. 1999-2007, 2015, doi: 10.1007/s00170-014-6738-y.
- [153] Momoh, J. A. and Button, R., "Design and analysis of aerospace dc arcing faults using fast fourier transformation and artificial neural network," 2003, vol. 2, no. Conference Proceedings: IEEE, pp. 788-793 Vol. 2, doi: 10.1109/PES.2003.1270407.
- [154] Brezak, D., Udiljak, T., Majetic, D. *et al.*, "Tool wear monitoring using radial basis function neural network," 2004, vol. 3, no. Conference Proceedings: IEEE, pp. 1859-1862 vol.3, doi: 10.1109/IJCNN.2004.1380892.
- [155] Hong, G. S., Rahman, M., and Zhou, Q., "Using neural network for tool condition monitoring based on wavelet decomposition," *International journal of machine tools & manufacture*, vol. 36, no. 5, pp. 551-566, 1996, doi: 10.1016/0890-6955(95)00067-4.
- [156] Lee, B. Y., "Application of the discrete wavelet transform to the monitoring of tool failure in end milling using the spindle motor current," *International journal of advanced manufacturing technology*, vol. 15, no. 4, pp. 238-243, 1999, doi: 10.1007/s001700050062.
- [157] Anoh, O. O., Abd-Alhameed, R. A., Jones, S. M. *et al.*, "Comparison of orthogonal and biorthogonal wavelets for multicarrier systems," in *2013 8th IEEE Design and Test Symposium*, 16-18 Dec. 2013 2013, pp. 1-4, doi: 10.1109/IDT.2013.6727137.
- [158] Chapa, J. O. and Rao, R. M., "Algorithms for designing wavelets to match a specified signal," *IEEE transactions on signal processing*, vol. 48, no. 12, pp. 3395-3406, 2000, doi: 10.1109/78.887001.
- [159] Daubechies, I., "Ten lectures on wavelets," Philadelphia, Pa, 1992, vol. 61, no. Conference Proceedings: Society for Industrial and Applied Mathematics.
- [160] Di, W., Bhardwaj, A., and Wei, J., *Deep learning essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling* (no. Book, Whole). Birmingham: Packt, 2018.
- [161] Srivastava, N., Hinton, G., Krizhevsky, A. *et al.*, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [162] Pouyanfar, S., Sadiq, S., Yan, Y. *et al.*, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1-36, 2019, doi: 10.1145/3234150.
- [163] Ioannidou, A., Chatzilari, E., Nikolopoulos, S. *et al.*, "Deep learning advances in computer vision with 3d data: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1-38, 2017, doi: 10.1145/3042064.

- [164] Lecun, Y., Bottou, L., Bengio, Y. *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.
- [165] Bengio, Y., "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade: Second edition*, Montavon, G., Orr, G. B., and Müller, K.-R. Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 437-478.
- [166] Bottou, L., "Stochastic gradient descent tricks," in *Neural networks: Tricks of the trade: Second edition*, Montavon, G., Orr, G. B., and Müller, K.-R. Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 421-436.
- [167] Xu, Y., Jia, Z., Wang, L.-B. *et al.*, "Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features," *BMC bioinformatics*, vol. 18, no. 1, pp. 281-17, 2017, doi: 10.1186/s12859-017-1685-x.
- [168] Marmanis, D., Datcu, M., Esch, T. *et al.*, "Deep learning earth observation classification using imagenet pretrained networks," *IEEE geoscience and remote sensing letters*, vol. 13, no. 1, pp. 105-109, 2016, doi: 10.1109/lgrs.2015.2499239.
- [169] Koh, B. H. D., Lim, C. L. P., Rahimi, H. *et al.*, "Deep temporal convolution network for time series classification," (in English U6 - institution=44HUD_INST&vid=44HUD_INST%3AServices&%3Fctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-8&rft_id=info%3Asid%2Fsummon.serialssolutions.com&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.genre=article&rft.atitle=Deep+Temporal+Convolution+Network+for+Time+Series+Classification&rft.jtitle=Sensors+%28Basel%2C+Switzerland%29&rft.date=2021-01-16&rft.eissn=1424-8220&rft.volume=21&rft.issue=2&rft_id=info:doi/10.3390%2Fs21020603&rft.externalDBID=NO_FULL_TEXT¶mdict=en-US U7 - Journal Article), *Sensors (Basel, Switzerland)*, vol. 21, no. 2, p. 603, 2021, doi: 10.3390/s21020603.
- [170] Moore, J., Stammers, J., and Dominguez-Caballero, J., "The application of machine learning to sensor signals for machine tool and process health assessment," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 235, no. 10, pp. 1543-1557, 2021, doi: 10.1177/0954405420960892.
- [171] Maxim_Integrated. "Programmable resolution 1-wire digital thermometer ds18b20 datasheet." <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> (accessed 2020).
- [172] "Bs iso 230-2:2014: Test code for machine tools. Determination of accuracy and repeatability of positioning of numerically controlled axes," ed: British Standards Institute, 2014.
- [173] Arenas, M. L. and San Antolín, A., "On symmetric compactly supported wavelets with vanishing moments associated to $\{e_d^{(2)}\}_z$ dilations," (in English), *The Journal of fourier analysis and applications*, vol. 26, no. 5, 2020, doi: 10.1007/s00041-020-09782-2.
- [174] Peng, Z. and Wang, G., "Study on optimal selection of wavelet vanishing moments for ecg denoising," (in English), *Scientific reports*, vol. 7, no. 1, pp. 4564-11, 2017, doi: 10.1038/s41598-017-04837-9.
- [175] MathWorks. "Classification learner toolbox: User's guide (r2018a)." <https://uk.mathworks.com/help/stats/> (accessed 2018).
- [176] Abdulshahed, A., Longstaff, A. P., Fletcher, S. *et al.*, "Application of gnmci(1, n) to environmental thermal error modelling of cnc machine tools," KTH Royal Institute of Technology, 2013.

- [177] Wu, C.-W., Tang, C.-H., Chang, C.-F. *et al.*, "Thermal error compensation method for machine center," *International journal of advanced manufacturing technology*, vol. 59, no. 5-8, pp. 681-689, 2011, doi: 10.1007/s00170-011-3533-x.
- [178] MathWorks. "Deep learning toolbox: User's guide (r2019a)." <https://uk.mathworks.com/help/stats/> (accessed 2019).

Appendices

Appendix A: MATLAB code

The MATLAB codes that were used in this thesis have been adapted from the MATLAB documentation version R2020a [178].

A1: DCNN code _ improved version

```

1   % gpuDevice(1)
2   clear
3   %Get training images
4   ballscrewds = imageDatastore('pictures10degrees', 'IncludeSubfolders', true, ...
5       'LabelSource', 'foldernames', 'ReadFcn', @ballscrewimageread);
6   %Split into training and testing sets
7   [trainImgs, testImgs] = splitEachLabel(ballscrewds, 0.8, 'randomize');
8   %Determine the number of categories
9   numClasses = numel(categories(ballscrewds.Labels));
10
11  %Get the layers from AlexNet
12  net = alexnet;
13  layers = net.Layers;
14  %Modify the classification and output layers
15  layers(end-2) = fullyConnectedLayer(numClasses);
16  layers(end) = classificationLayer;
17
18  %define training algorithm options
19  options = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'Shuffle', 'every-epoch', ...
20      'ValidationData', testImgs, 'ValidationFrequency', 5, ...
21      'Verbose', false, ...
22      'Plots', 'training-progress');
23
24  %Perform training
25  [ballscrewnet, info] = trainNetwork(trainImgs, layers, options);
26  %Use the trained network to classify test images
27  [testpreds, prob] = classify(ballscrewnet, testImgs);
28
29  %Calculate the accuracy
30  nnz(testpreds == testImgs.Labels) / numel(testpreds)
31  %Visualize the confusion matrix
32  [ballscrewconf, ballscrewnames] = confusionmat(testImgs.Labels, testpreds);
33  heatmap(ballscrewnames, ballscrewnames, ballscrewconf);

```

A2: DCNN code _ original version

```

1   % gpuDevice(1)
2 -  clear
3   %Get training images
4 -  ballscrewds = imageDatastore('pictures10degrees','IncludeSubfolders',true,...
5     'LabelSource','foldernames','ReadFcn',@ballscrewimageread);
6   %Split into training and testing sets
7 -  [trainImgs,testImgs] = splitEachLabel(ballscrewds,0.8);
8   %Determine the number of categories
9 -  numClasses = numel(categories(ballscrewds.Labels));
10
11  %Get the layers from AlexNet
12 -  net = alexnet;
13 -  layers = net.Layers;
14  %Modify the classification and output layers
15 -  layers(end-2) = fullyConnectedLayer(numClasses);
16 -  layers(end) = classificationLayer;
17
18  %define training algorithm options
19 -  options = trainingOptions('sgdm','InitialLearnRate', 0.001);
20
21  %Perform training
22 -  [ballscrewnet,info] = trainNetwork(trainImgs, layers, options);
23  %Use the trained network to classify test images
24 -  testpreds = classify(ballscrewnet,testImgs);
25  |
26  %Calculate the accuracy
27 -  nnz(testpreds == testImgs.Labels)/numel(testpreds)
28  %Visualize the confusion matrix
29 -  [ballscrewconf,ballscrewnames] = confusionmat(testImgs.Labels,testpreds);
30 -  heatmap(ballscrewnames,ballscrewnames,ballscrewconf);

```

A3: Helper function for DCNN

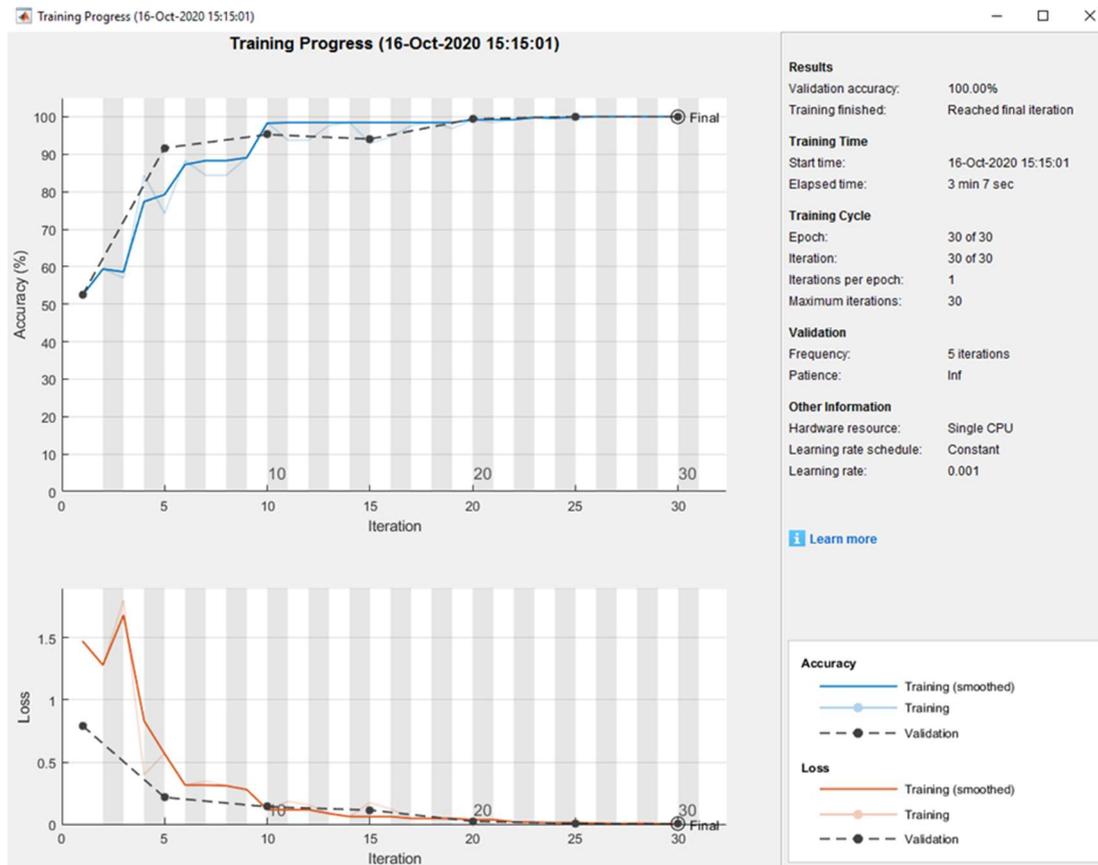
```

1   function img = ballscrewimageread(file)
2     % Read in image
3 -   img = imread(file);
4     % Resize image
5 -   img = imresize(img,[227 227]);
6     % Convert grayscale to color (RGB)
7 -   img = repmat(img,[1 1 3]);
8 -   end

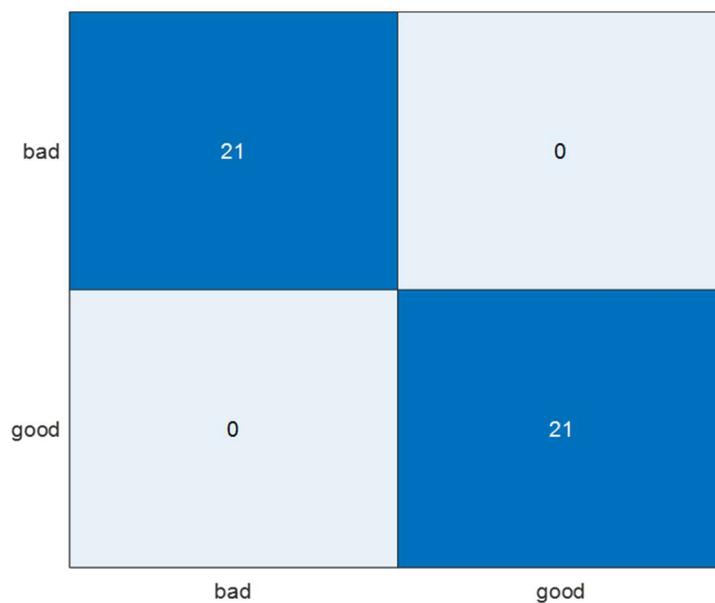
```

Appendix B: DCNN results for misaligned sensor axis test

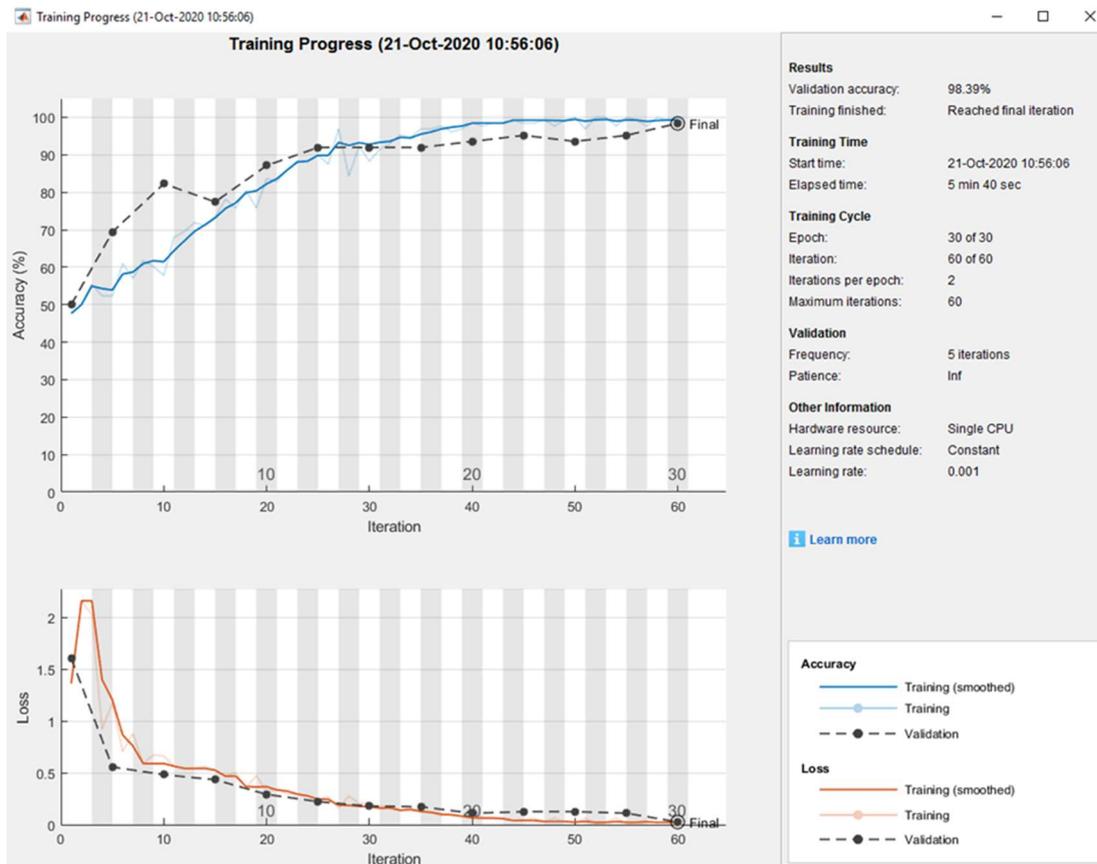
B1: Training progress for 9:1 ratio on training and test set



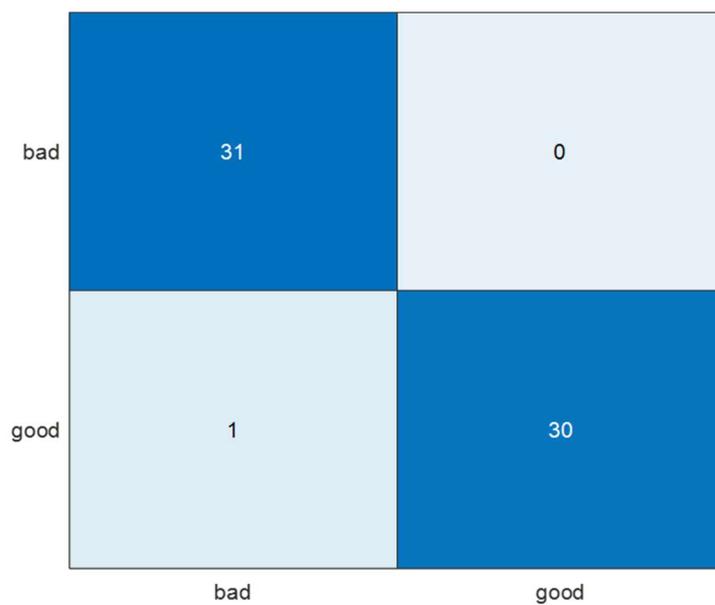
B2: Confusion matrix for 9:1 ratio on training and test set



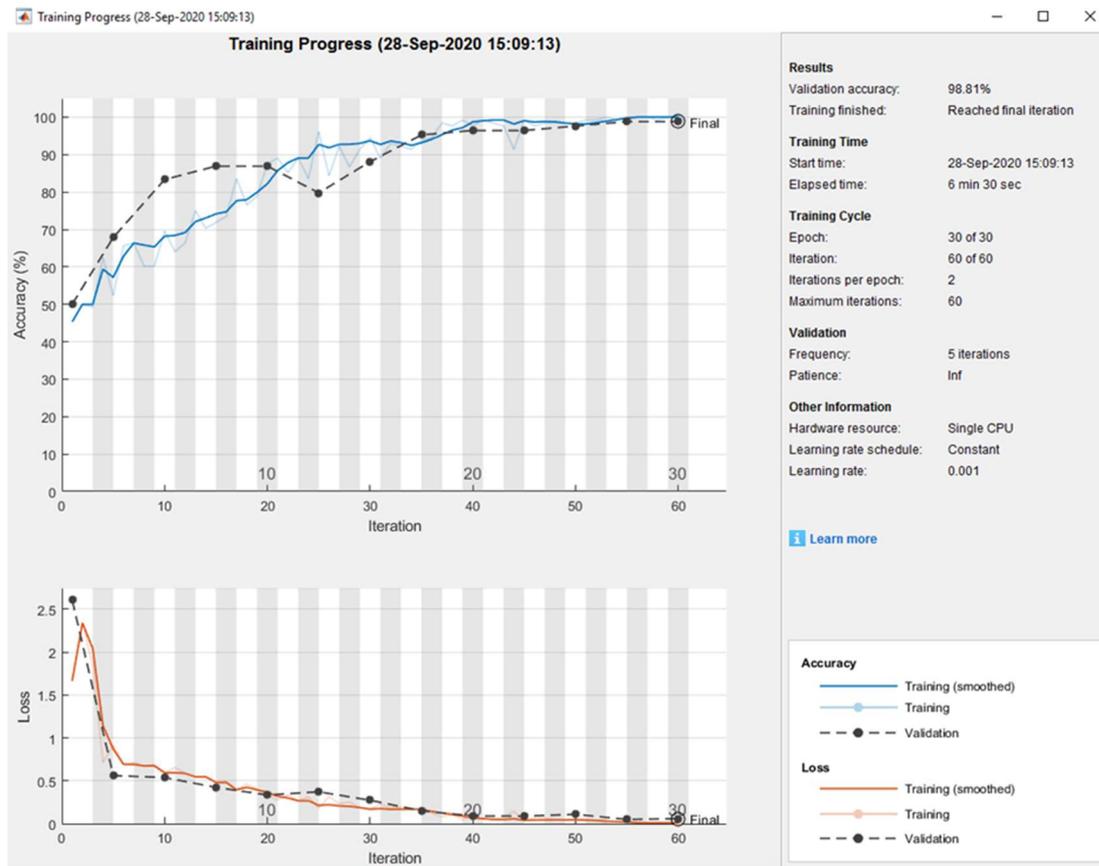
B3: Training progress for 8.5:1.5 ratio on training and test set



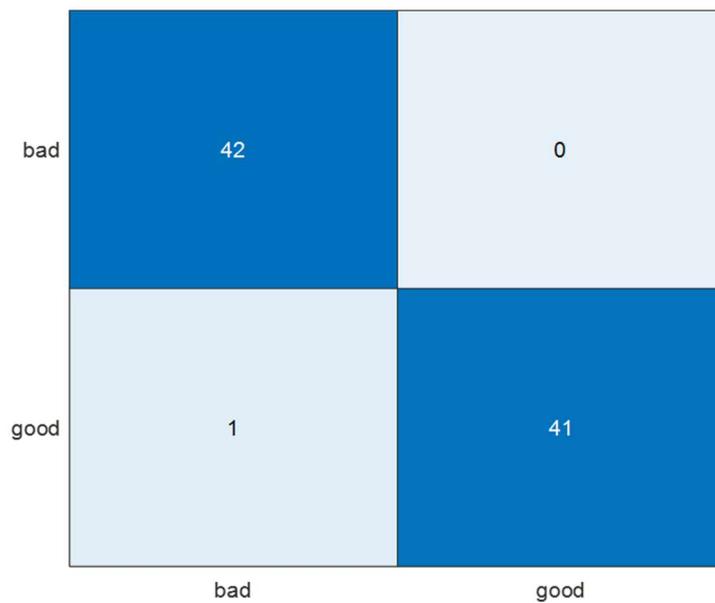
B4: Confusion matrix for 8.5:1.5 ratio on training and test set



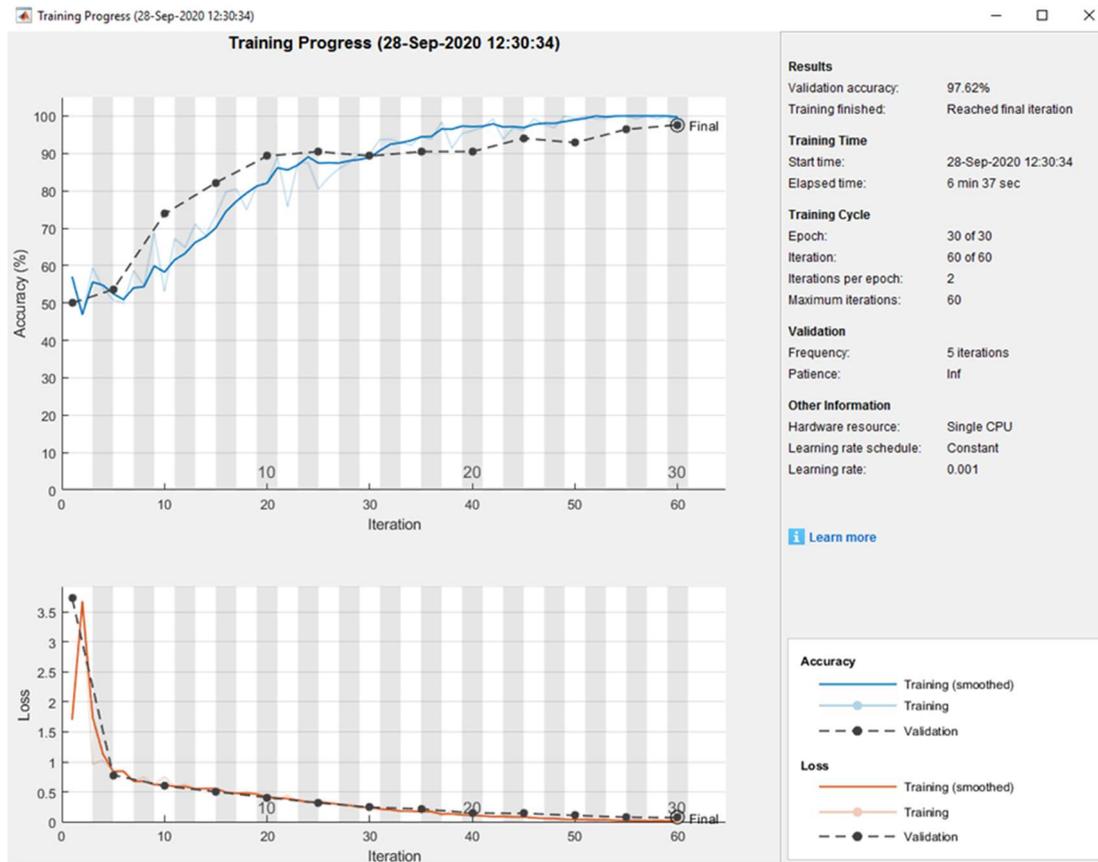
B5: Training progress for 4:1 ratio on training and test set



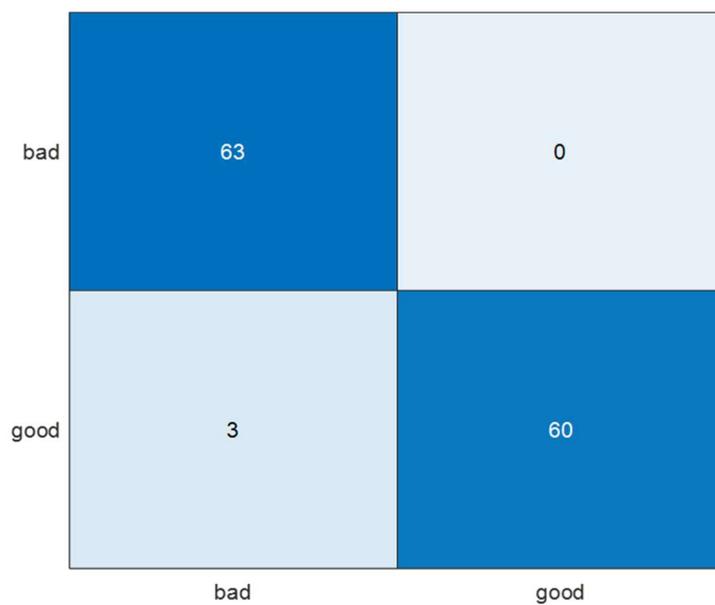
B6: Confusion matrix for 4:1 ratio on training and test set



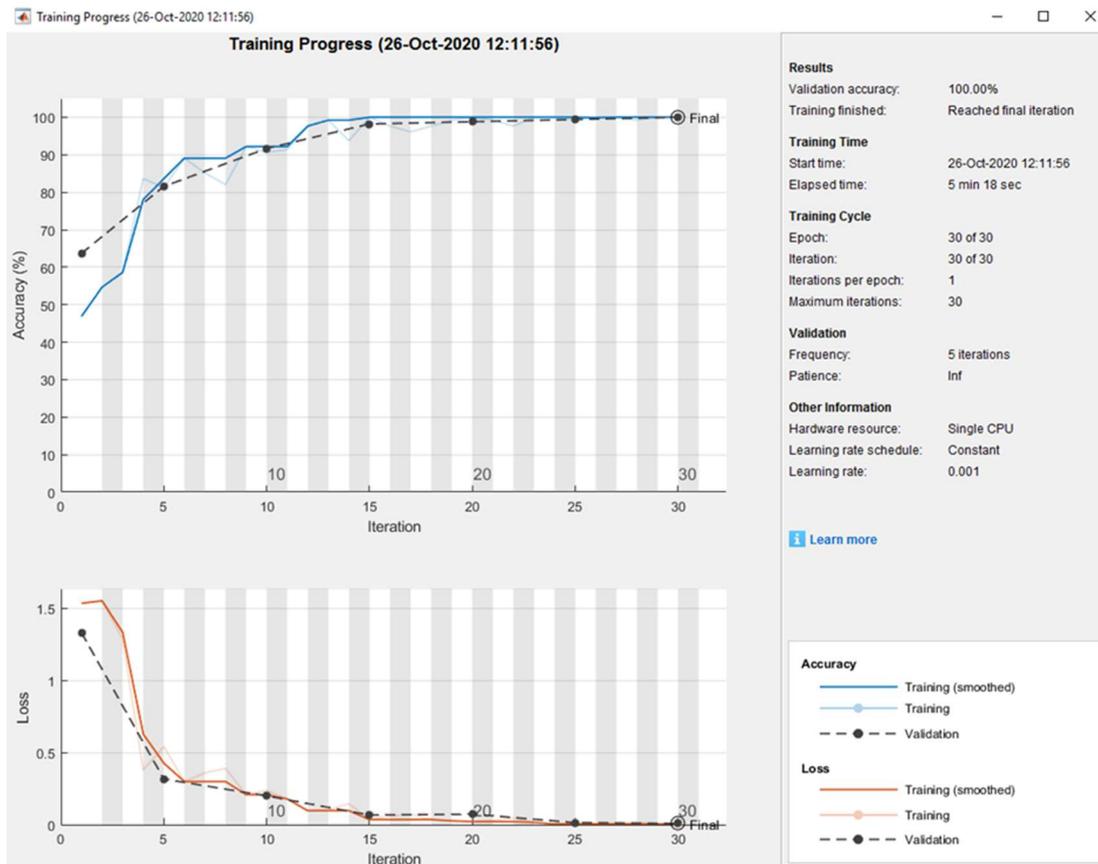
B7: Training progress for 7:3 ratio on training and test set



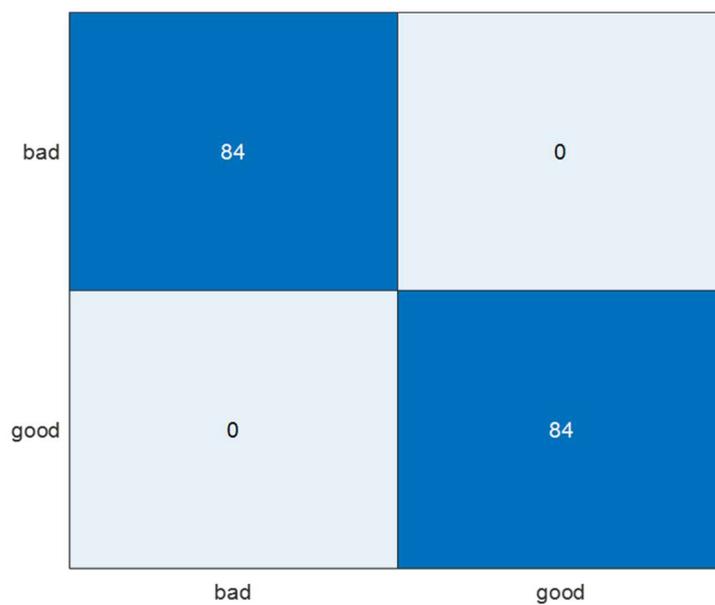
B8: Confusion matrix for 7:3 ratio on training and test set



B9: Training progress for 3:2 ratio on training and test set

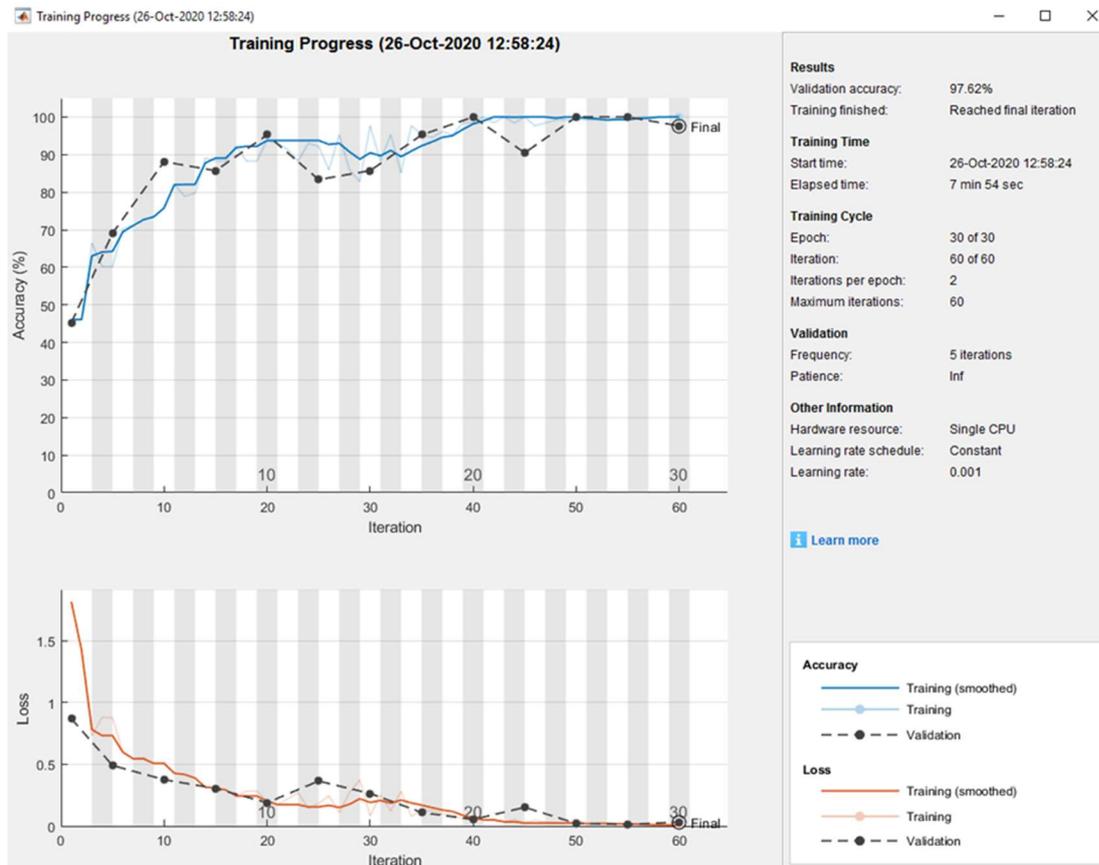


B10: Confusion matrix for 3:2 ratio on training and test set

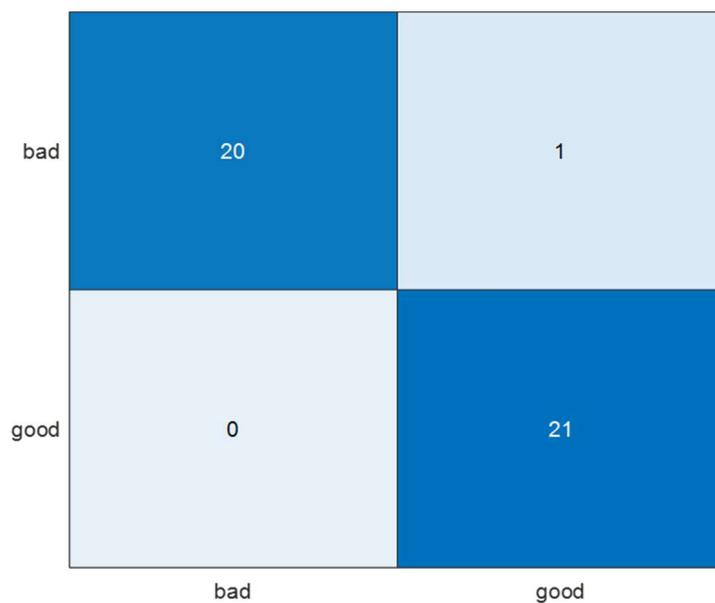


Appendix C: DCNN results for 5 °C thermal perturbation test

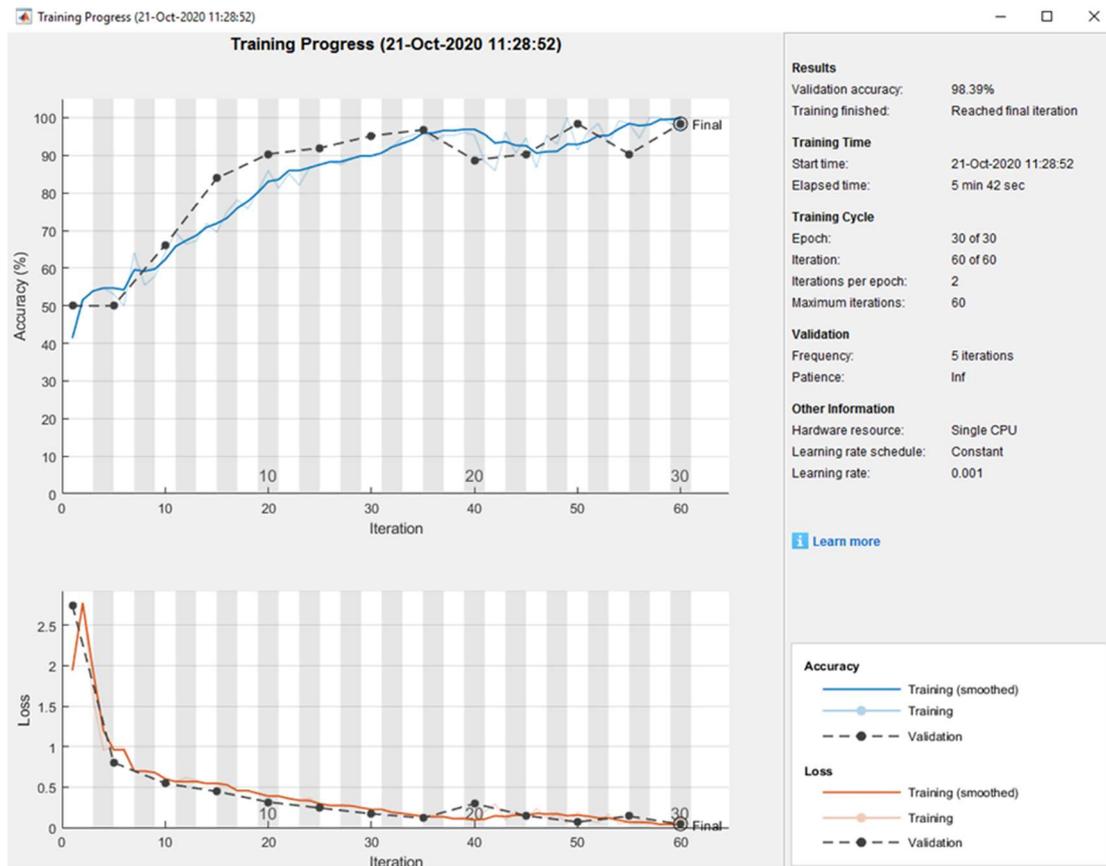
C1: Training progress for 9:1 ratio on training and test set



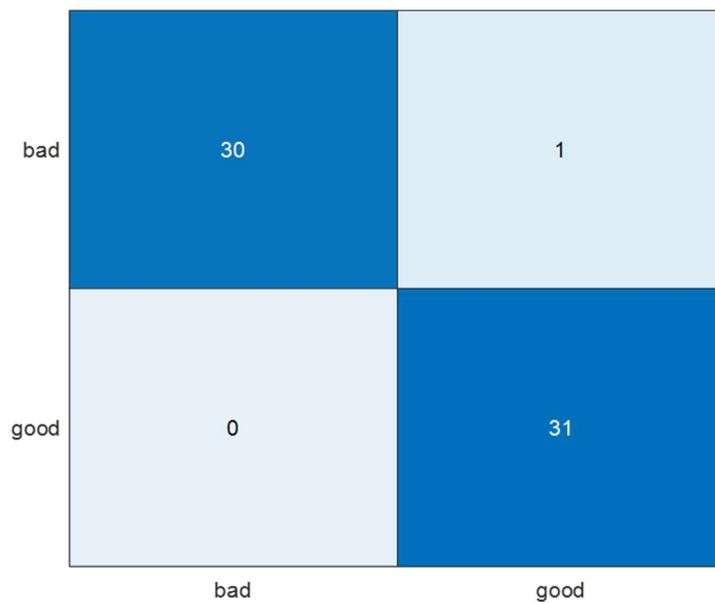
C2: Confusion matrix for 9:1 ratio on training and test set



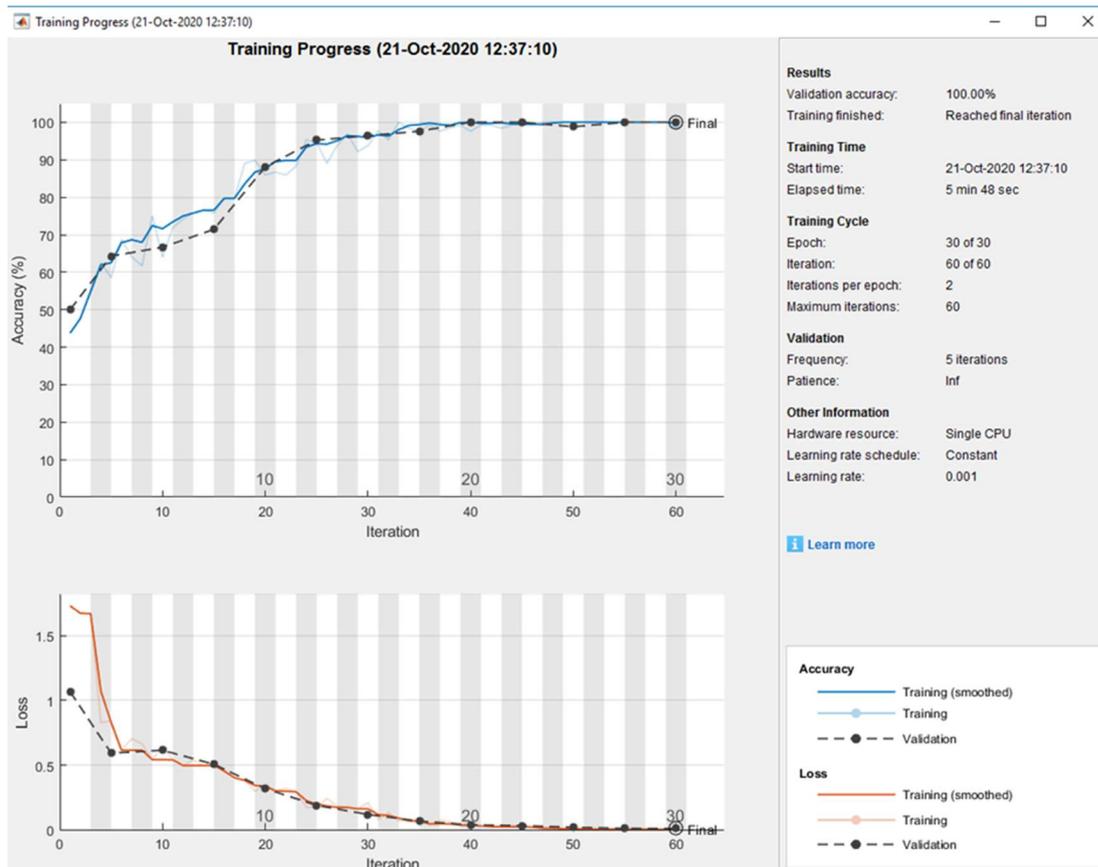
C3: Training progress for 8.5:1.5 ratio on training and test set



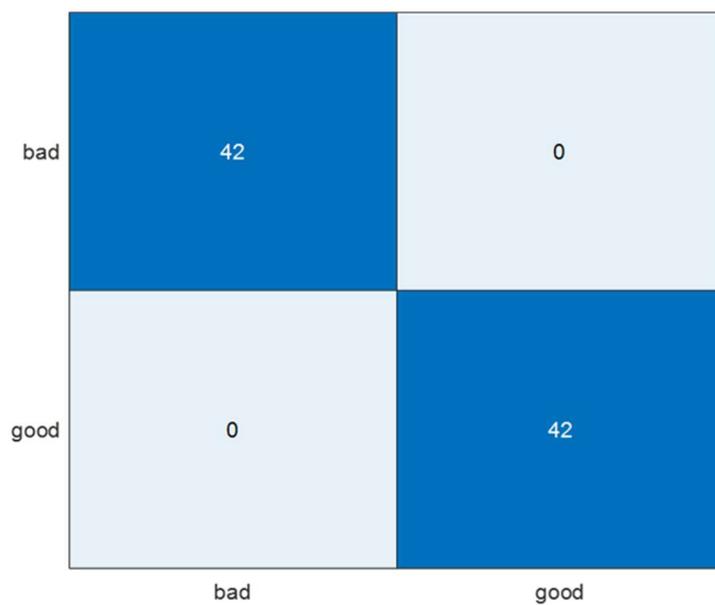
C4: Confusion matrix for 8.5:1.5 ratio on training and test set



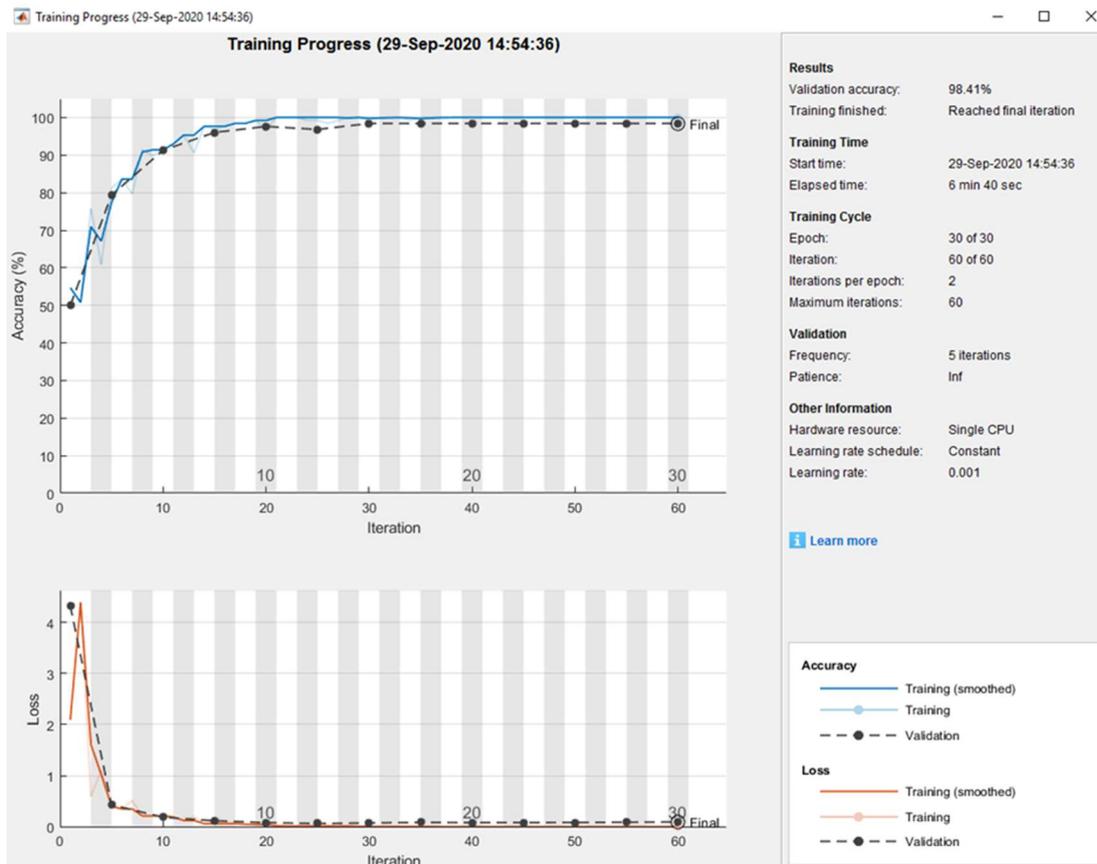
C5: Training progress for 4:1 ratio on training and test set



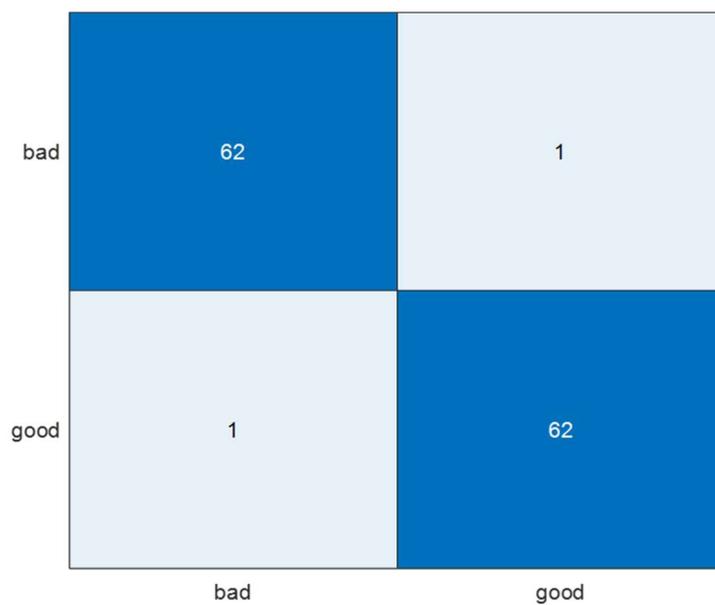
C6: Confusion matrix for 4:1 ratio on training and test set



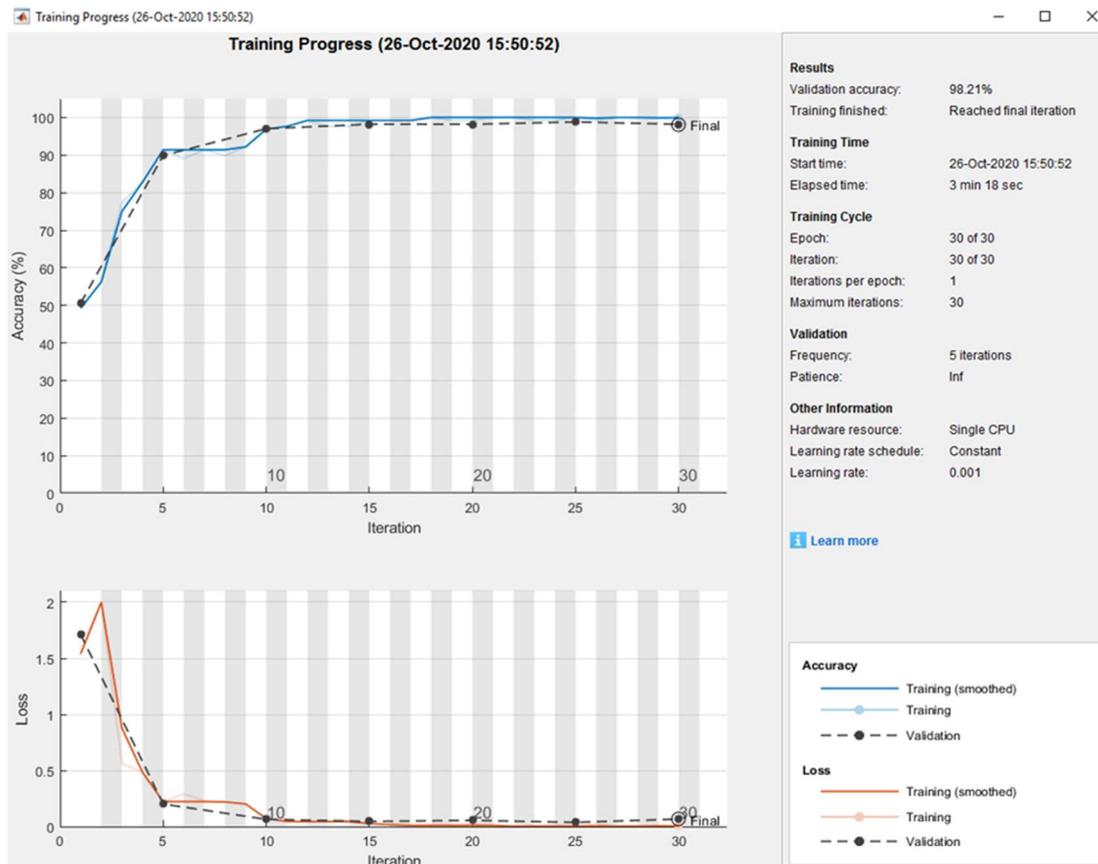
C7: Training progress for 7:1 ratio on training and test set



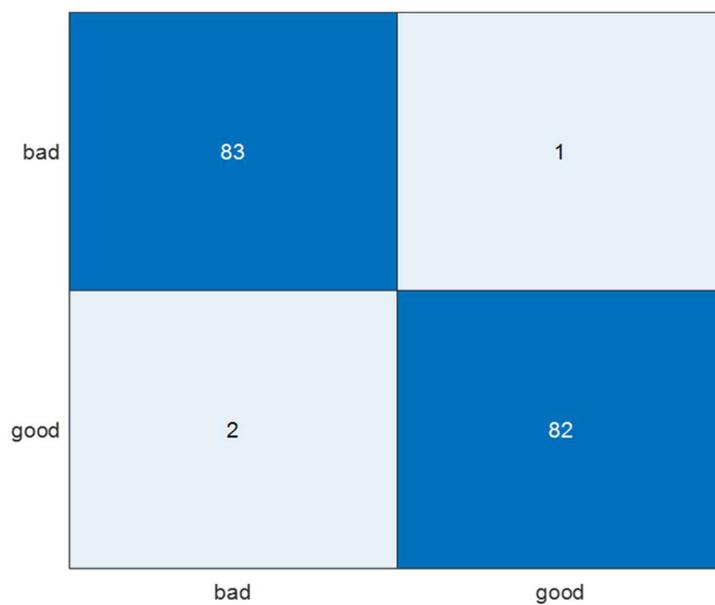
C8: Confusion matrix for 7:1 ratio on training and test set



C9: Training progress for 3:2 ratio on training and test set

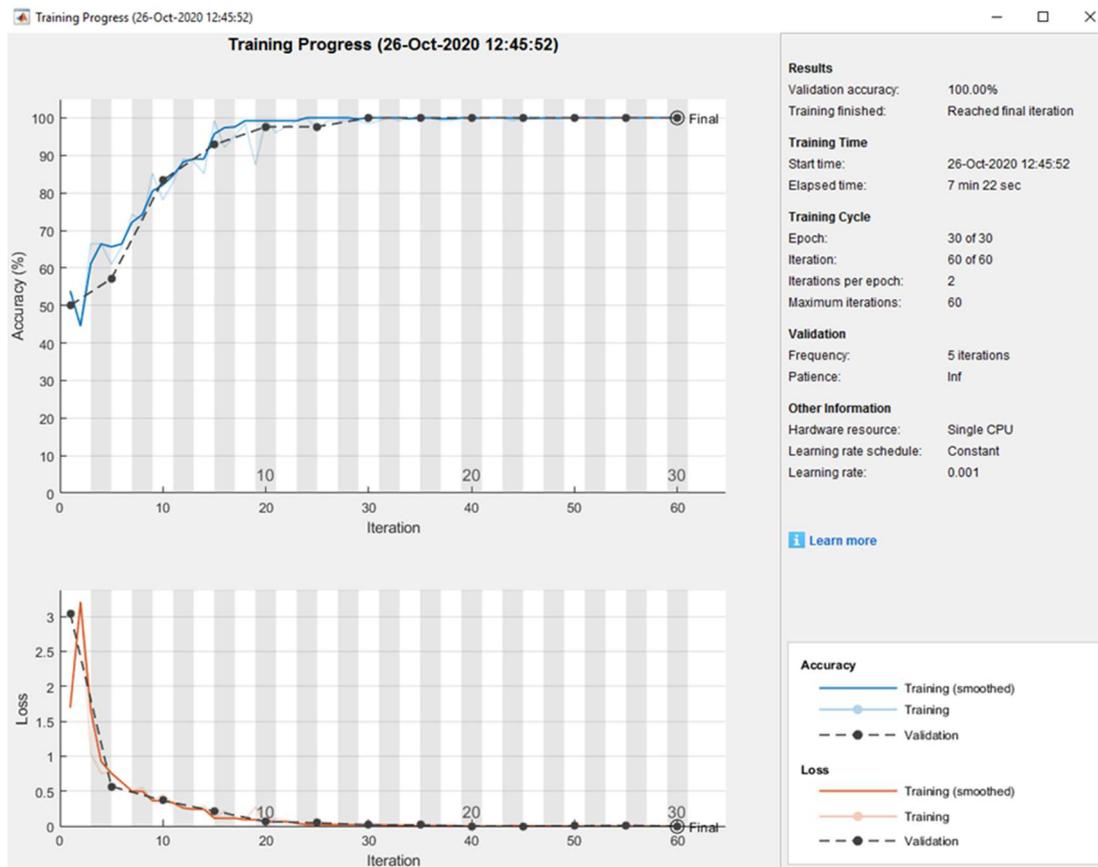


C10: Confusion matrix for 3:2 ratio on training and test set

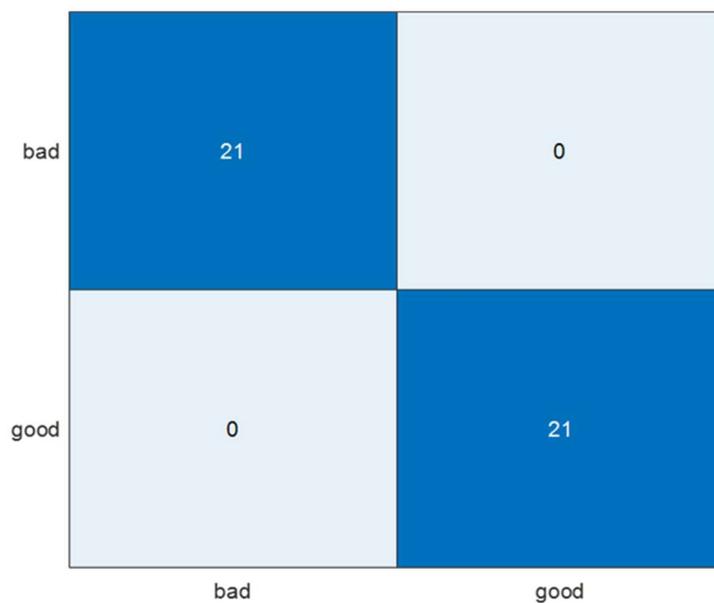


Appendix D: DCNN results for 10 °C thermal perturbation test

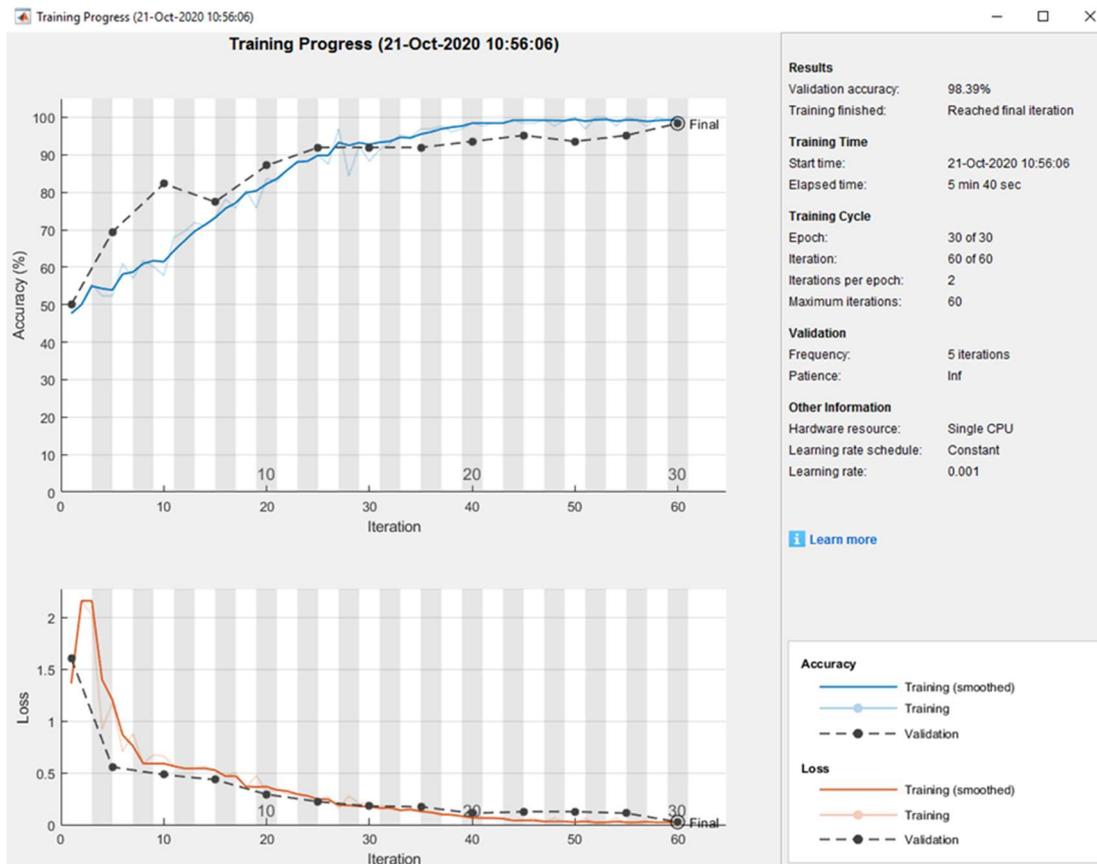
D1: Training progress for 9:1 ratio on training and test set



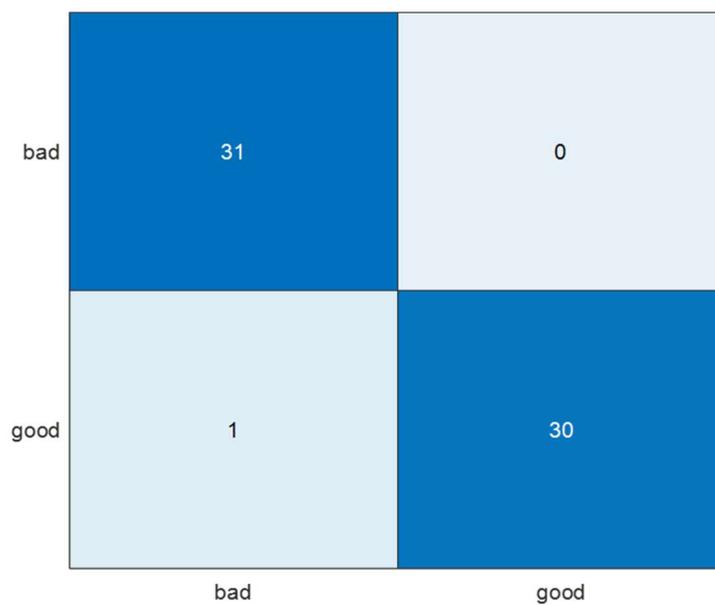
D2: Confusion matrix for 9:1 ratio on training and test set



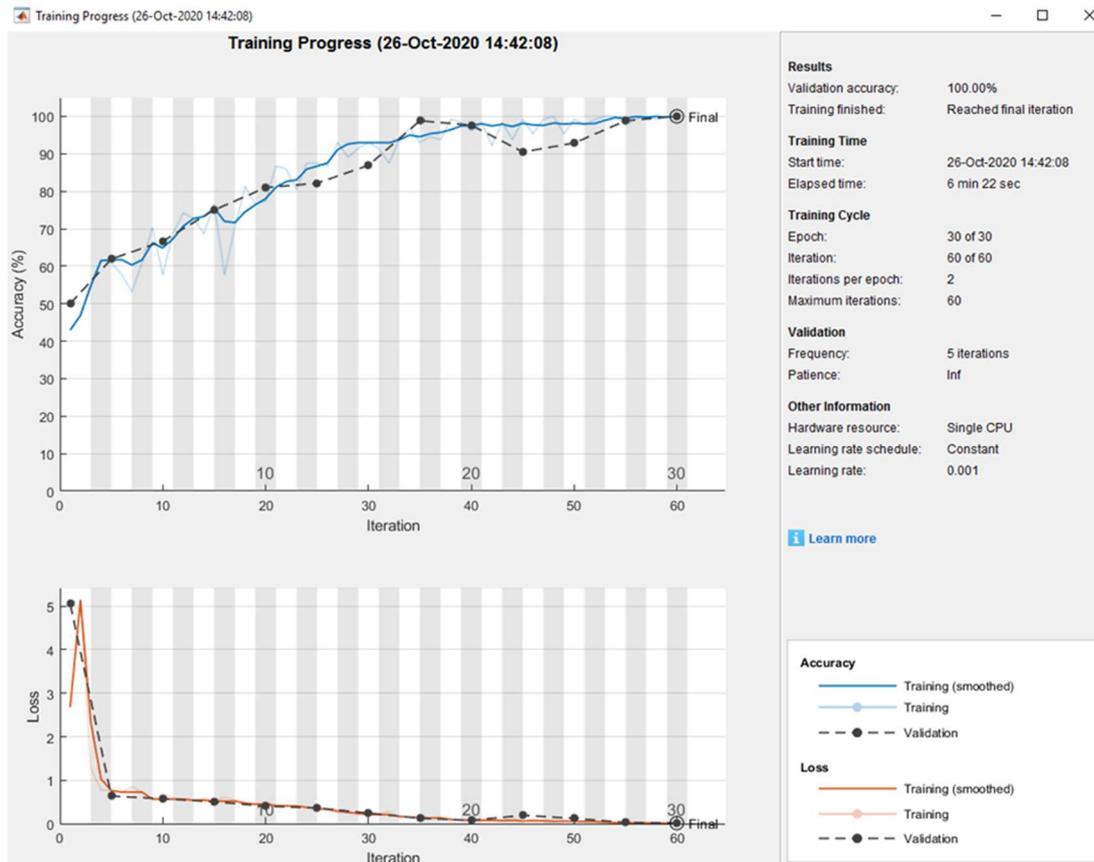
D3: Training progress for 8.5:1.5 ratio on training and test set



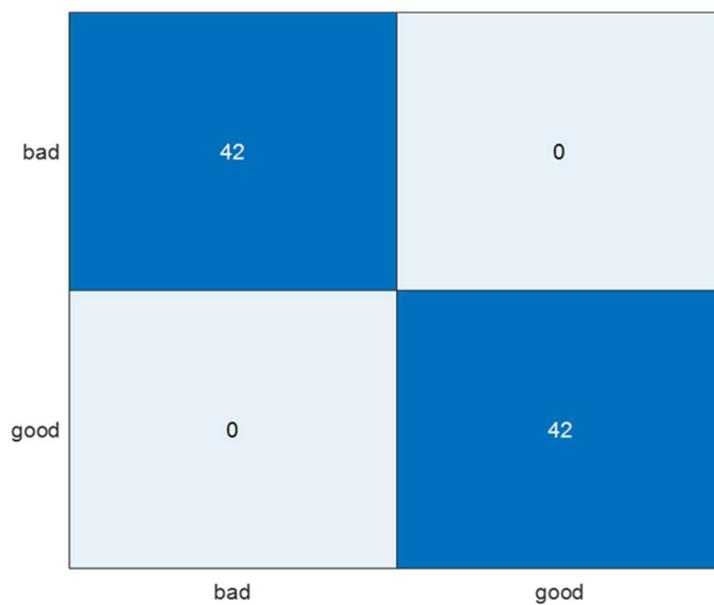
D4: Confusion matrix for 8.5:1.5 ratio on training and test set



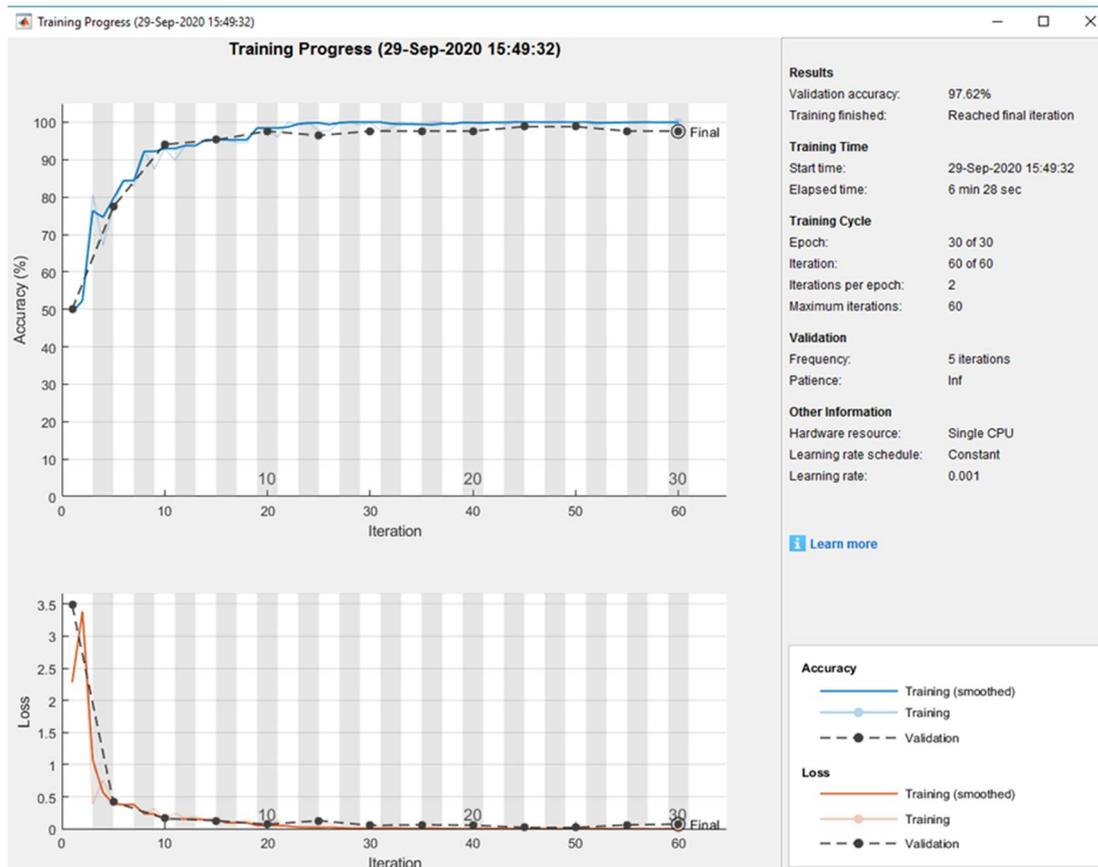
D5: Training progress for 4:1 ratio on training and test set



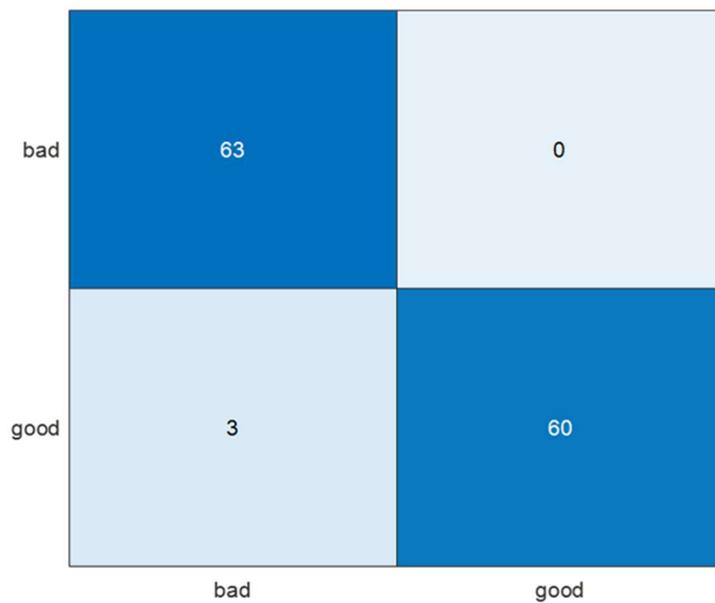
D6: Confusion matrix for 4:1 ratio on training and test set



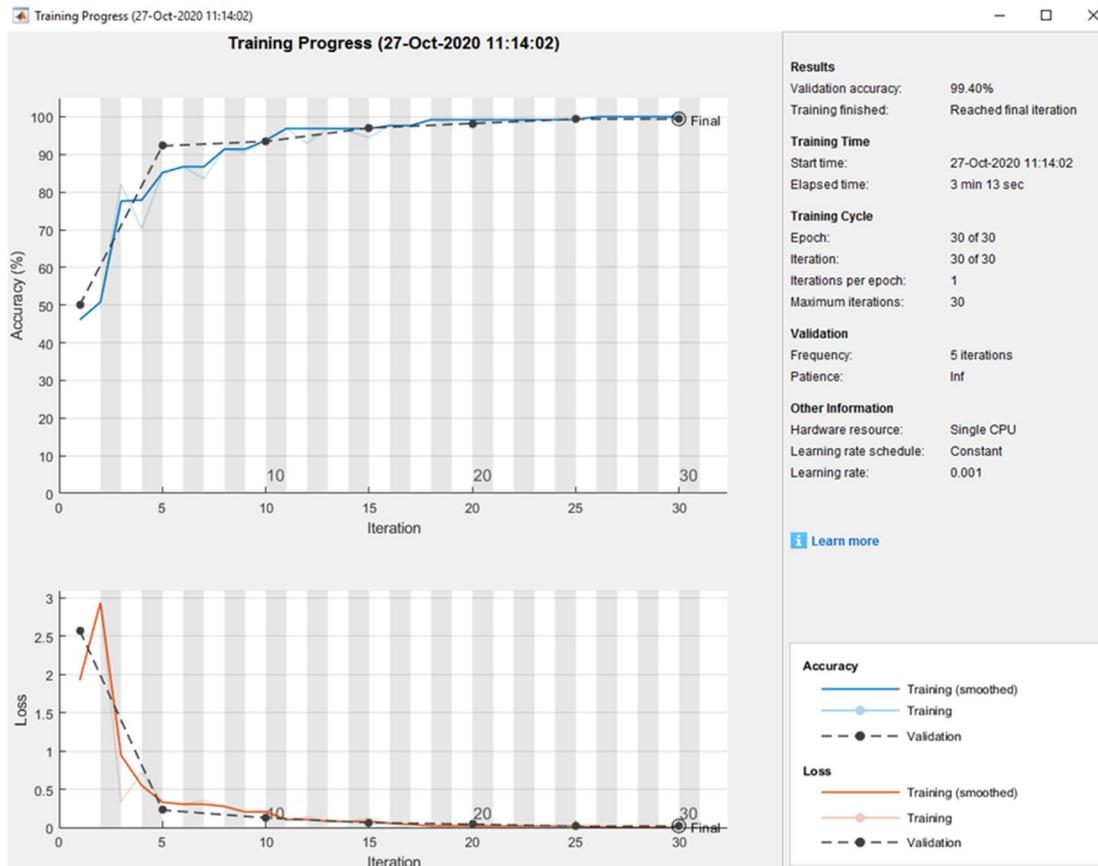
D7: Training progress for 7:3 ratio on training and test set



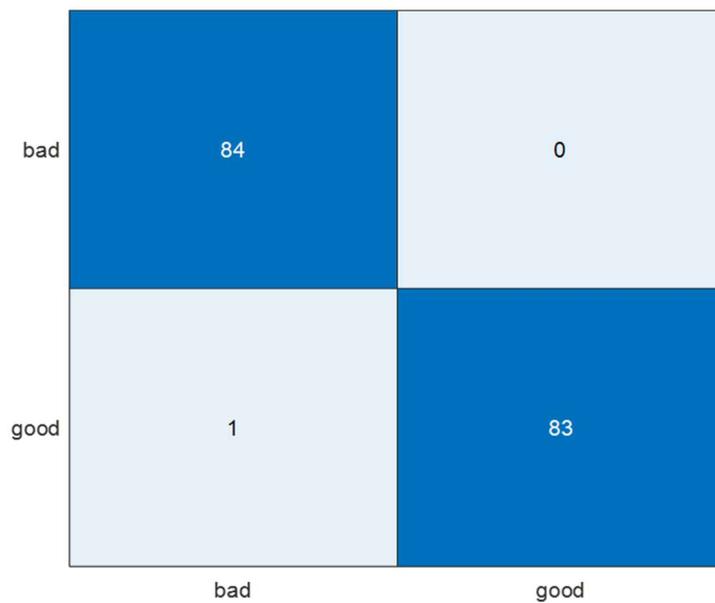
D8: Confusion matrix for 7:3 ratio on training and test set



D9: Training progress for 3:2 ratio on training and test set

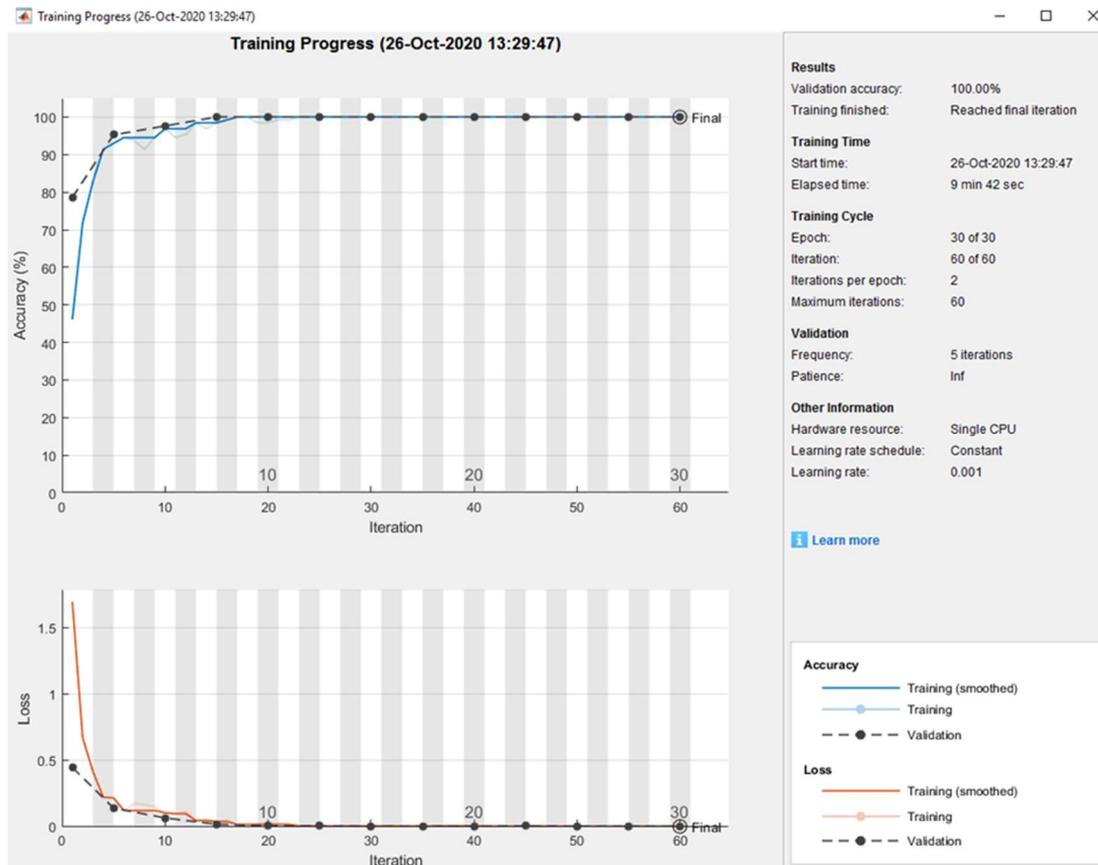


D10: Confusion matrix for 3:2 ratio on training and test set

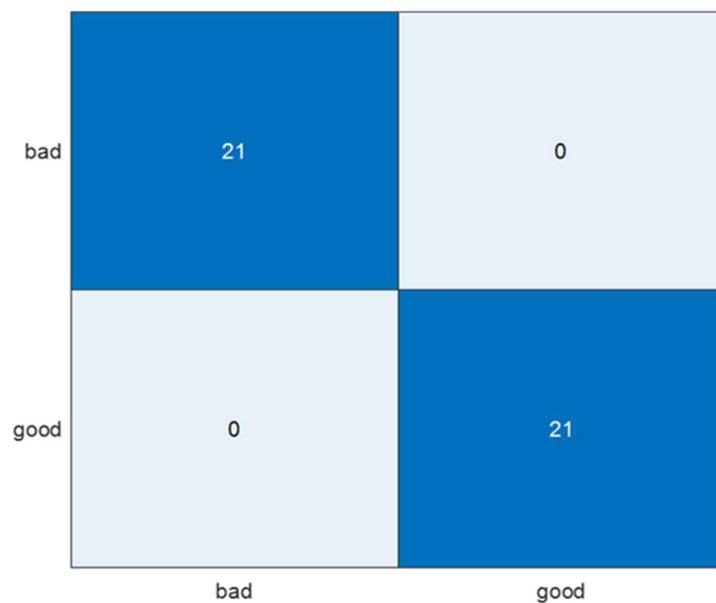


Appendix E: DCNN results for 15 °C thermal perturbation test

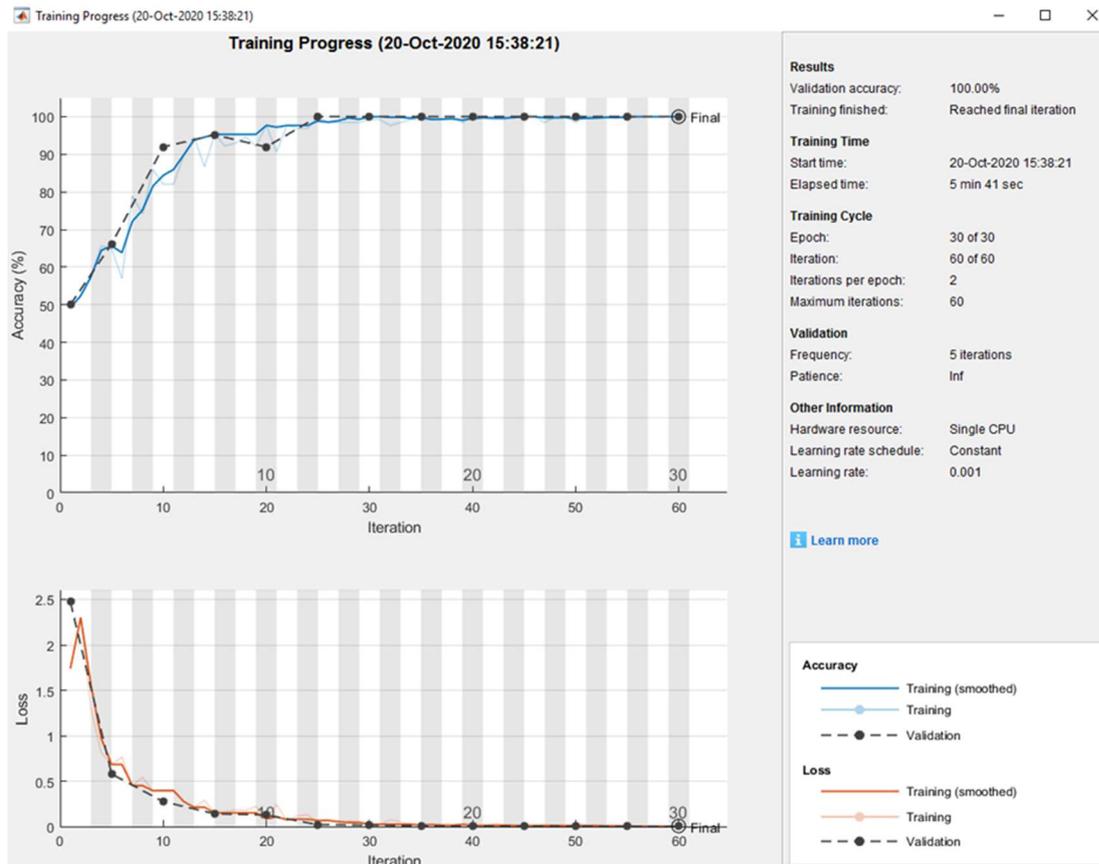
E1: Training progress for 9:1 ratio on training and test set



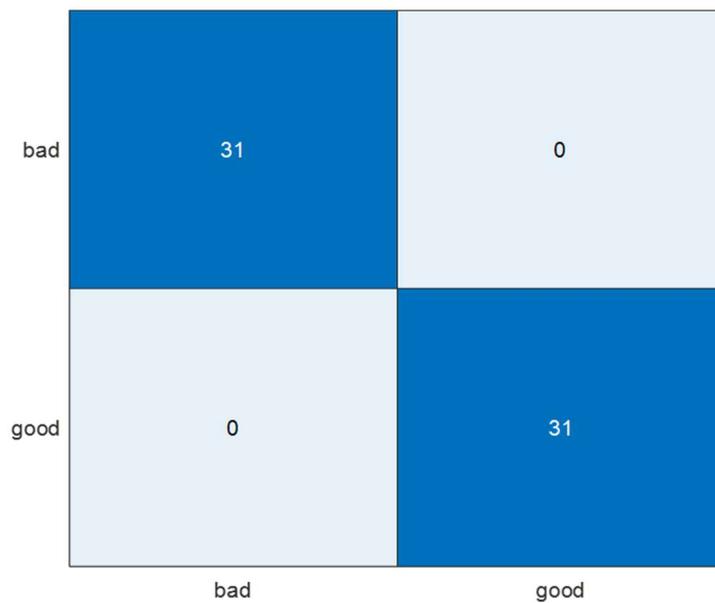
E2: Confusion matrix for 9:1 ratio on training and test set



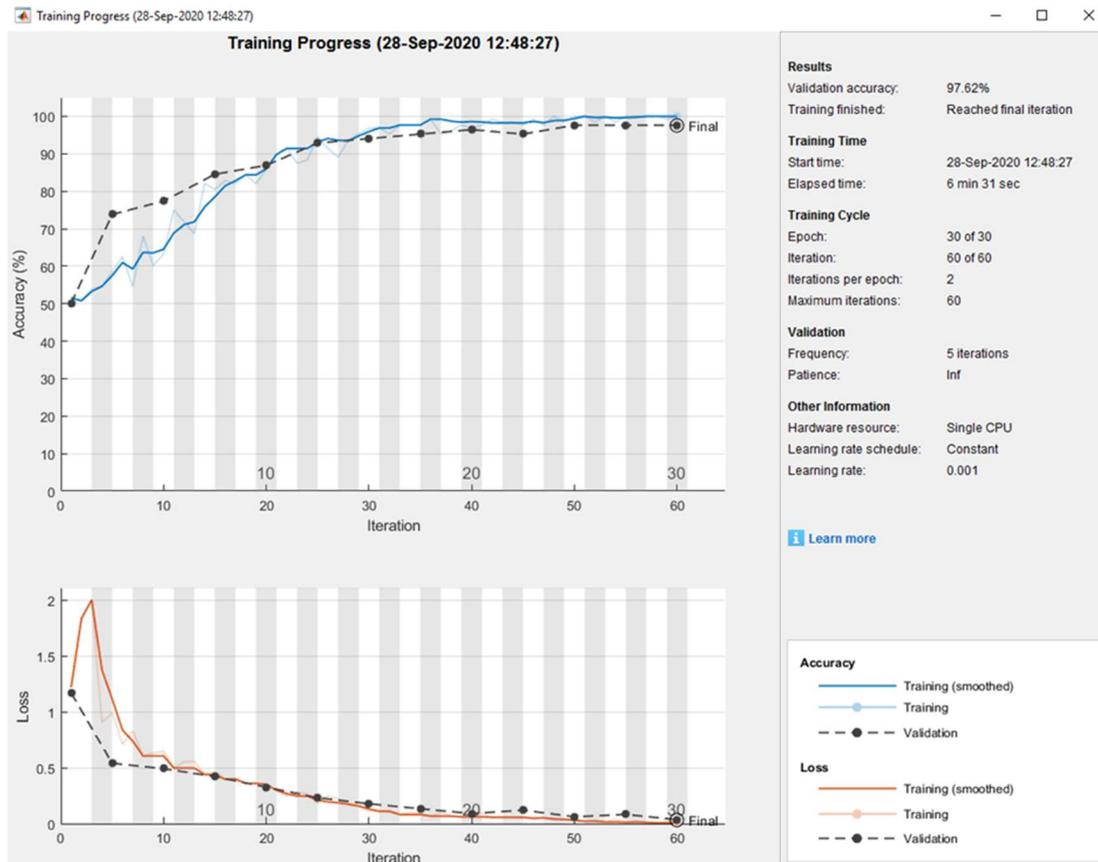
E3: Training progress for 8.5:1.5 ratio on training and test set



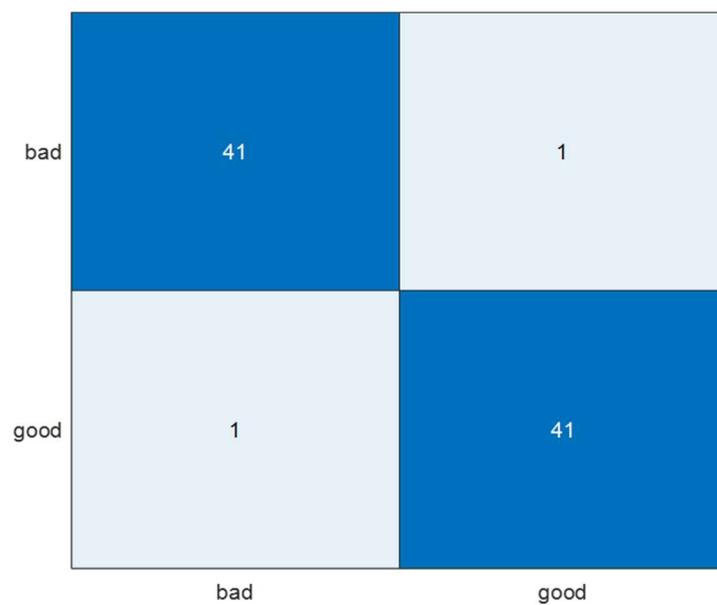
E4: Confusion matrix for 8.5:1.5 ratio on training and test set



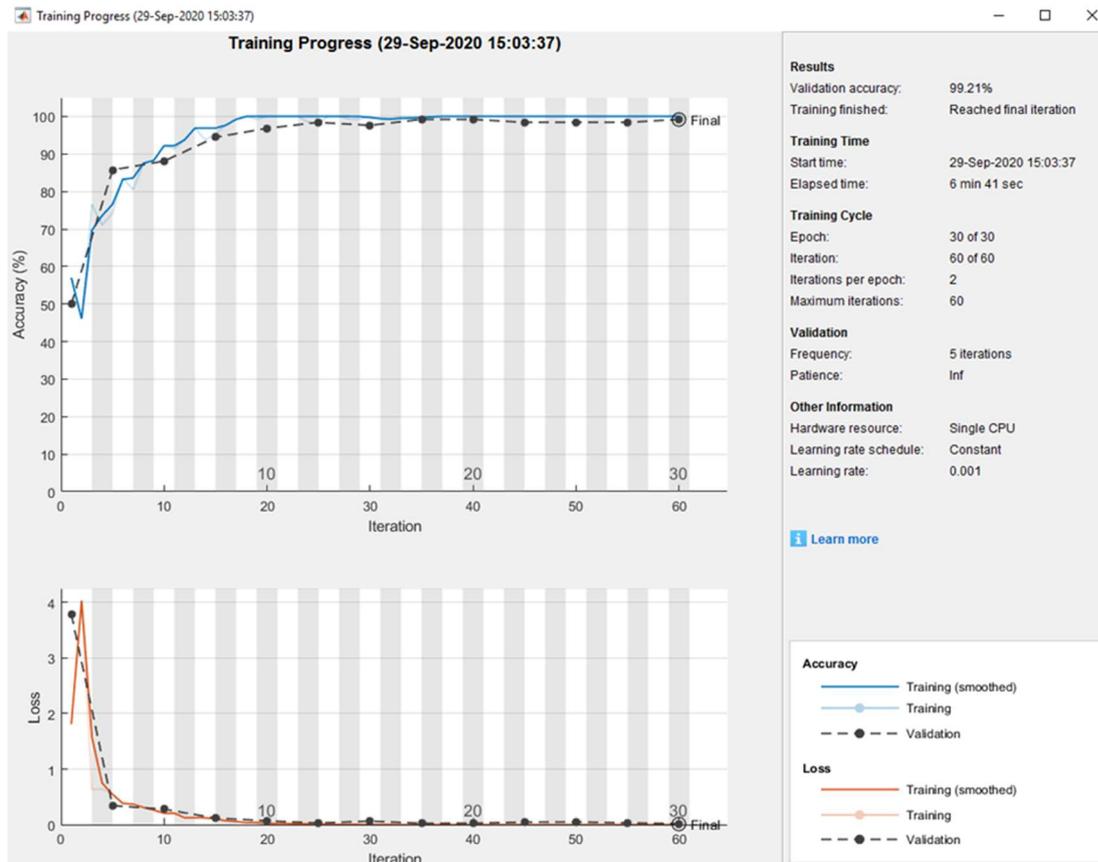
E5: Training progress for 4:1 ratio on training and test set



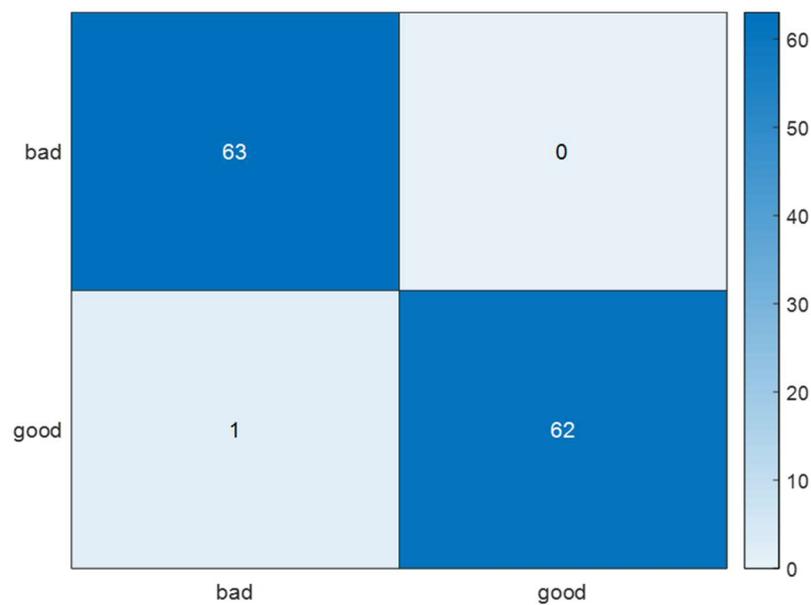
E6: Confusion matrix for 4:1 ratio on training and test set



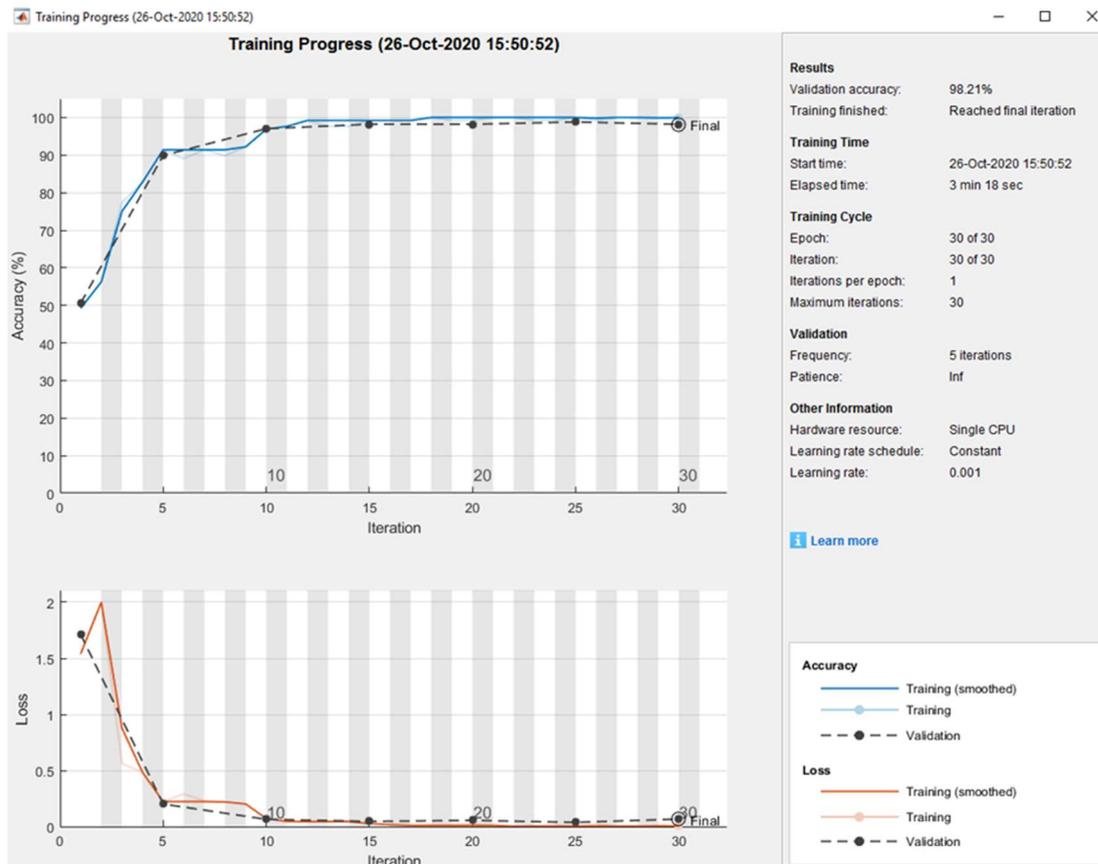
E7: Training progress for 7:3 ratio on training and test set



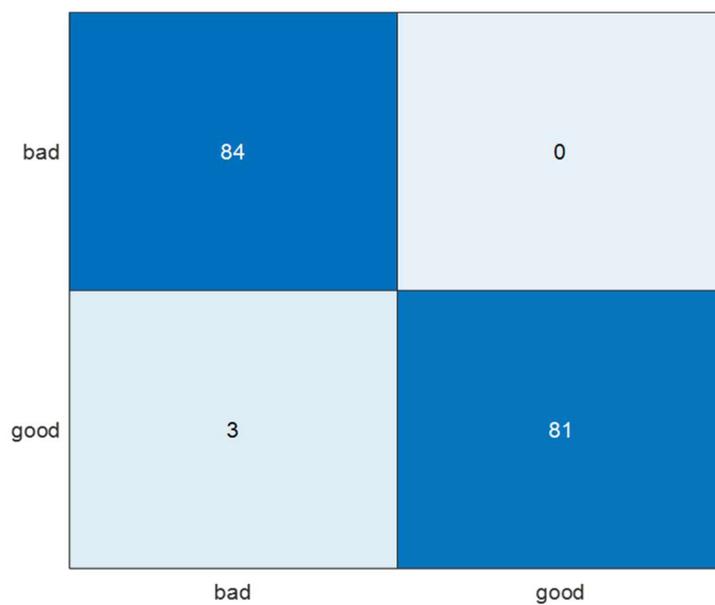
E8: Confusion matrix for 7:3 ratio on training and test set



E9: Training progress for 3:2 ratio on training and test set

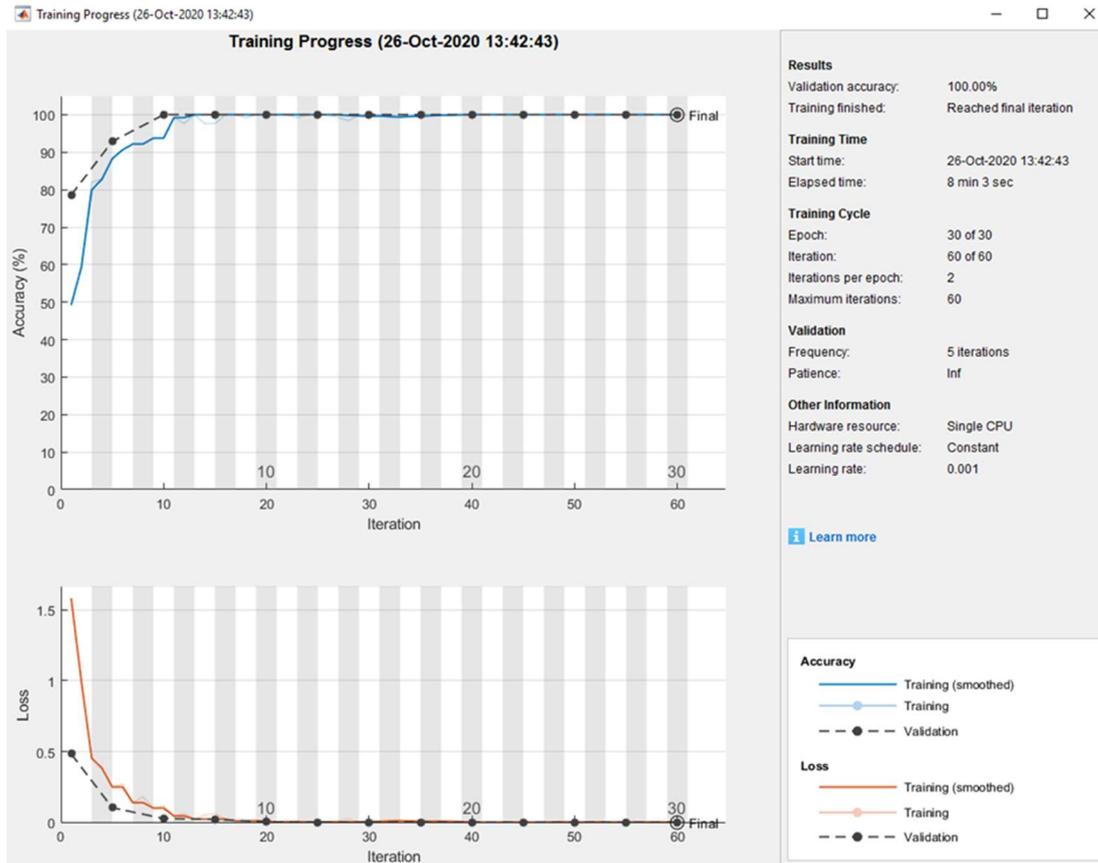


E10: Confusion matrix for 3:2 ratio on training and test set

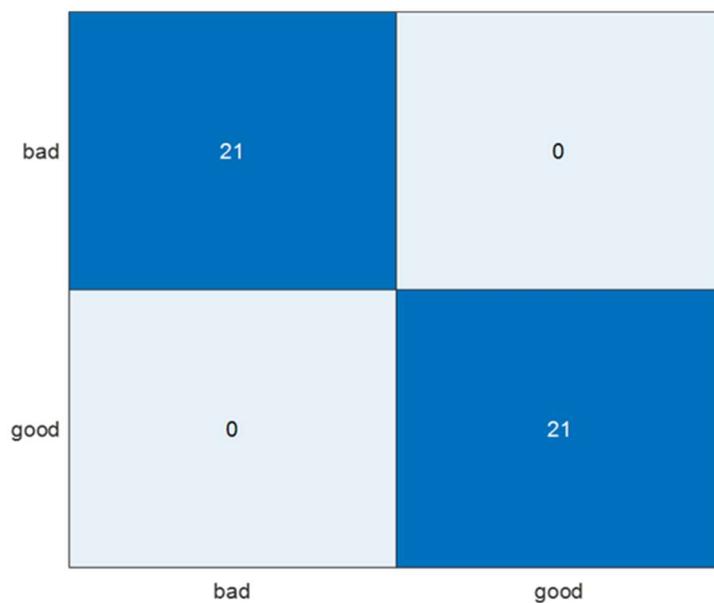


Appendix F: DCNN results for 20 °C thermal perturbation test

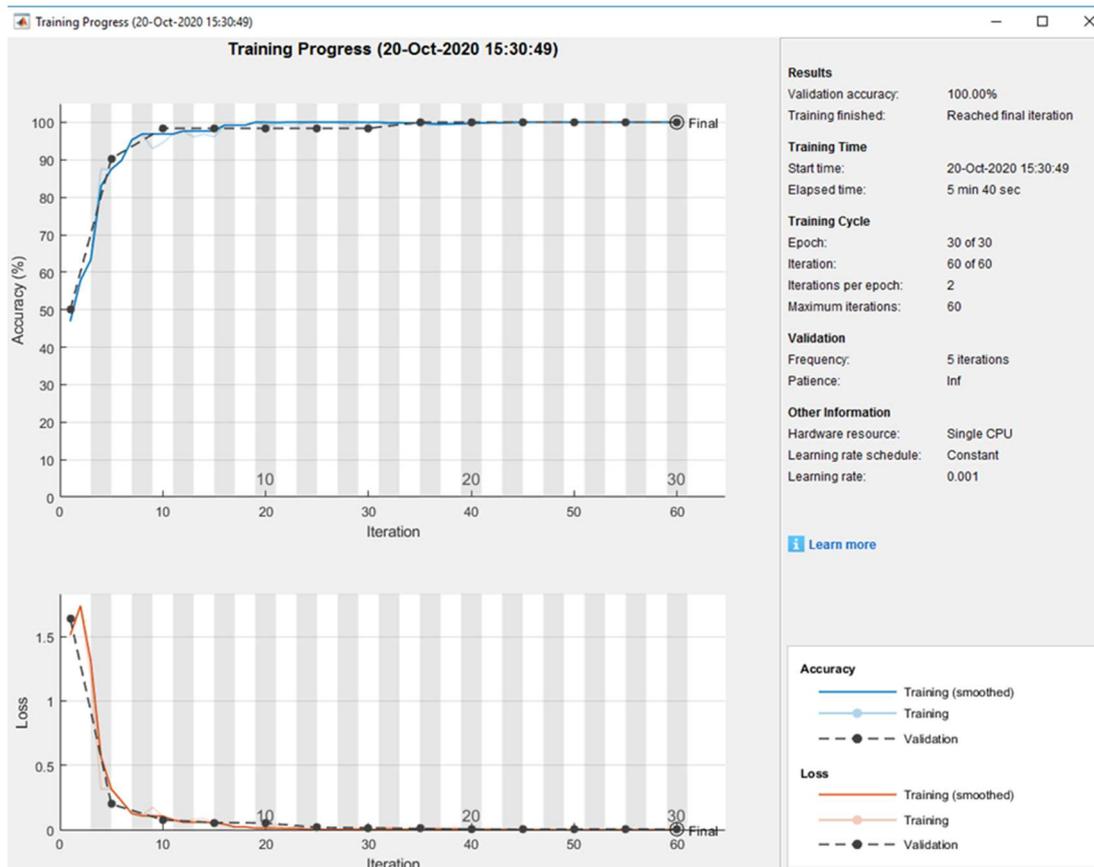
F1: Training progress for 9:1 ratio on training and test set



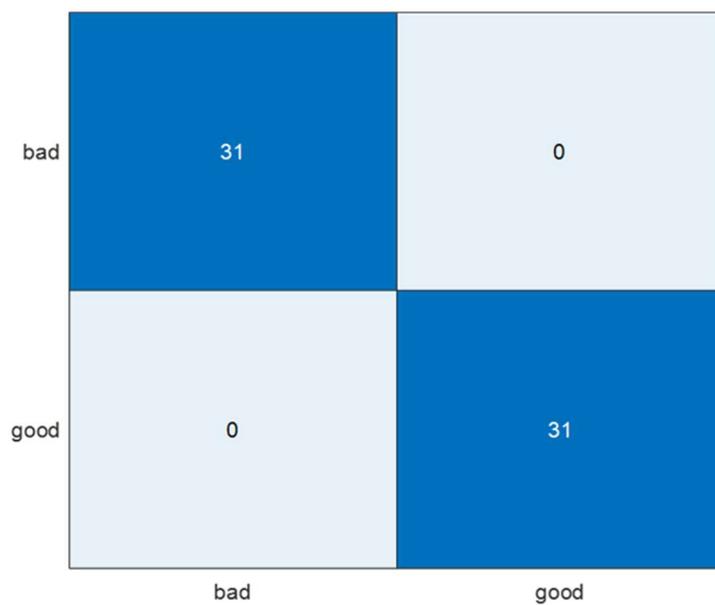
F2: Confusion matrix for 9:1 ratio on training and test set



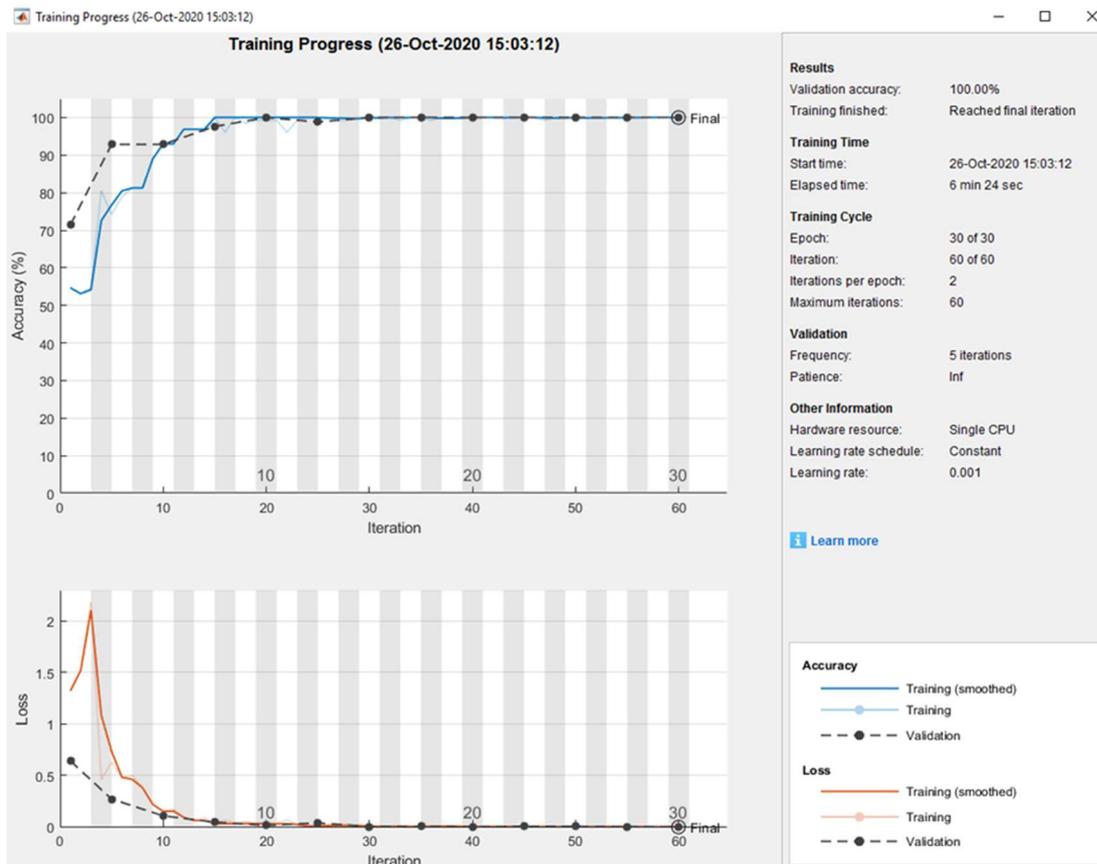
F3: Training progress for 8.5:1.5 ratio on training and test set



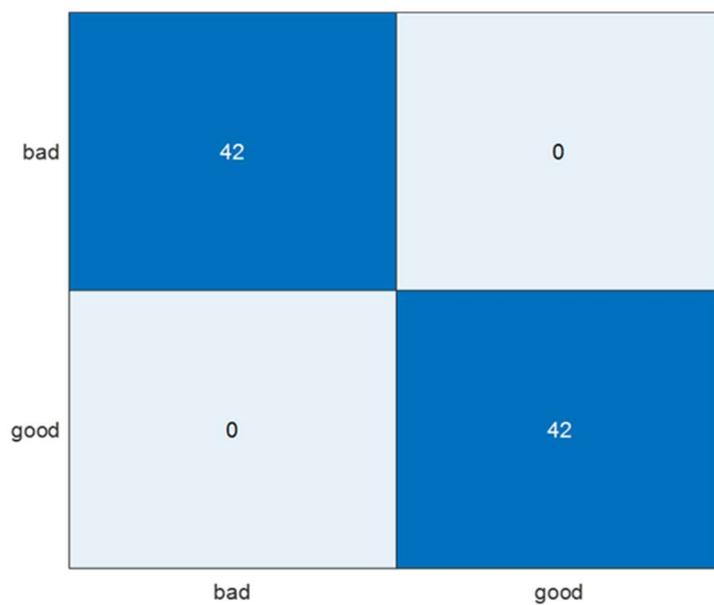
F4: Confusion matrix for 8.5:1.5 ratio on training and test set



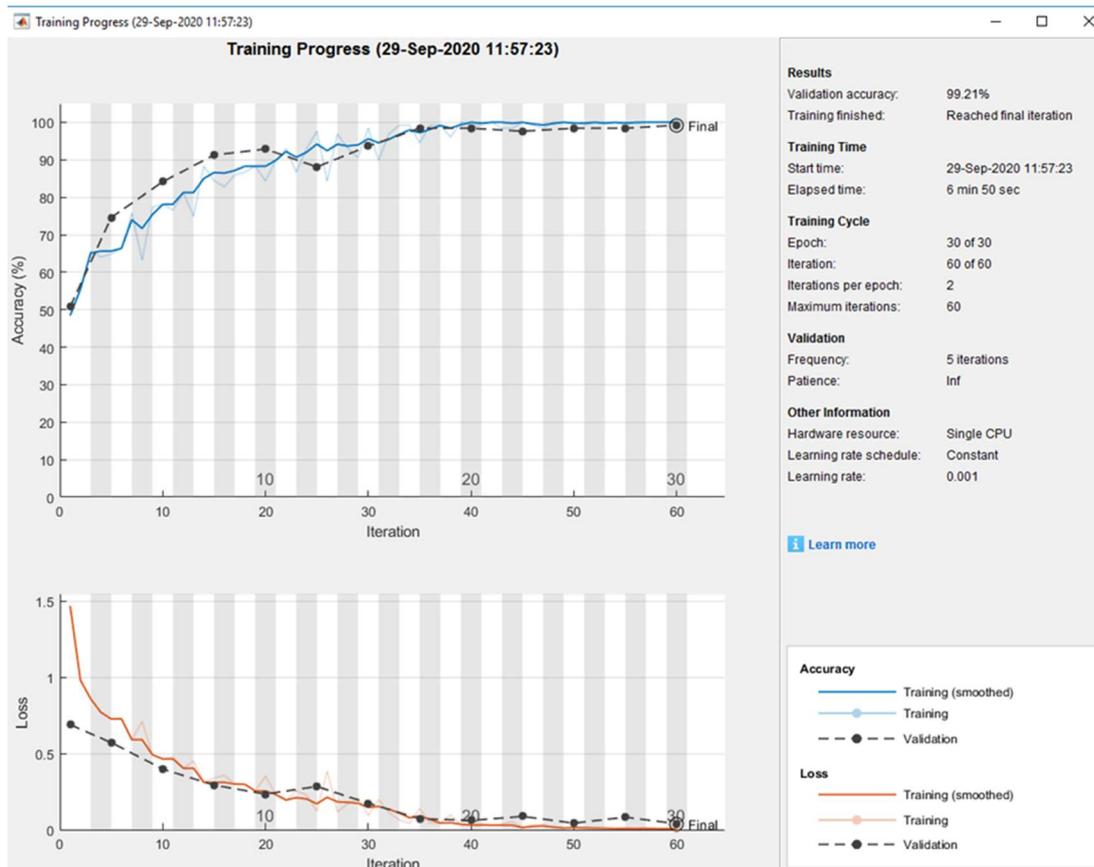
F5: Training progress for 4:1 ratio on training and test set



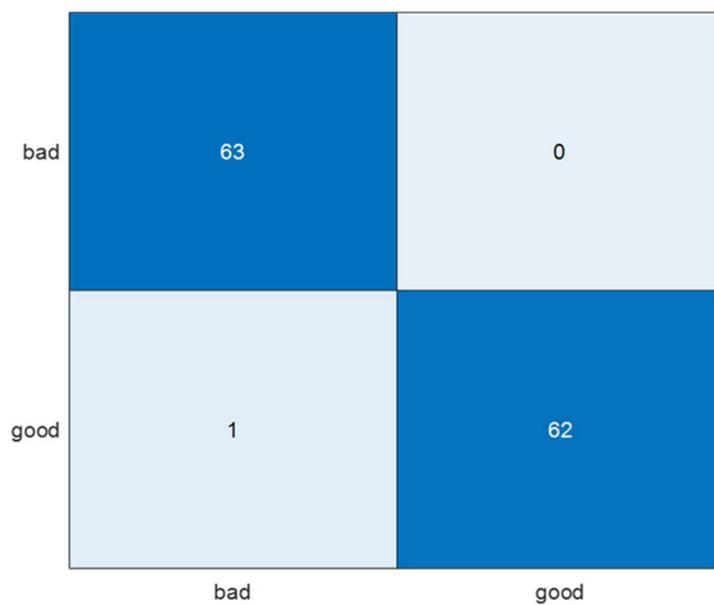
F6: Confusion matrix for 4:1 ratio on training and test set



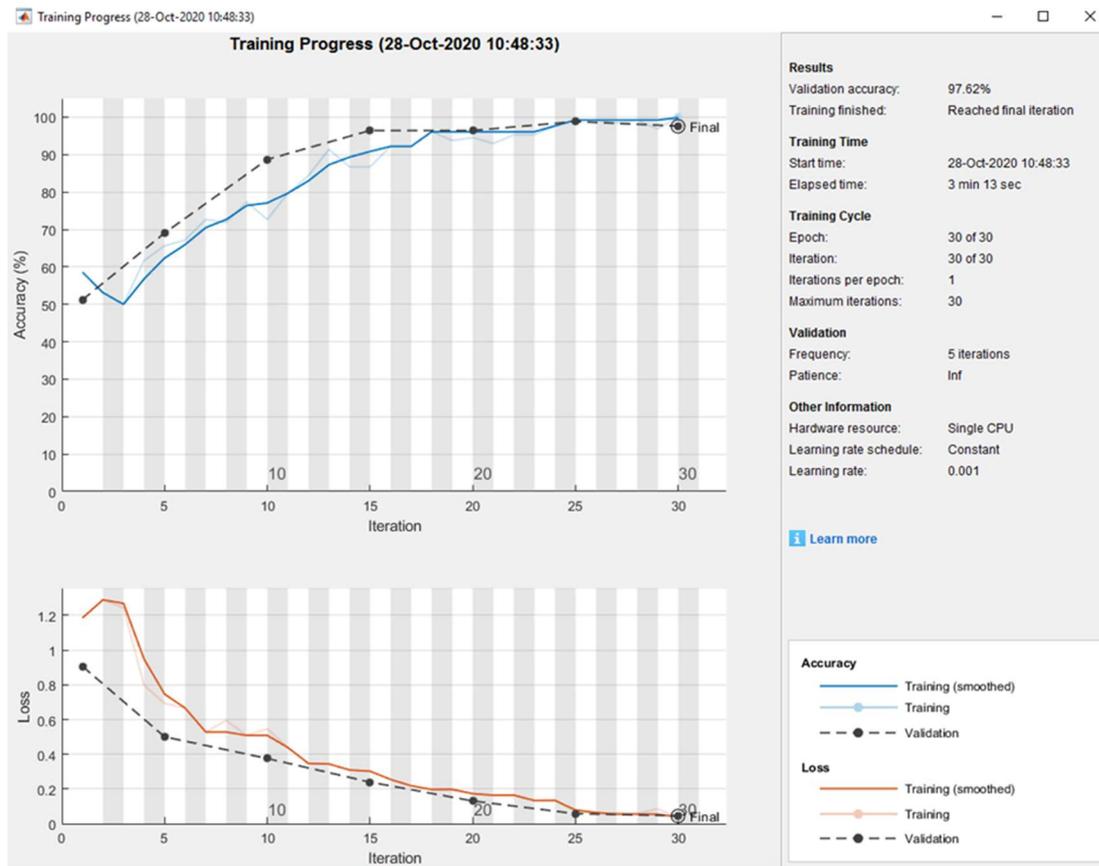
F7: Training progress for 7:3 ratio on training and test set



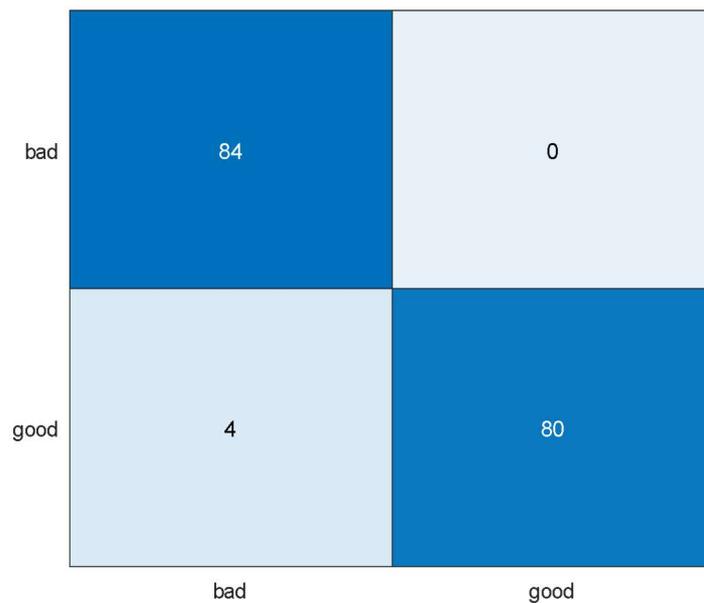
F8: Confusion matrix for 7:3 ratio on training and test set



F9: Training progress for 3:2 ratio on training and test set

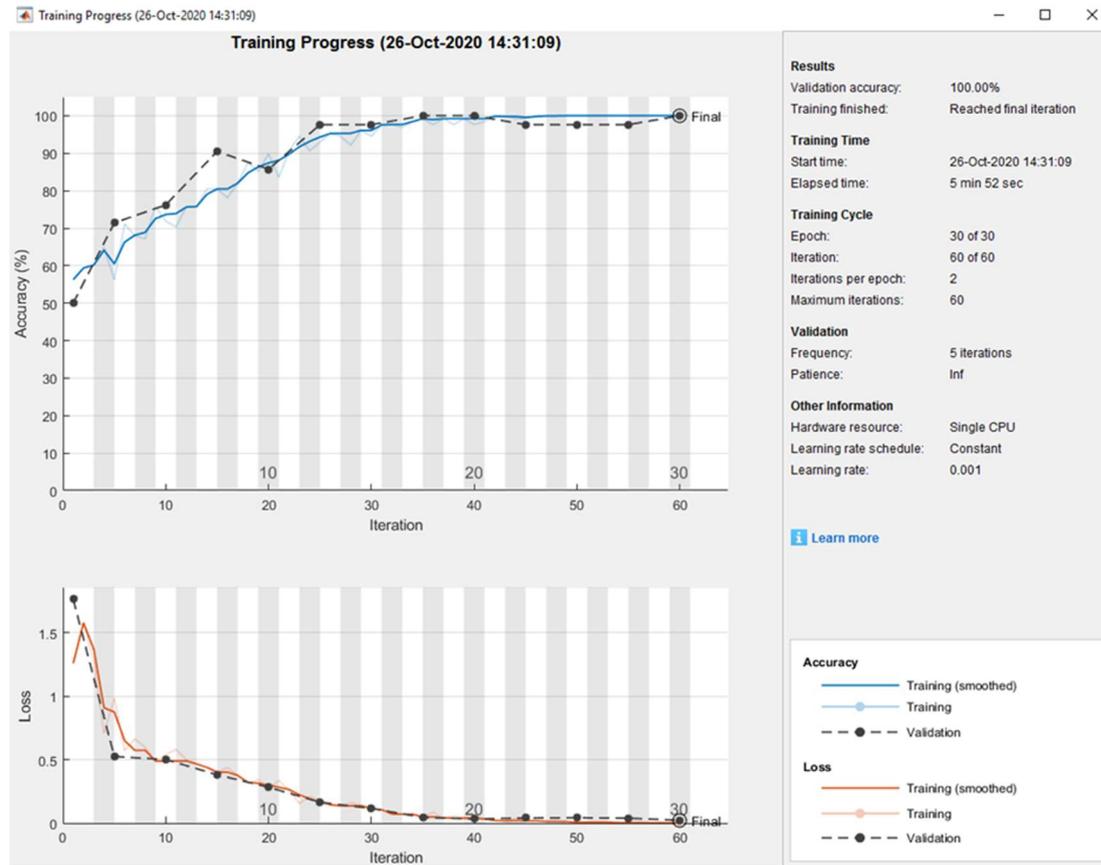


F10: Confusion matrix for 3:2 ratio on training and test set

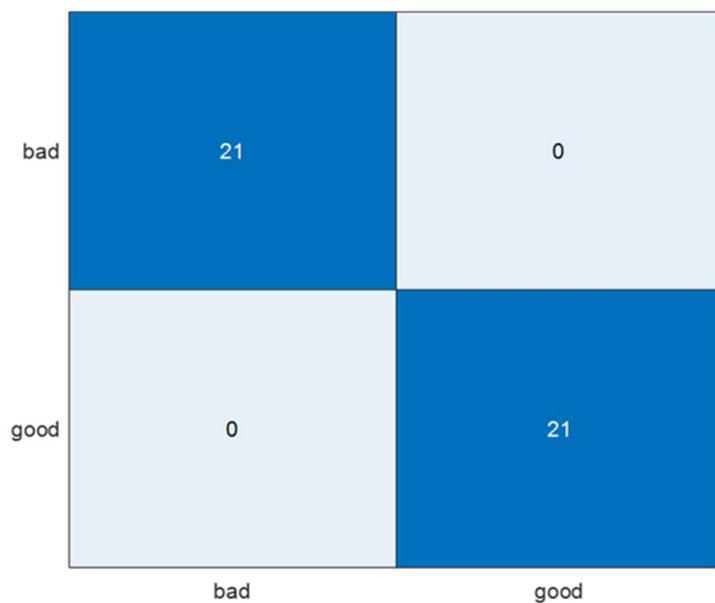


Appendix G: DCNN results for 25 °C thermal perturbation test

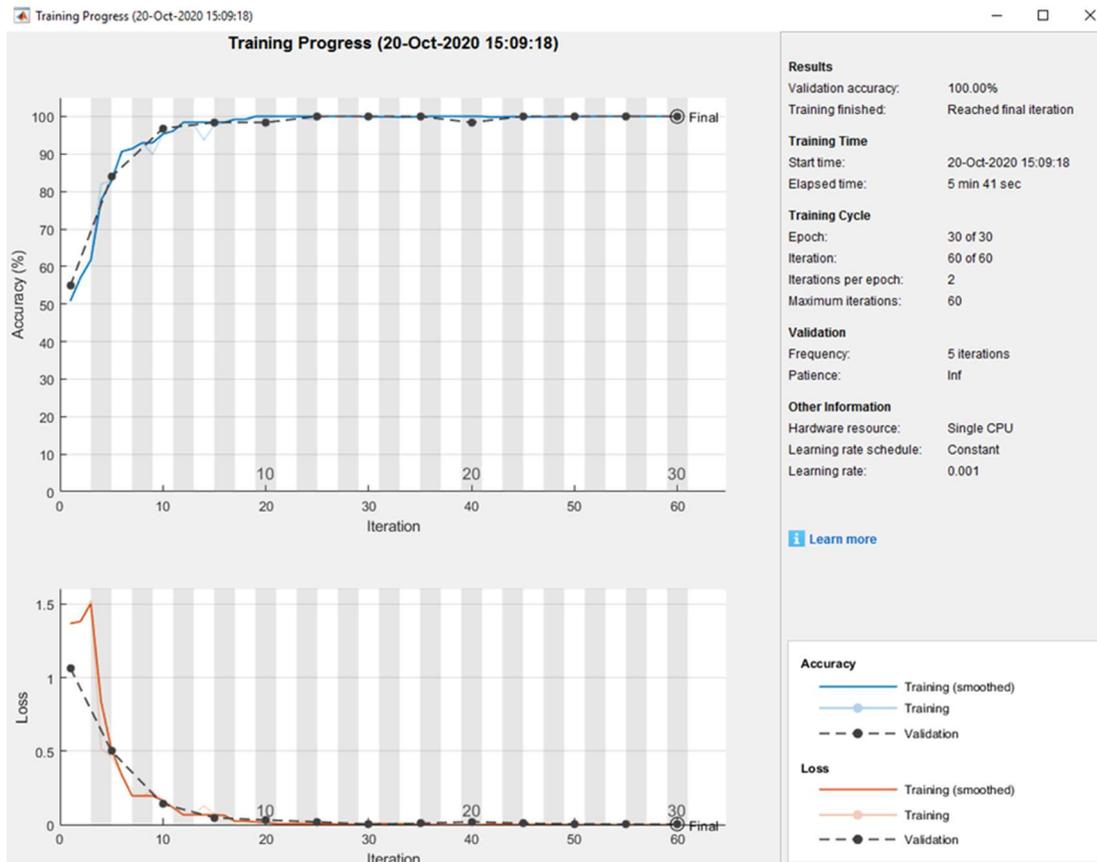
G1: Training progress for 9:1 ratio on training and test set



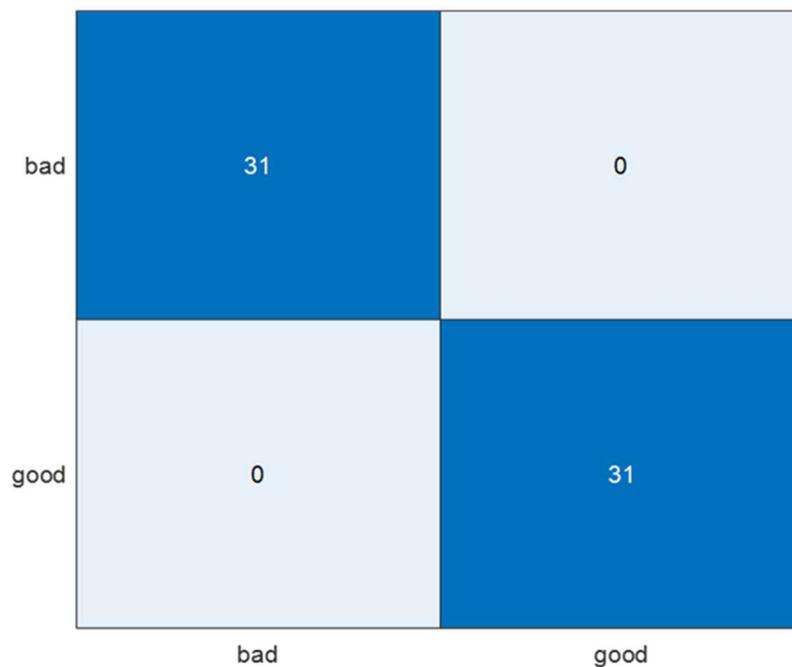
G2: Confusion matrix for 9:1 ratio on training and test set



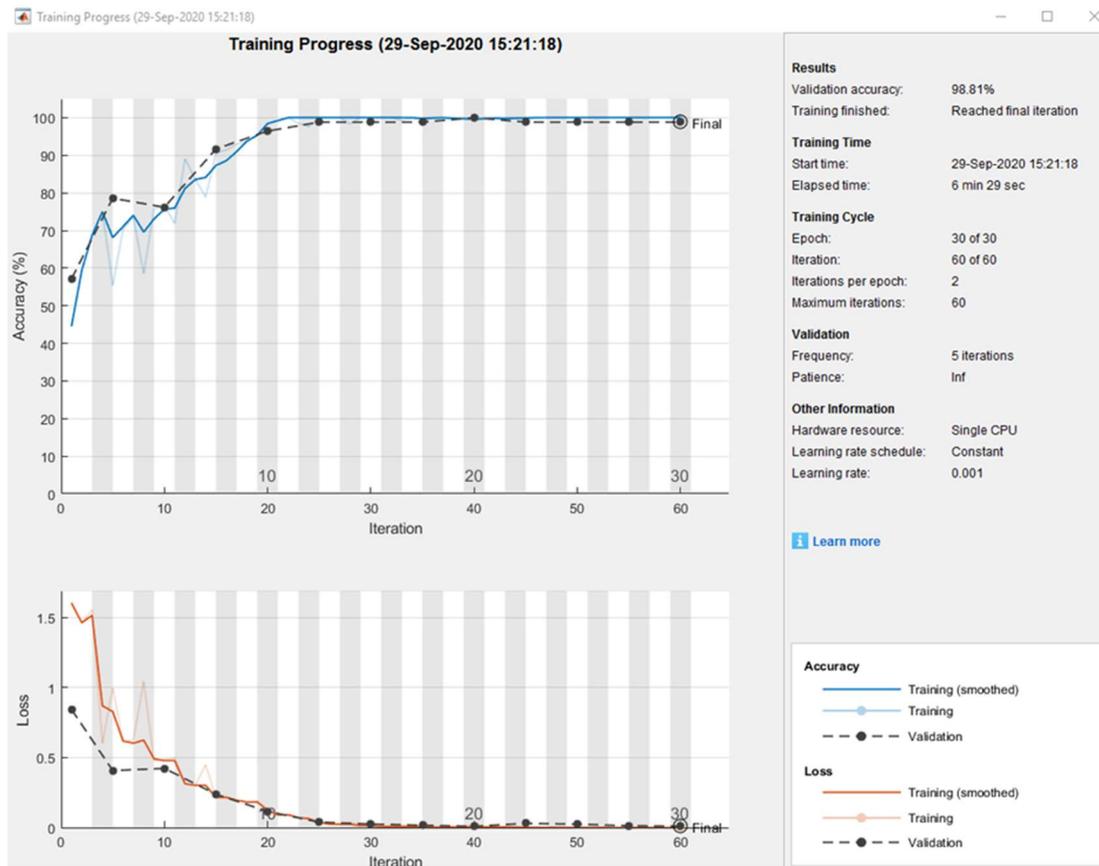
G3: Training progress for 8.5:1.5 ratio on training and test set



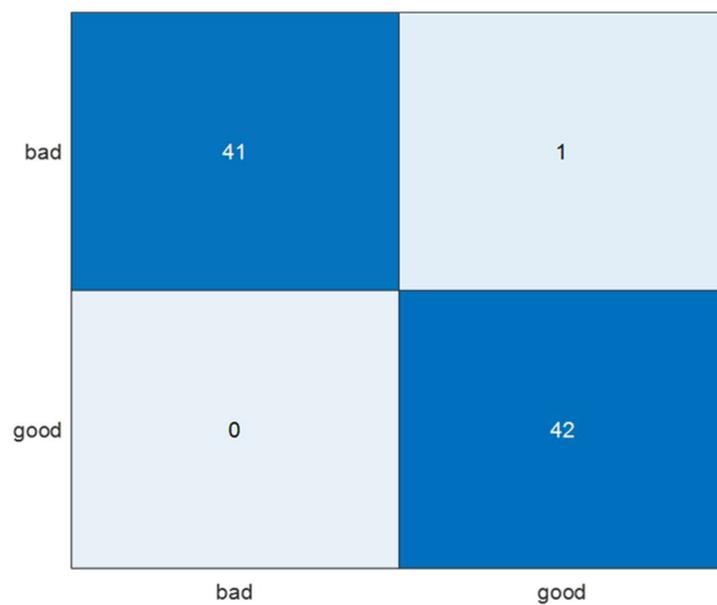
G4: Confusion matrix for 8.5:1.5 ratio on training and test set



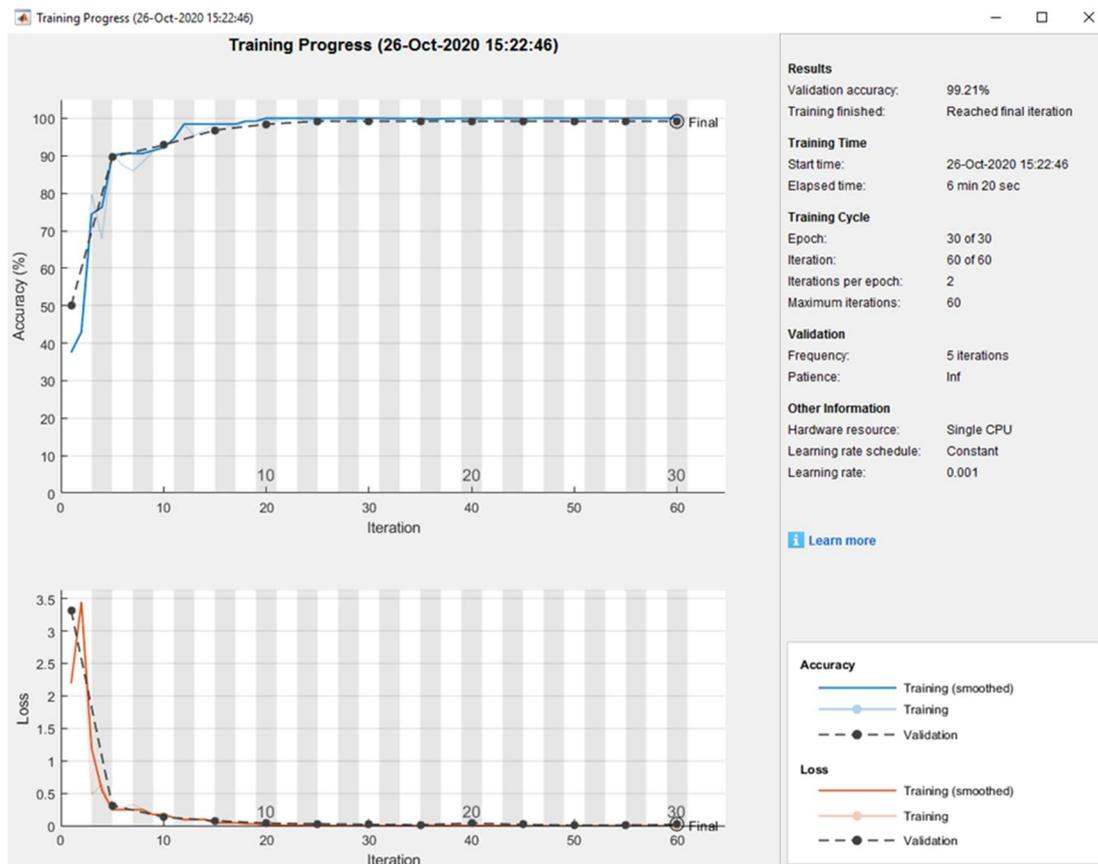
G5: Training progress for 4:1 ratio on training and test set



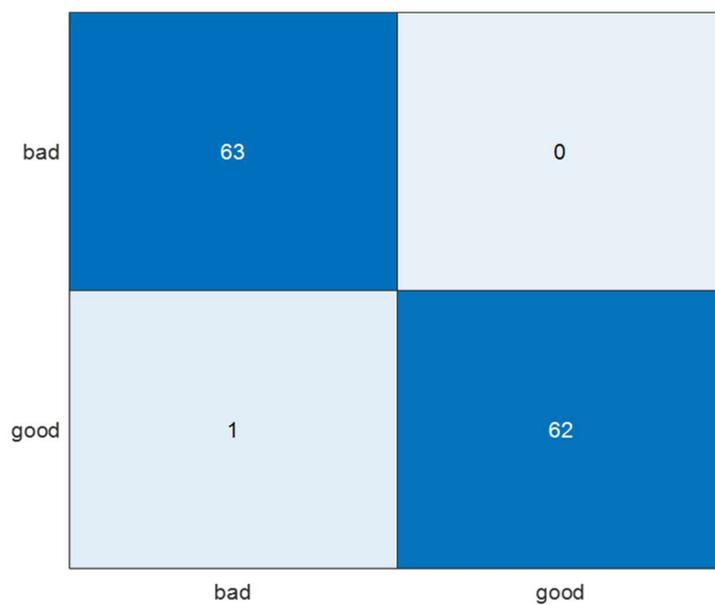
G6: Confusion matrix for 4:1 ratio on training and test set



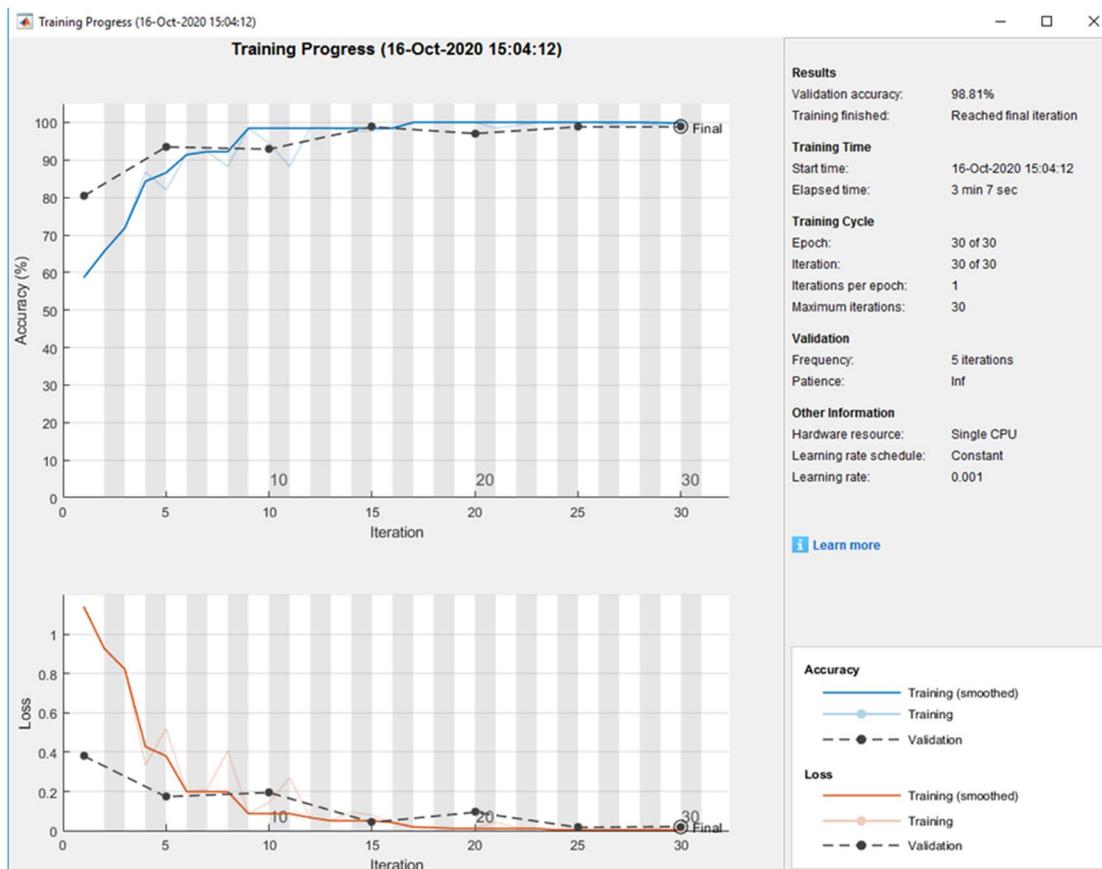
G7: Training progress for 7:3 ratio on training and test set



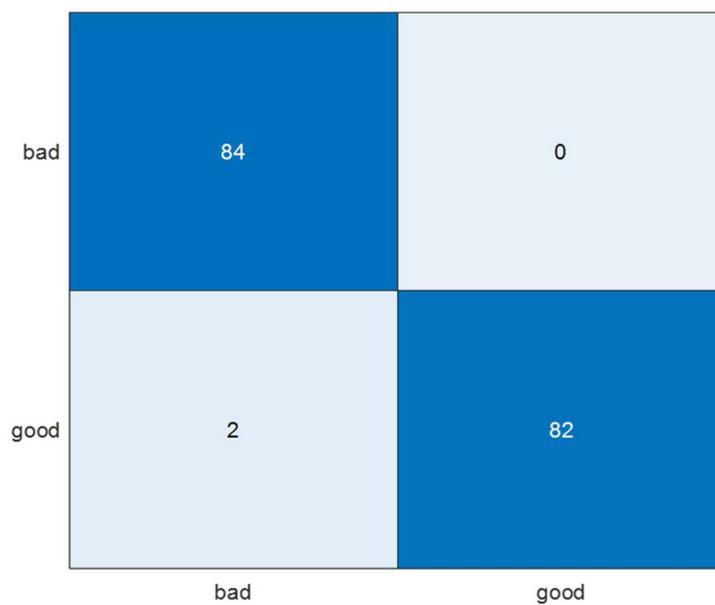
G8: Confusion matrix for 7:3 ratio on training and test set



G9: Training progress for 3:2 ratio on training and test set

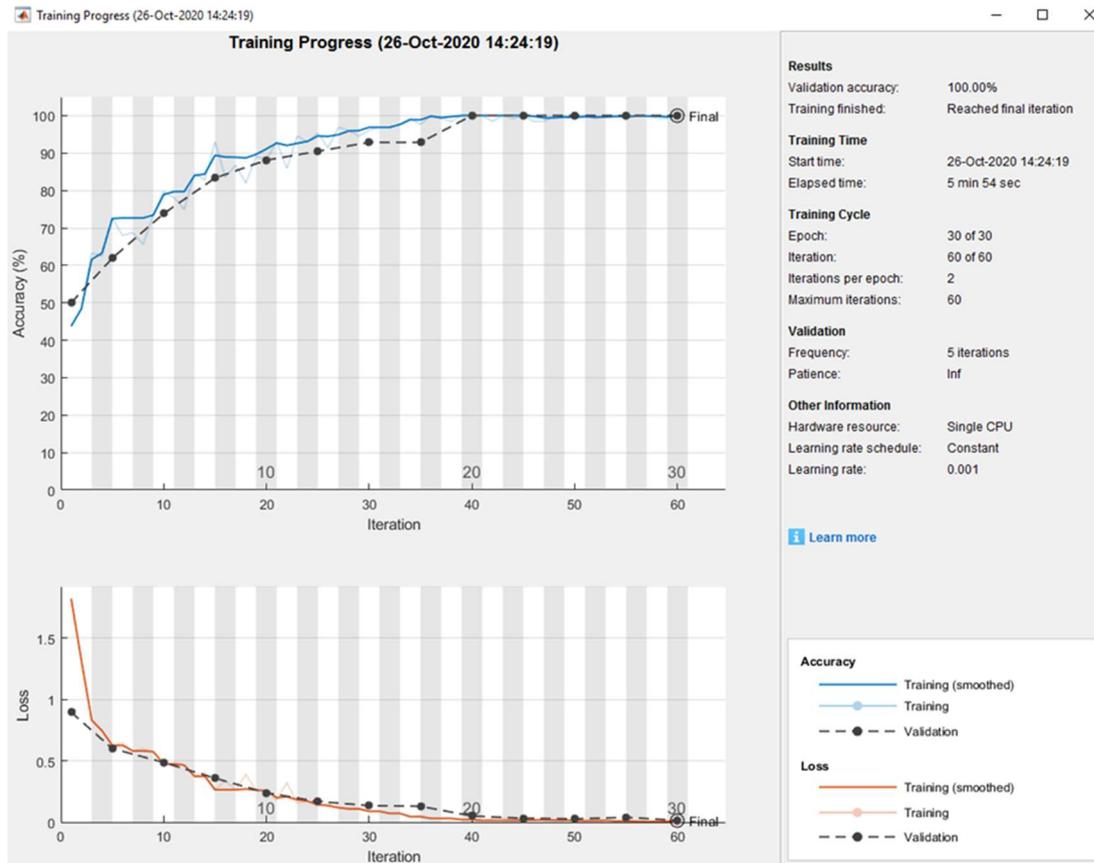


G10: Confusion matrix for 3:2 ratio on training and test set

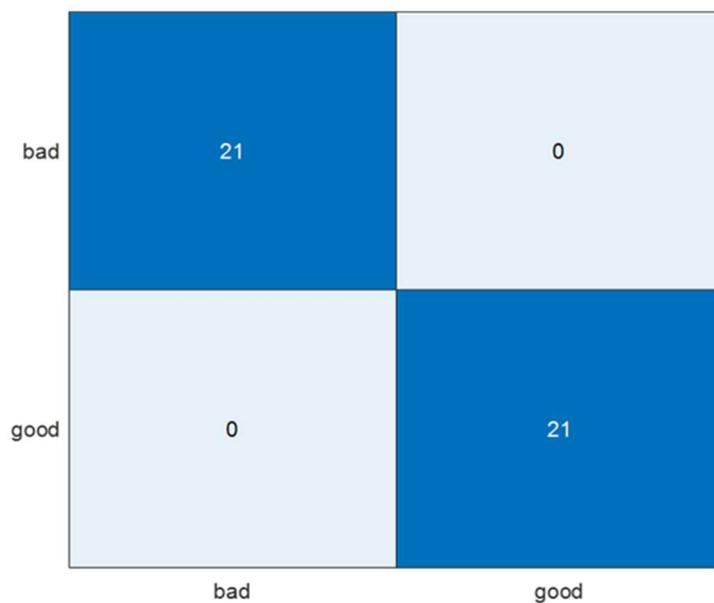


Appendix H: DCNN results for non-centuple feed rates test

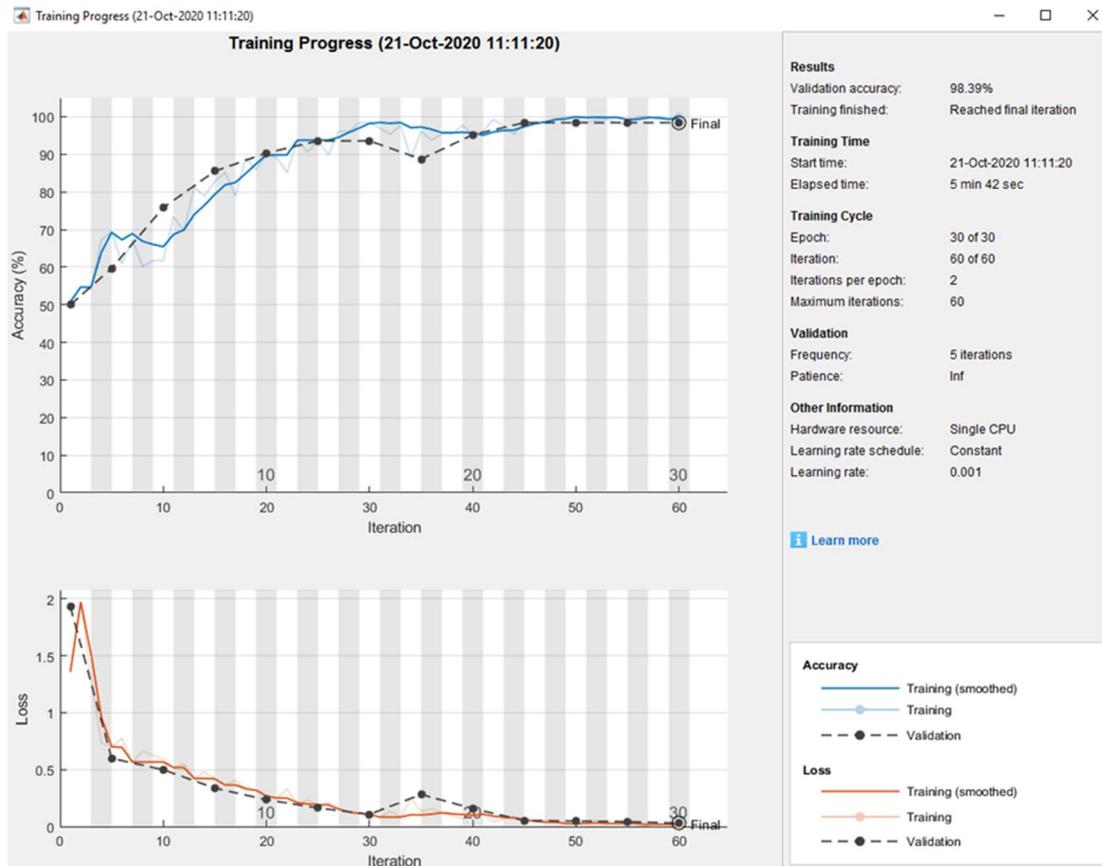
H1: Training progress for 9:1 ratio on training and test set



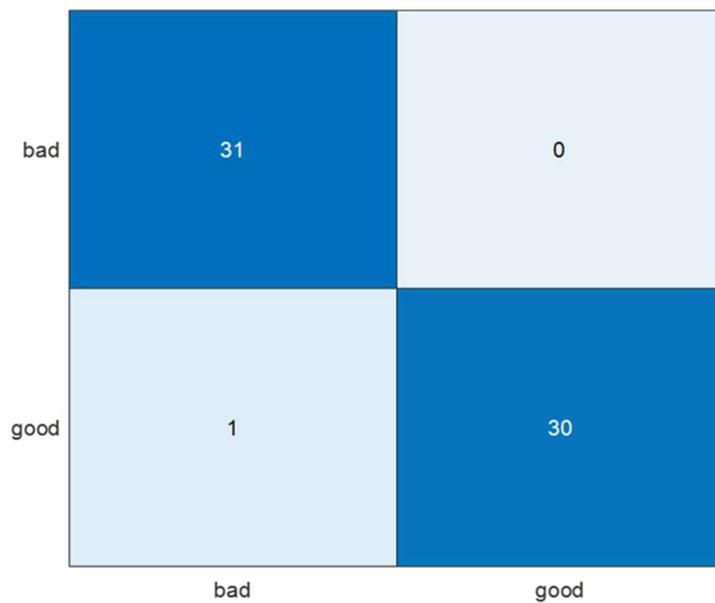
H2: Confusion matrix for 9:1 ratio on training and test set



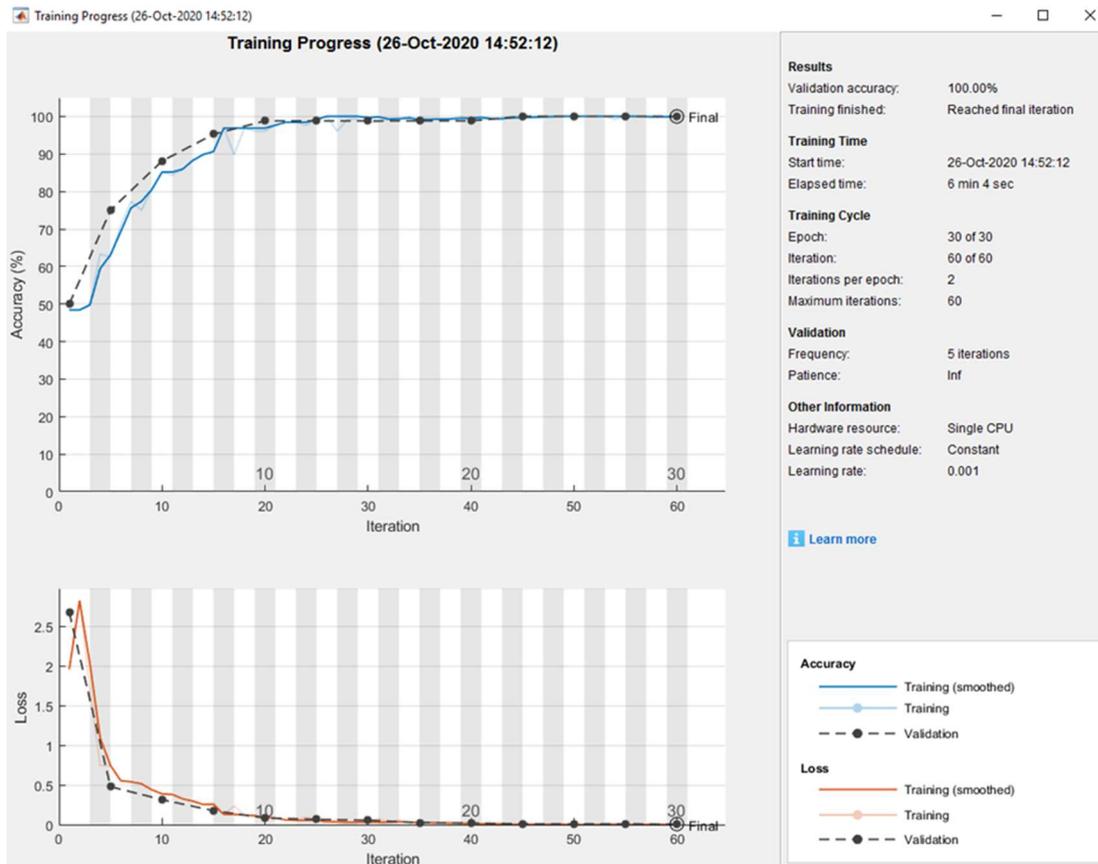
H3: Training progress for 8.5:1.5 ratio on training and test set



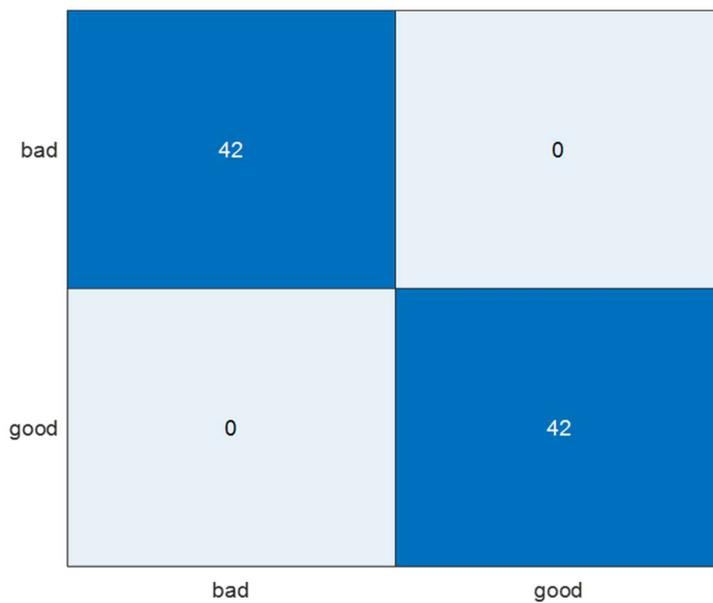
H4: Confusion matrix for 8.5:1.5 ratio on training and test set



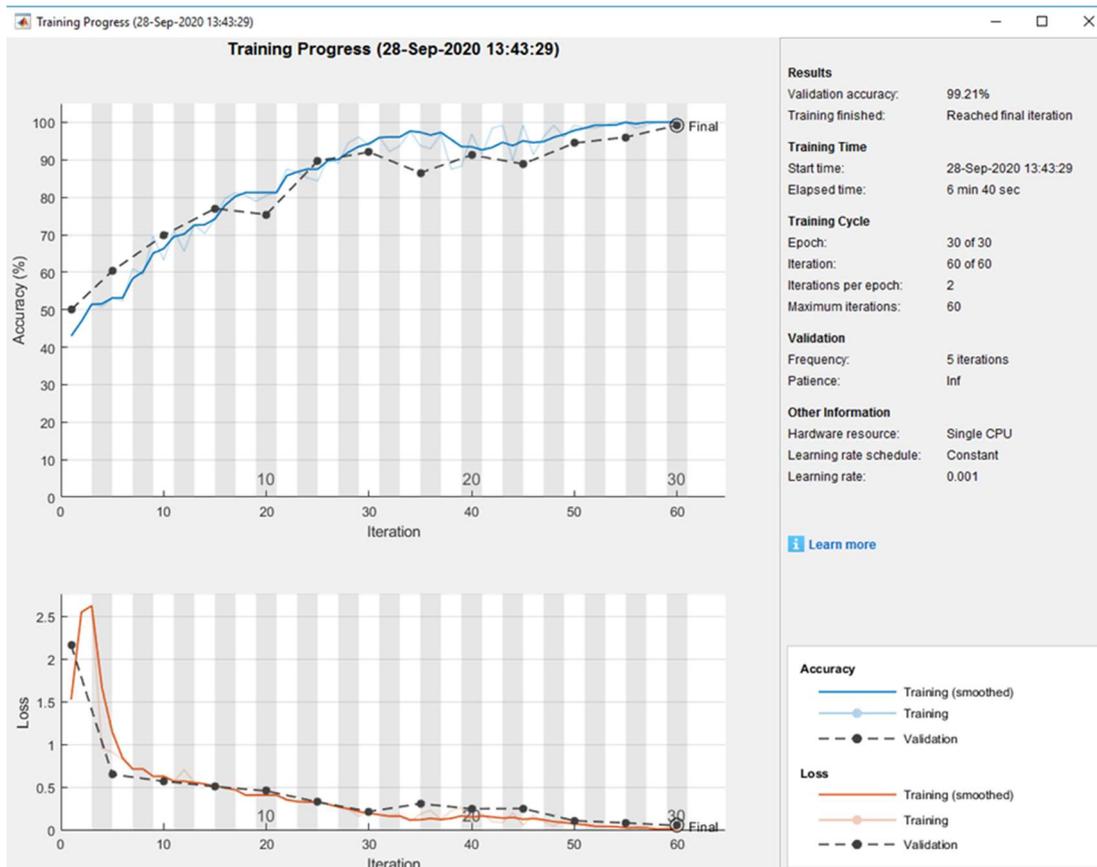
H5: Training progress for 4:1 ratio on training and test set



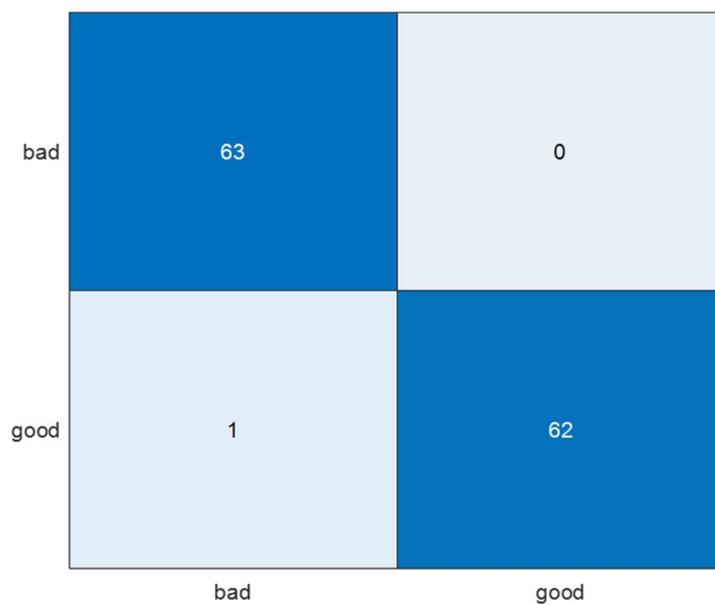
H6: Confusion matrix for 4:1 ratio on training and test set



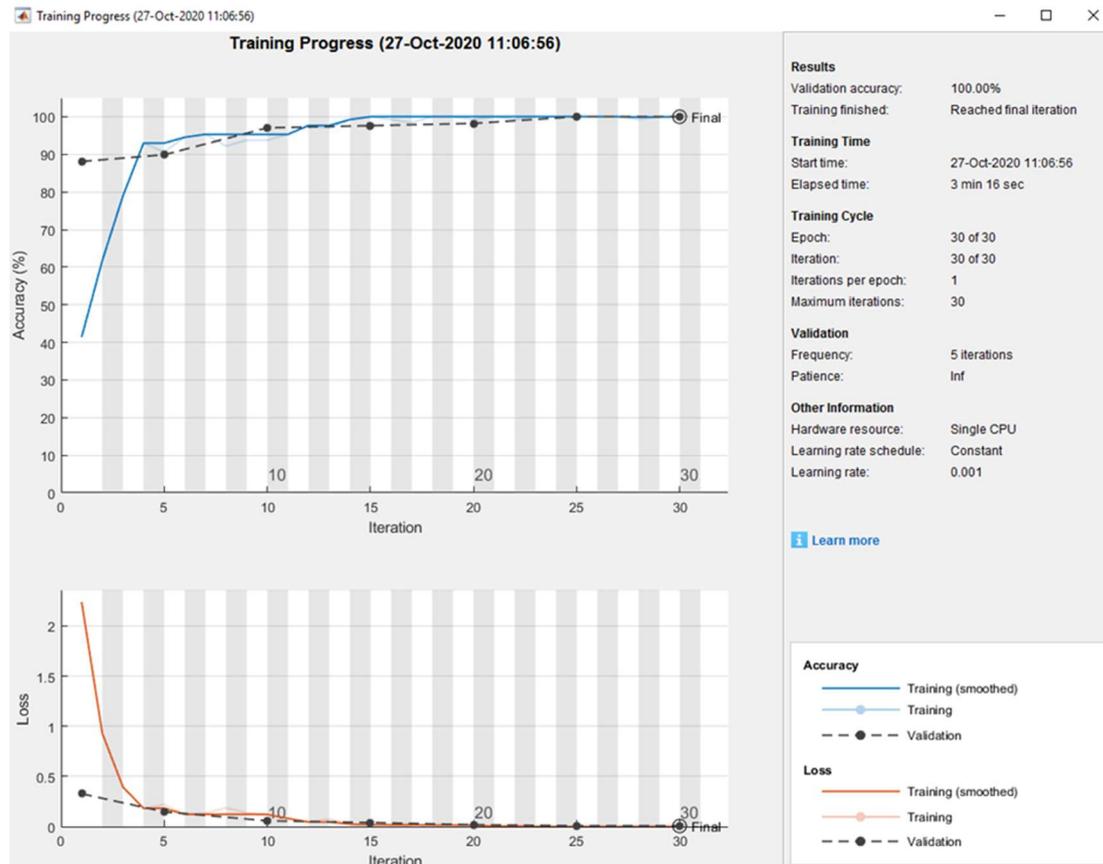
H7: Training progress for 7:3 ratio on training and test set



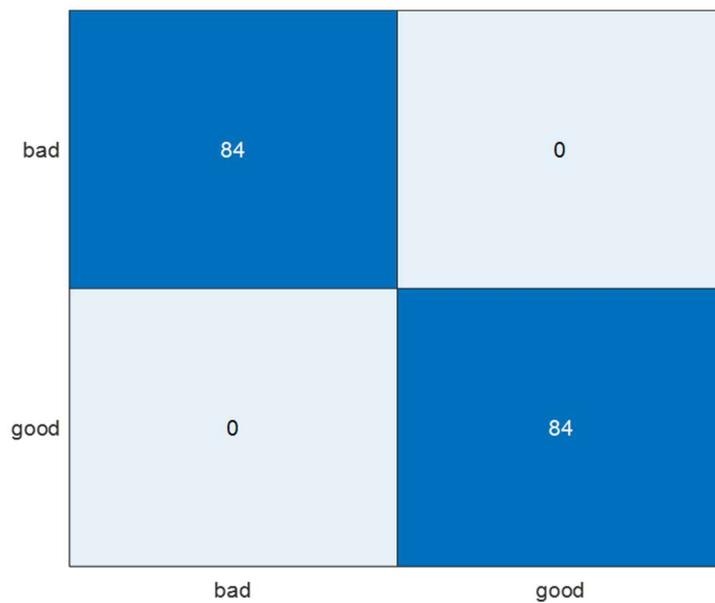
H8: Confusion matrix for 7:3 ratio on training and test set



H9: Training progress for 3:2 ratio on training and test set

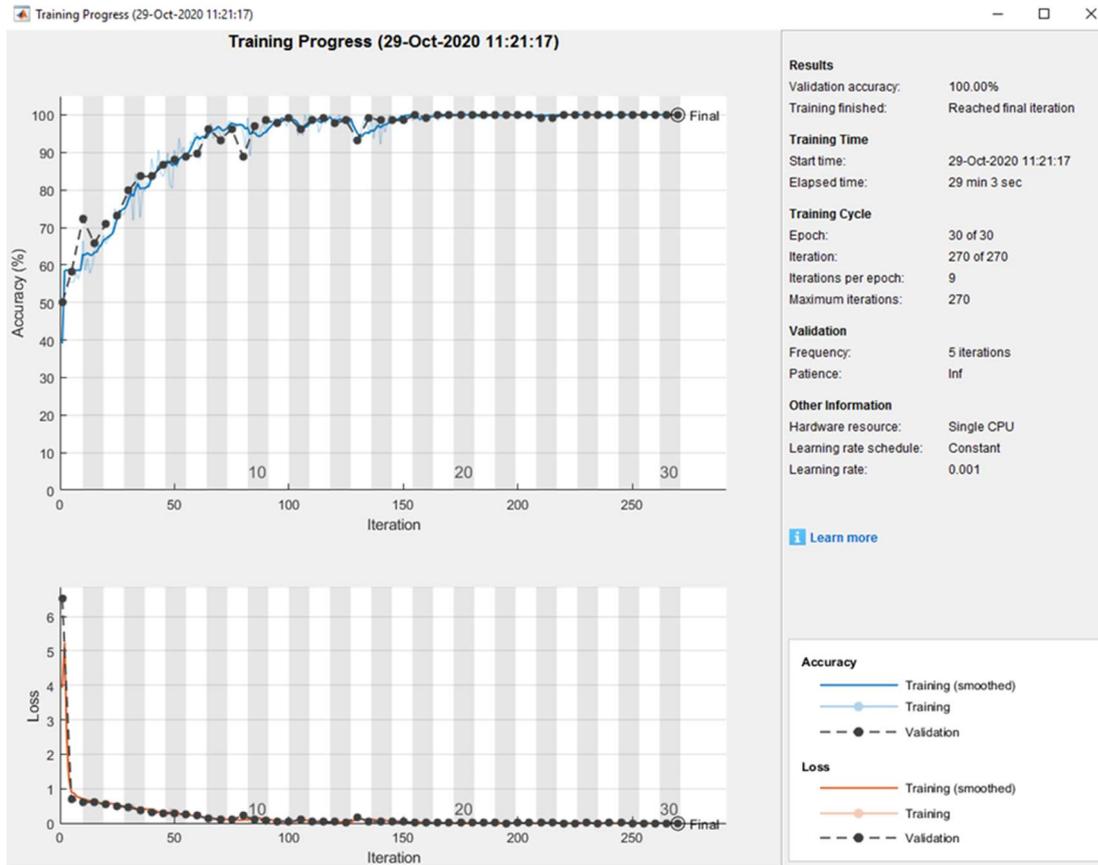


H10: Confusion matrix for 3:2 ratio on training and test set

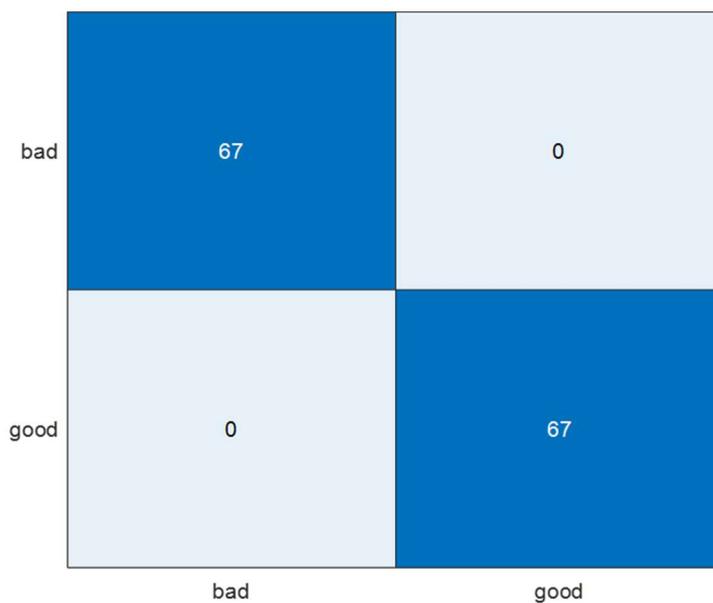


Appendix I: DCNN results for baseline test database

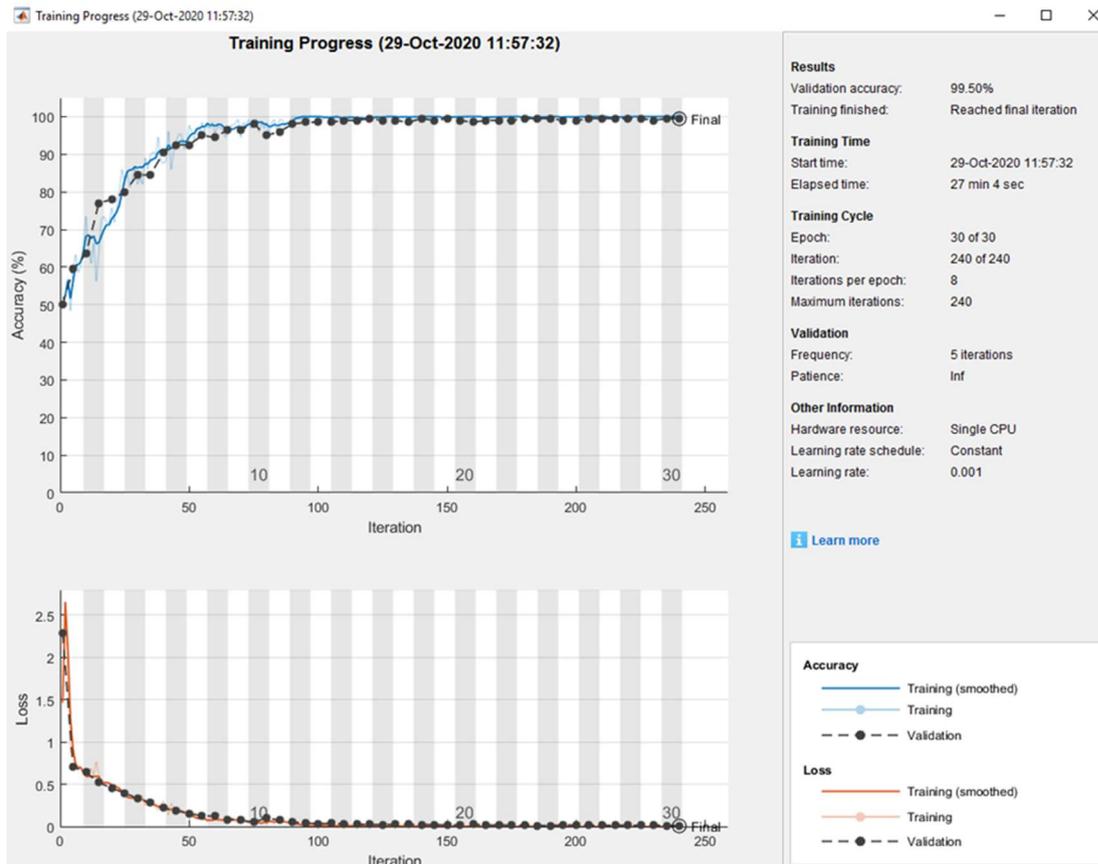
I1: Training progress for 9:1 ratio on training and test set



I2: Confusion matrix for 9:1 ratio on training and test set



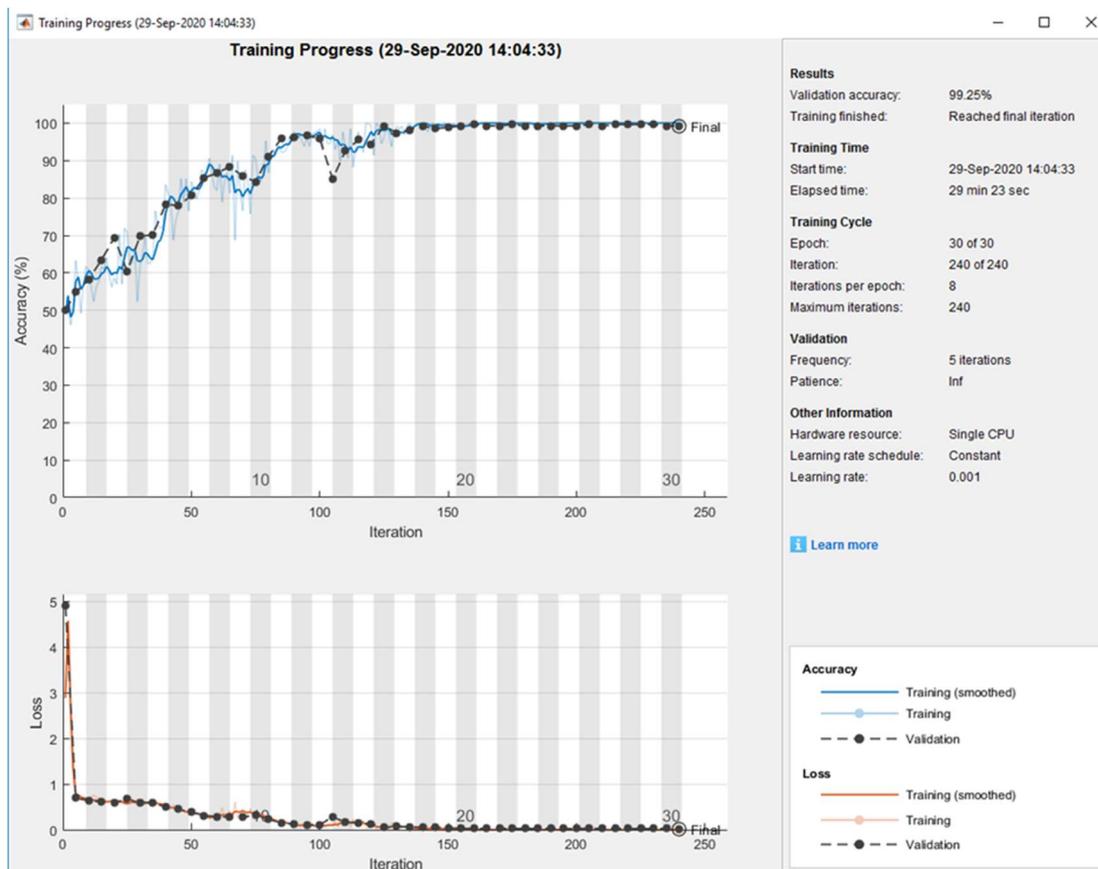
I3: Training progress for 8.5:1.5 ratio on training and test set



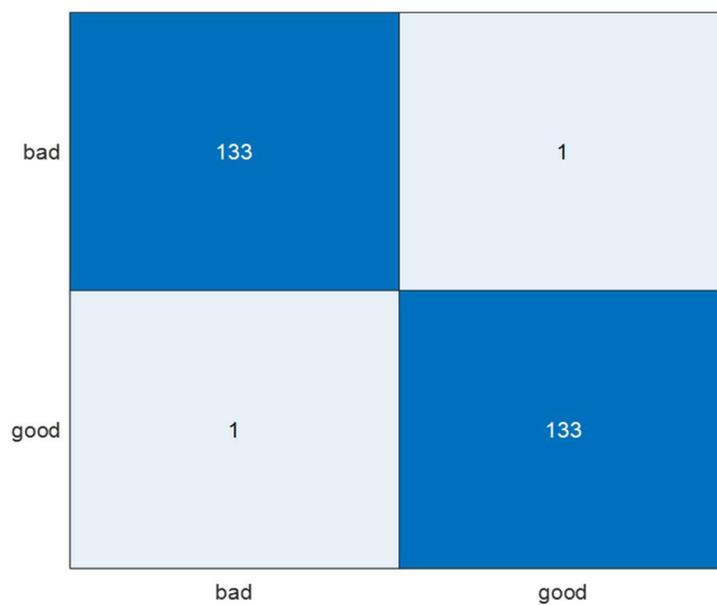
I4: Confusion matrix for 8.5:1.5 ratio on training and test set



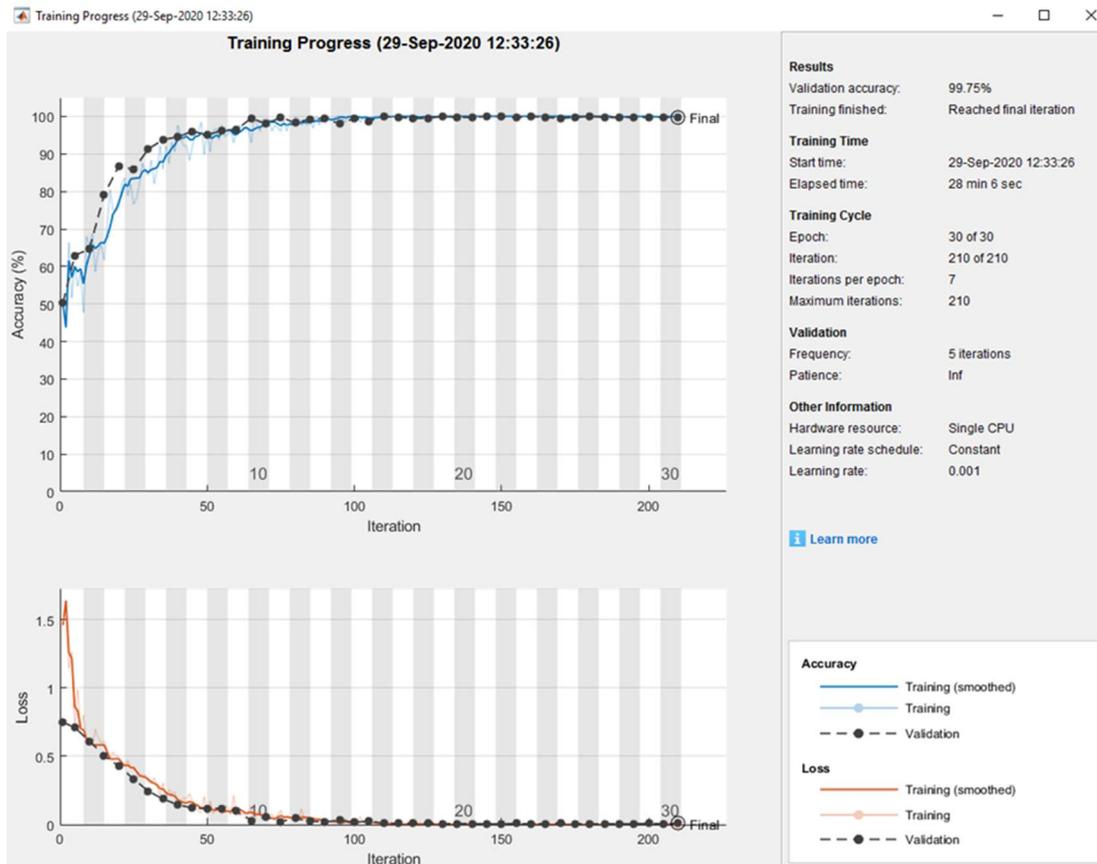
I5: Training progress for 4:1 ratio on training and test set



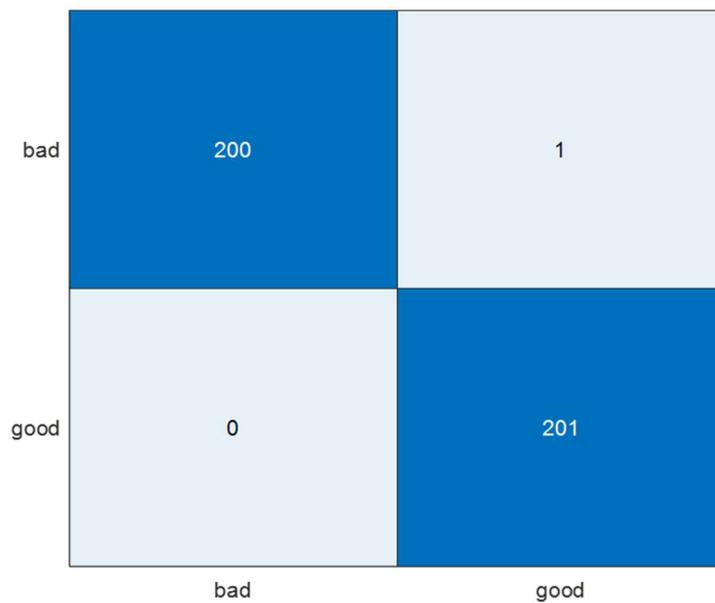
I6: Confusion matrix for 4:1 ratio on training and test set



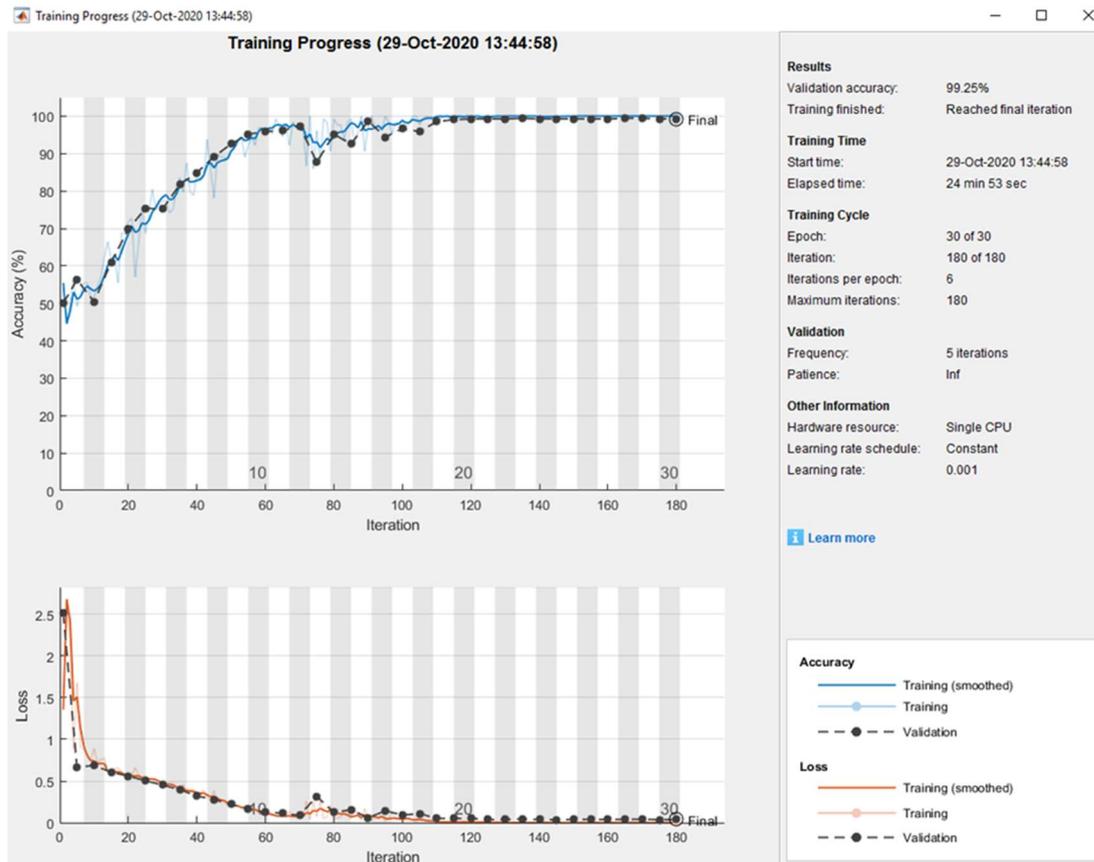
I7: Training progress for 7:3 ratio on training and test set



I8: Confusion matrix for 7:3 ratio on training and test set

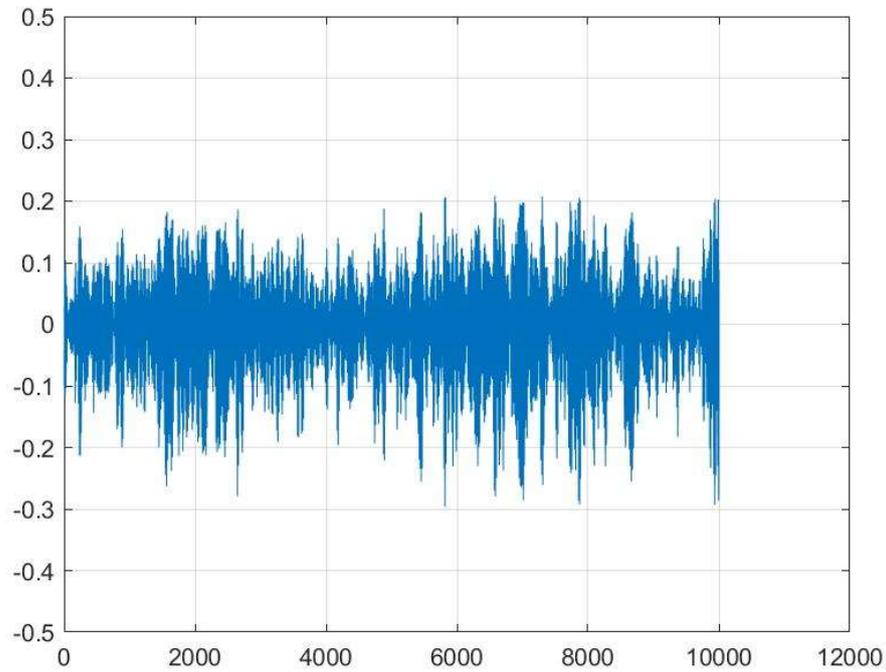
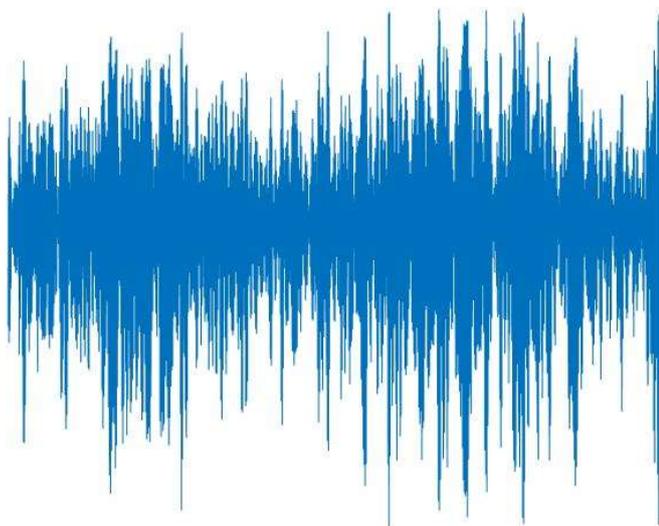


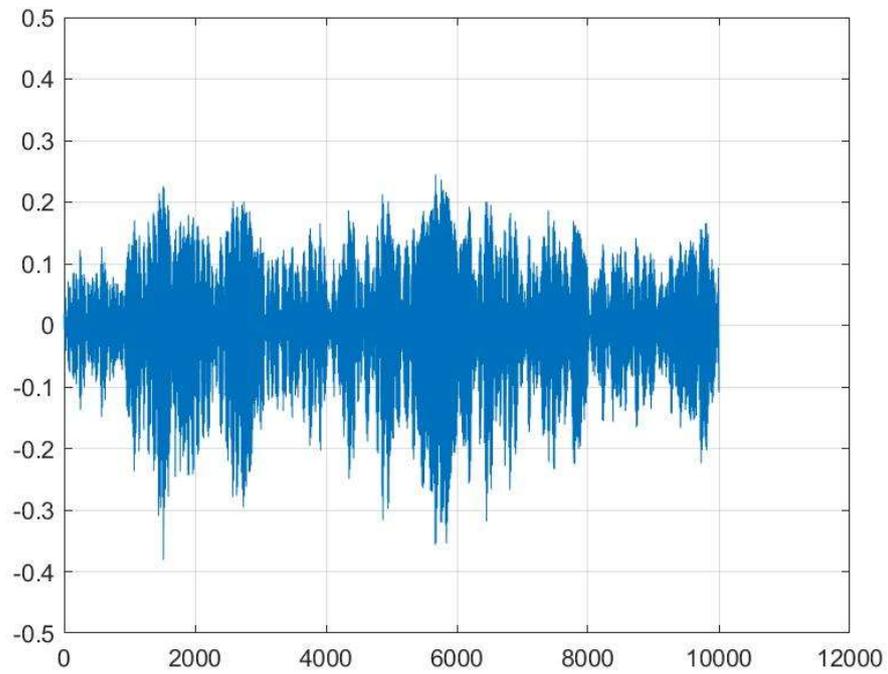
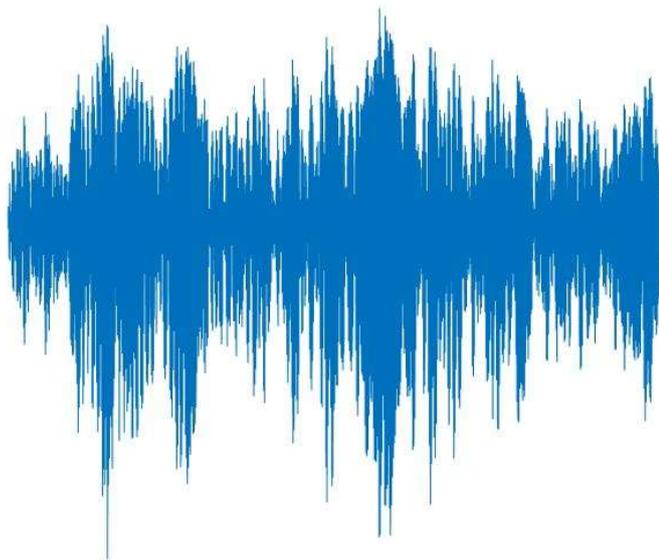
I9: Training progress for 3:2 ratio on training and test set

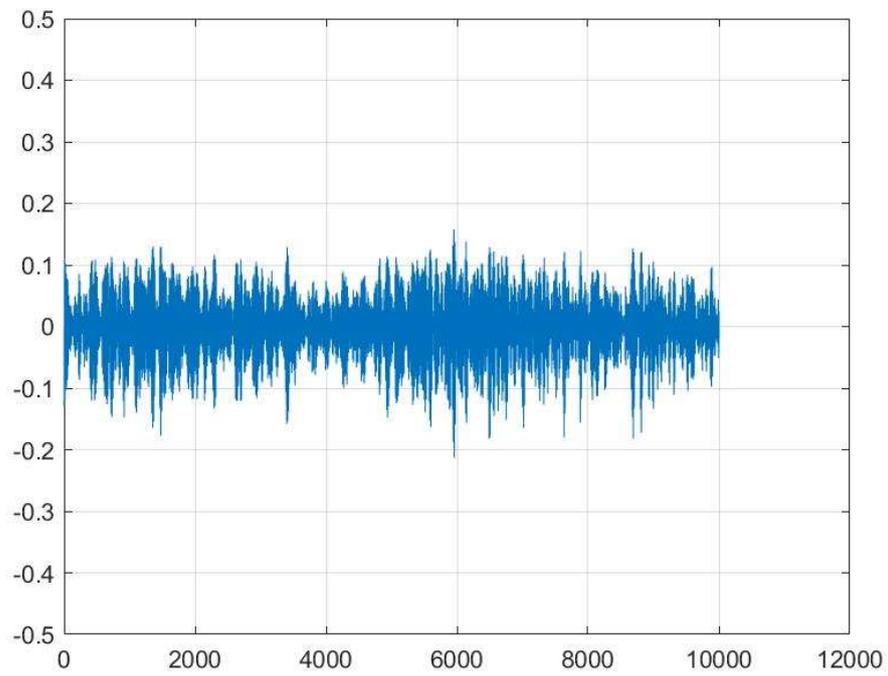
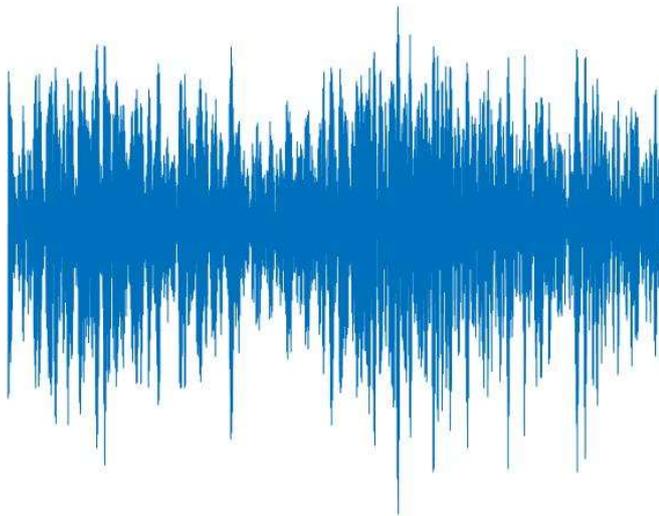


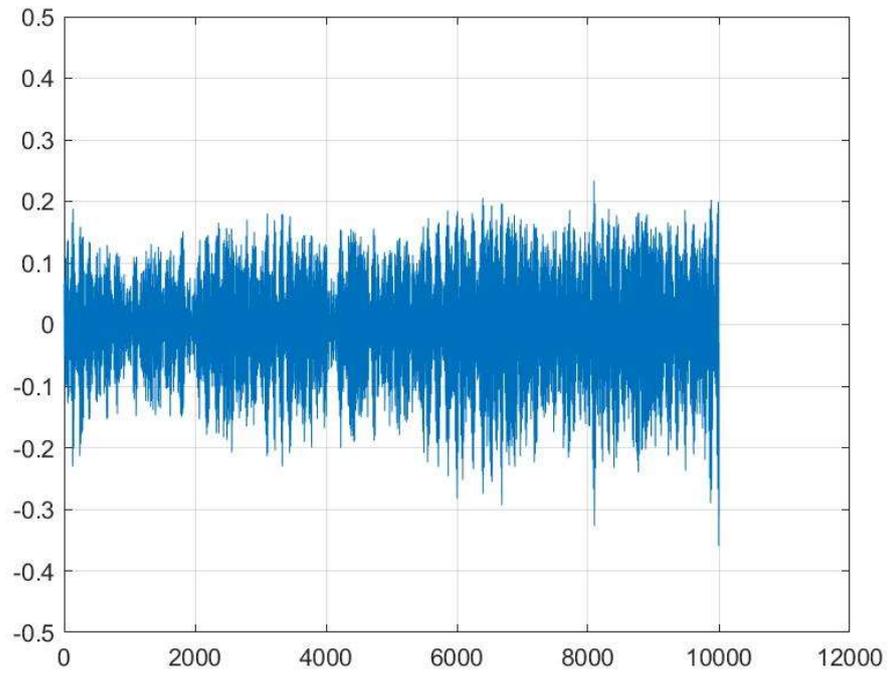
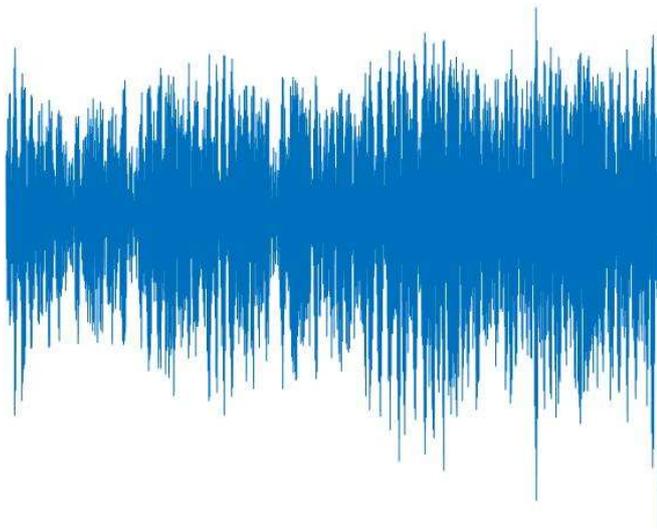
I10: Confusion matrix for 3:2 ratio on training and test set

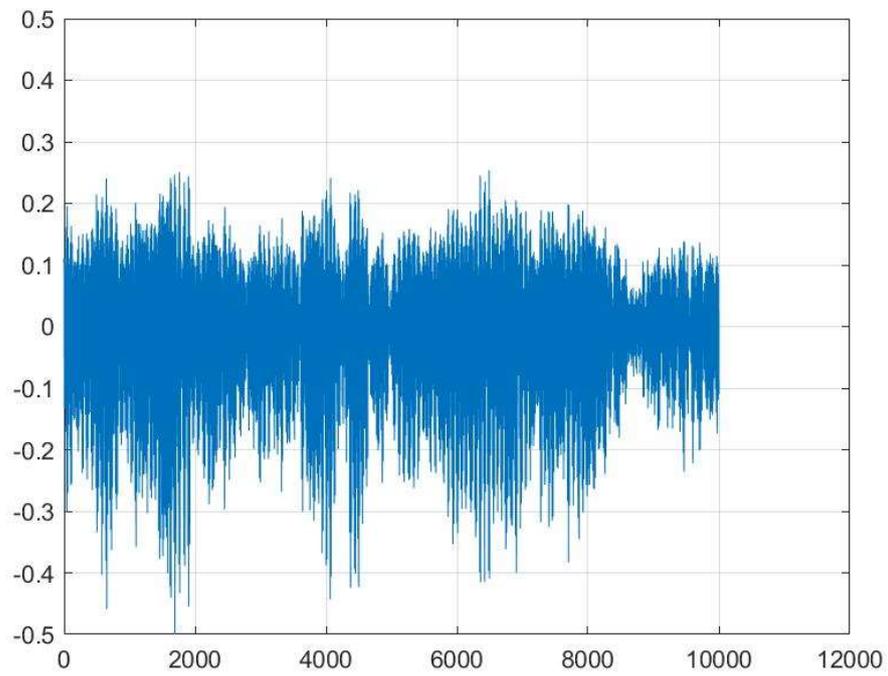
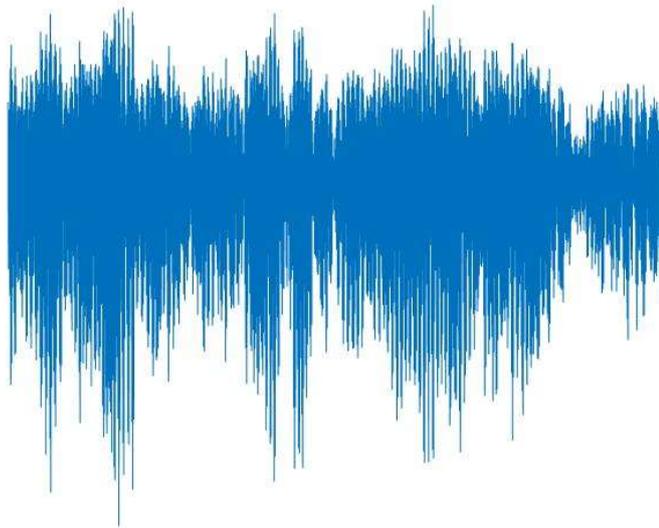


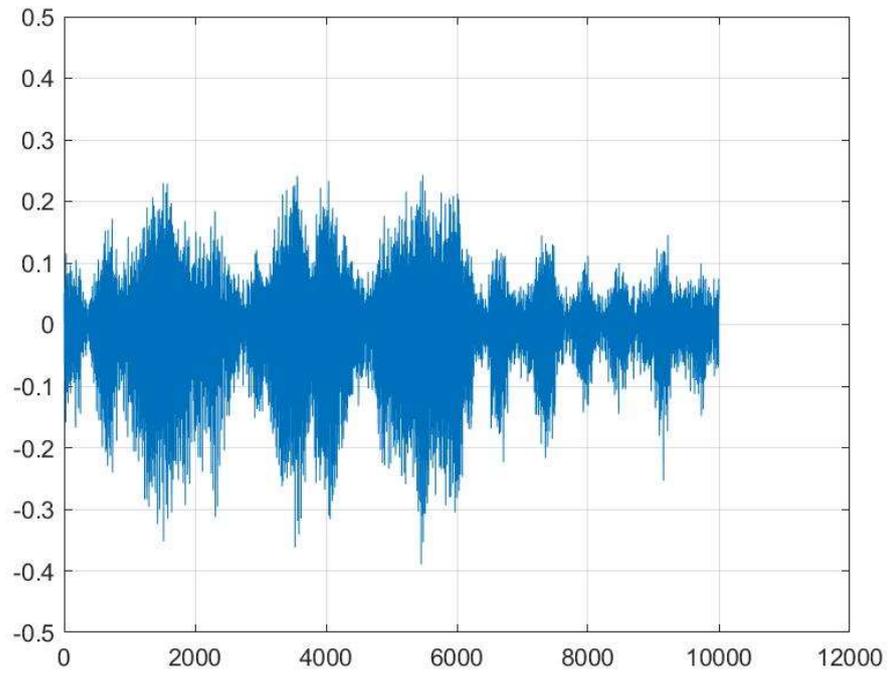
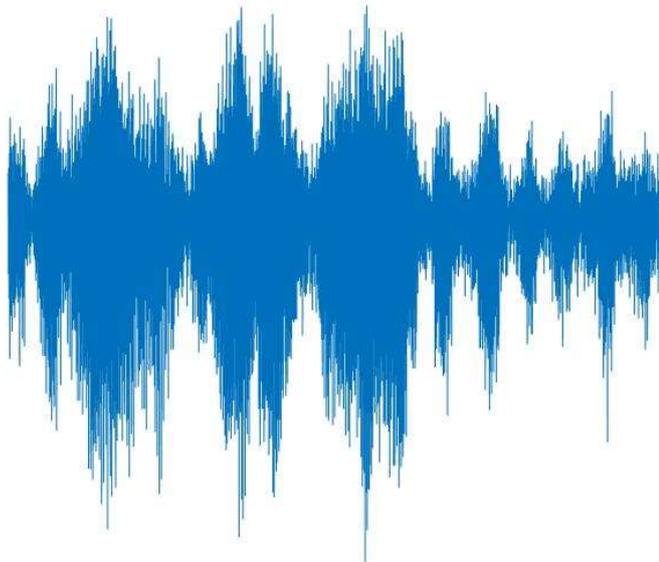
Appendix J: Sample data for DCNN speed classification for baseline test**J1A: 1000 mm/min with fixed grid and axis****J1B: 1000 mm/min without grid and axis**

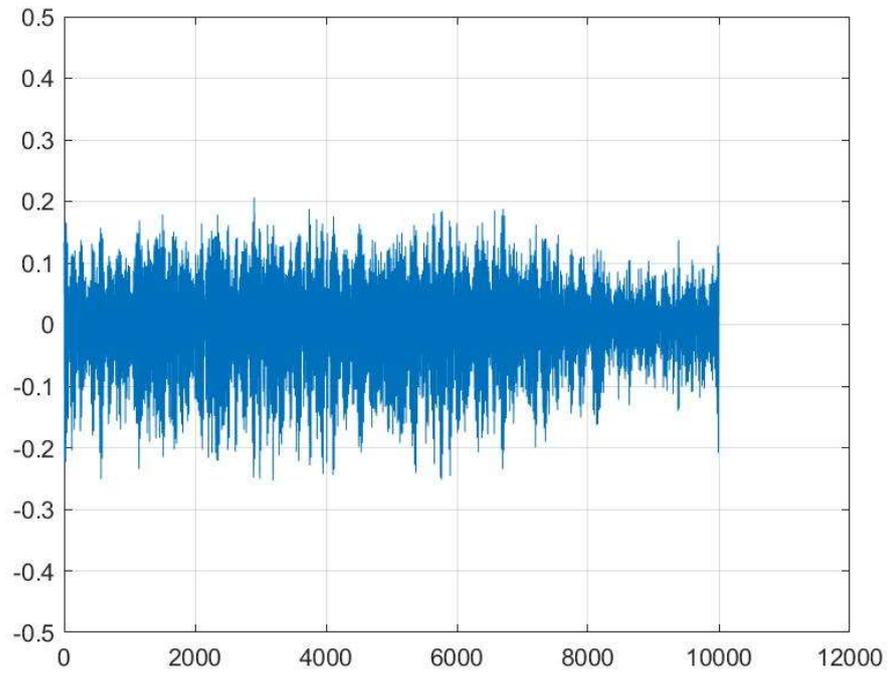
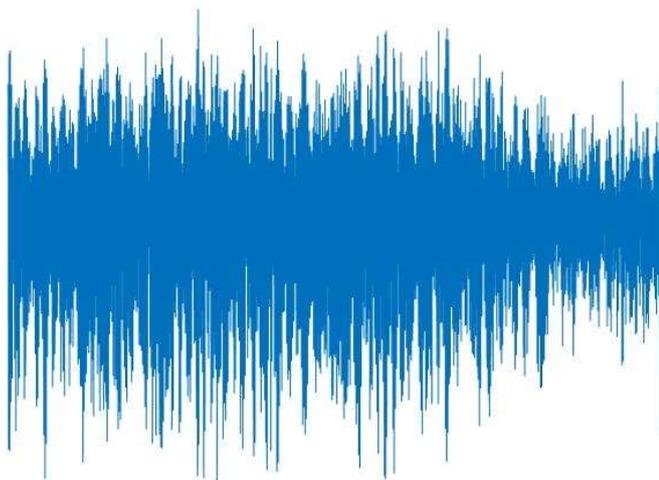
J2A: 2500 mm/min with fixed grid and axis**J2B: 2500 mm/min without grid and axis**

J3A: 3000 mm/min with fixed grid and axis**J3B: 3000 mm/min without grid and axis**

J4A: 5000 mm/min with fixed grid and axis**J4B: 5000 mm/min without grid and axis**

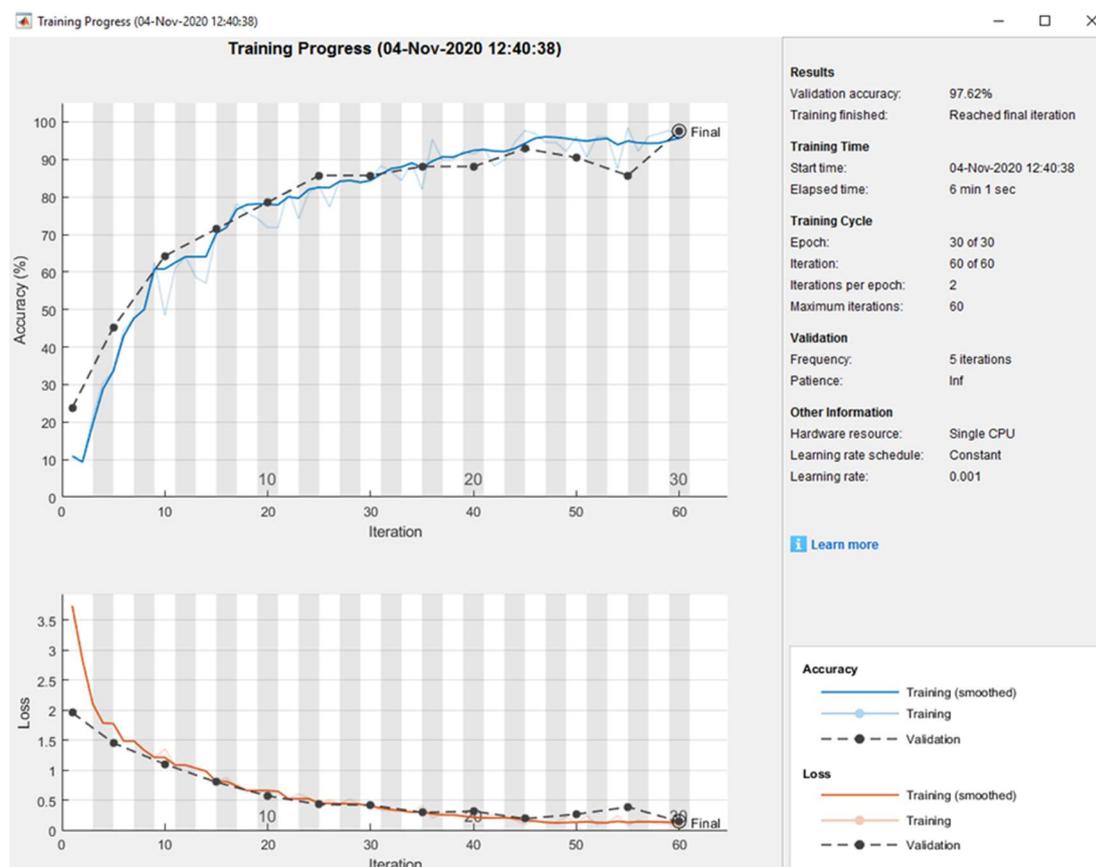
J5A: 7500 mm/min with fixed grid and axis**J5B: 7500 mm/min without grid and axis**

J6A: 9000 mm/min with fixed grids and axis**J6B: 9000 mm/min without grids and axis**

J7A: 10000 mm/min with fixed grids and axis**J7B: 10000 mm/min without grids and axis**

Appendix K: DCNN results on speed classification on baseline test

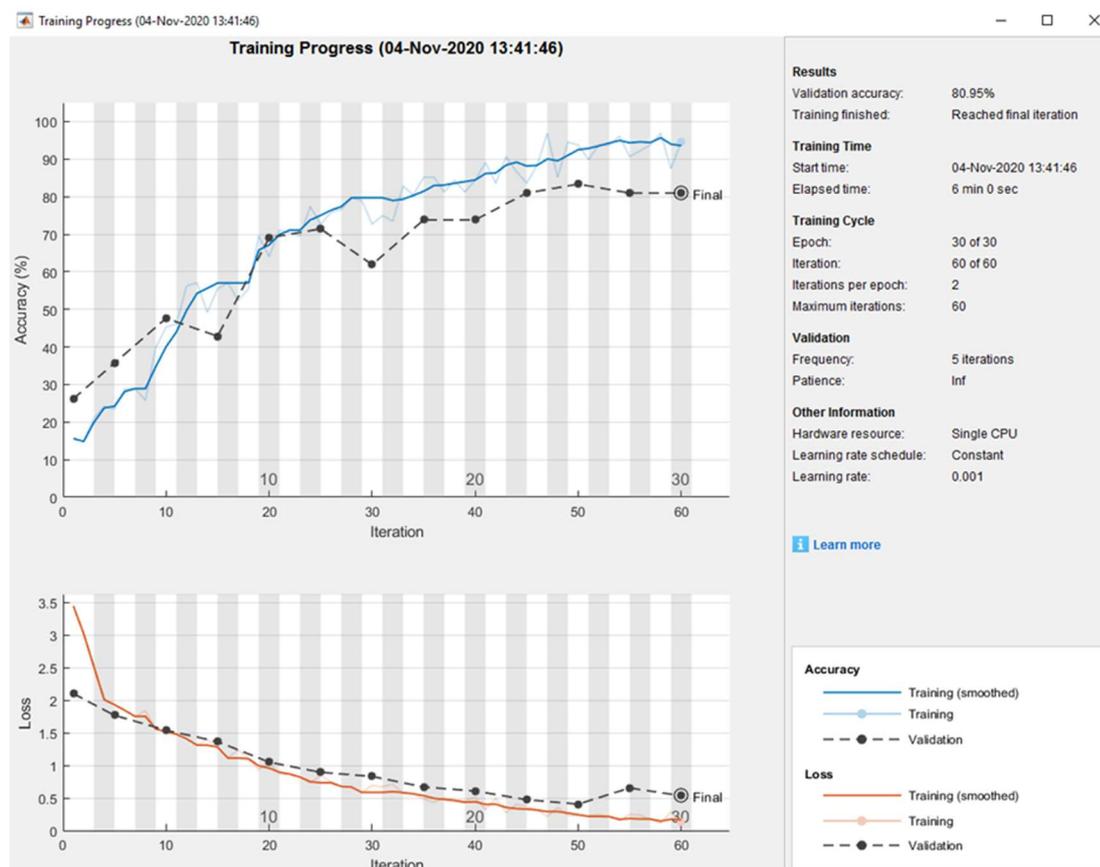
K1A: Training progress for 9:1 ratio on training and test data (with grid/axis)



K1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis)

10k	6	0	0	0	0	0	0
1k	0	6	0	0	0	0	0
2k5	1	0	5	0	0	0	0
3k	0	0	0	6	0	0	0
5k	0	0	0	0	6	0	0
7k5	0	0	0	0	0	6	0
9k	0	0	0	0	0	0	6
	10k	1k	2k5	3k	5k	7k5	9k

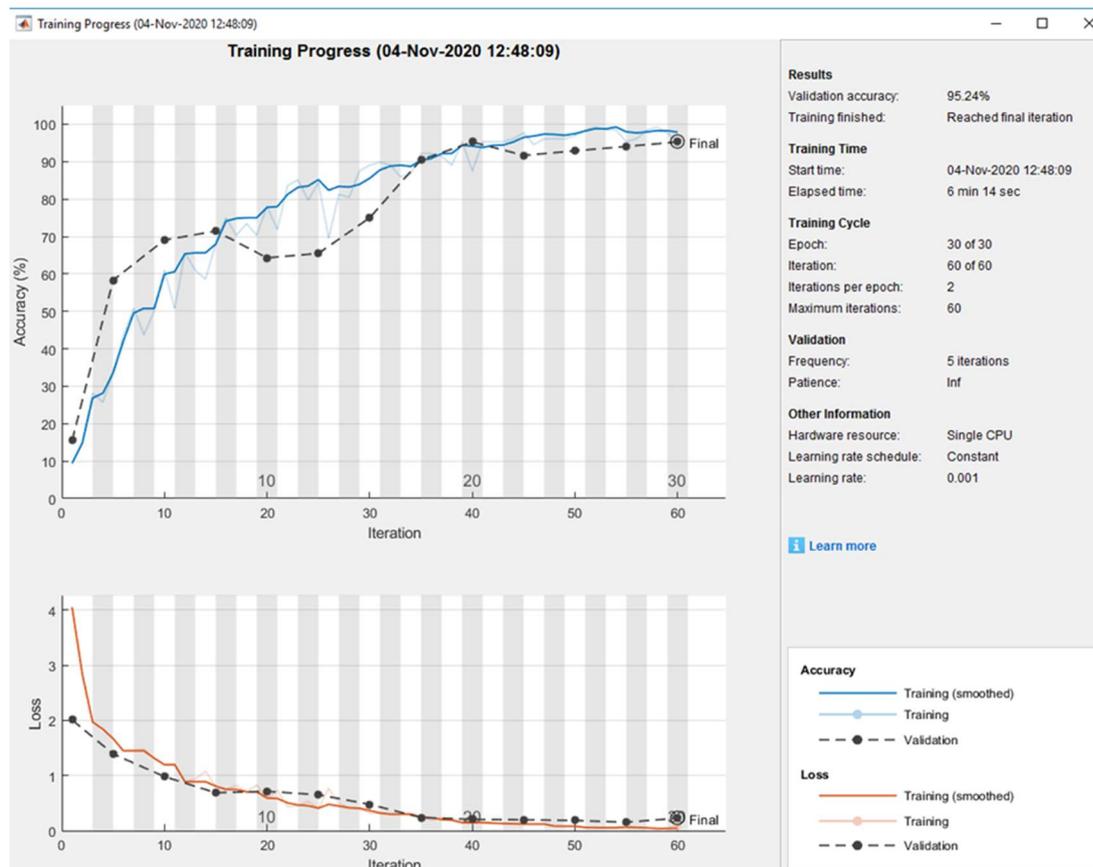
K1C: Training progress for 9:1 ratio on training and test data (no grid/axis)



K1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis)

10k	4	1	0	1	0	0	0
1k	0	6	0	0	0	0	0
2k5	0	3	3	0	0	0	0
3k	0	0	0	6	0	0	0
5k	0	0	1	0	5	0	0
7k5	0	0	0	0	0	6	0
9k	0	0	1	1	0	0	4
	10k	1k	2k5	3k	5k	7k5	9k

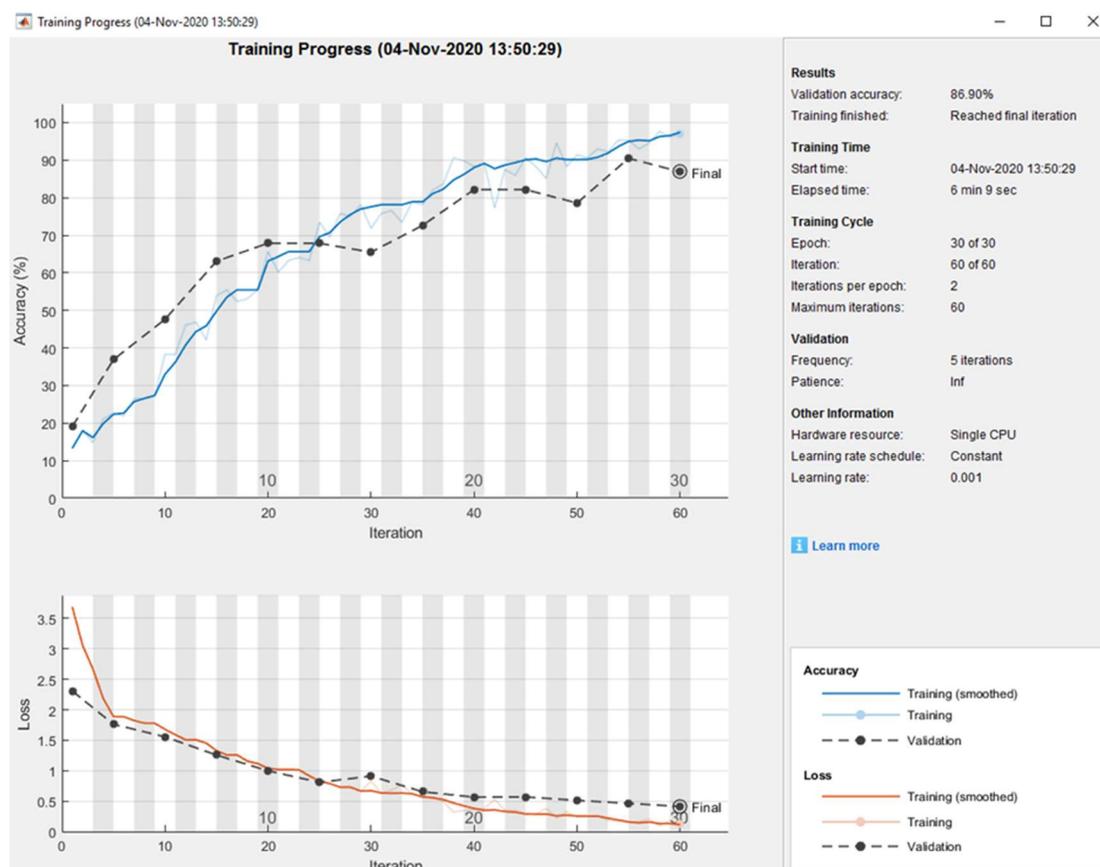
K2A: Training progress for 4:1 ratio on training and test data (with grid/axis)



K2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis)

10k	12	0	0	0	0	0	0
1k	2	10	0	0	0	0	0
2k5	0	0	11	1	0	0	0
3k	0	0	0	12	0	0	0
5k	0	0	0	0	12	0	0
7k5	0	0	0	0	0	12	0
9k	0	0	0	1	0	0	11
	10k	1k	2k5	3k	5k	7k5	9k

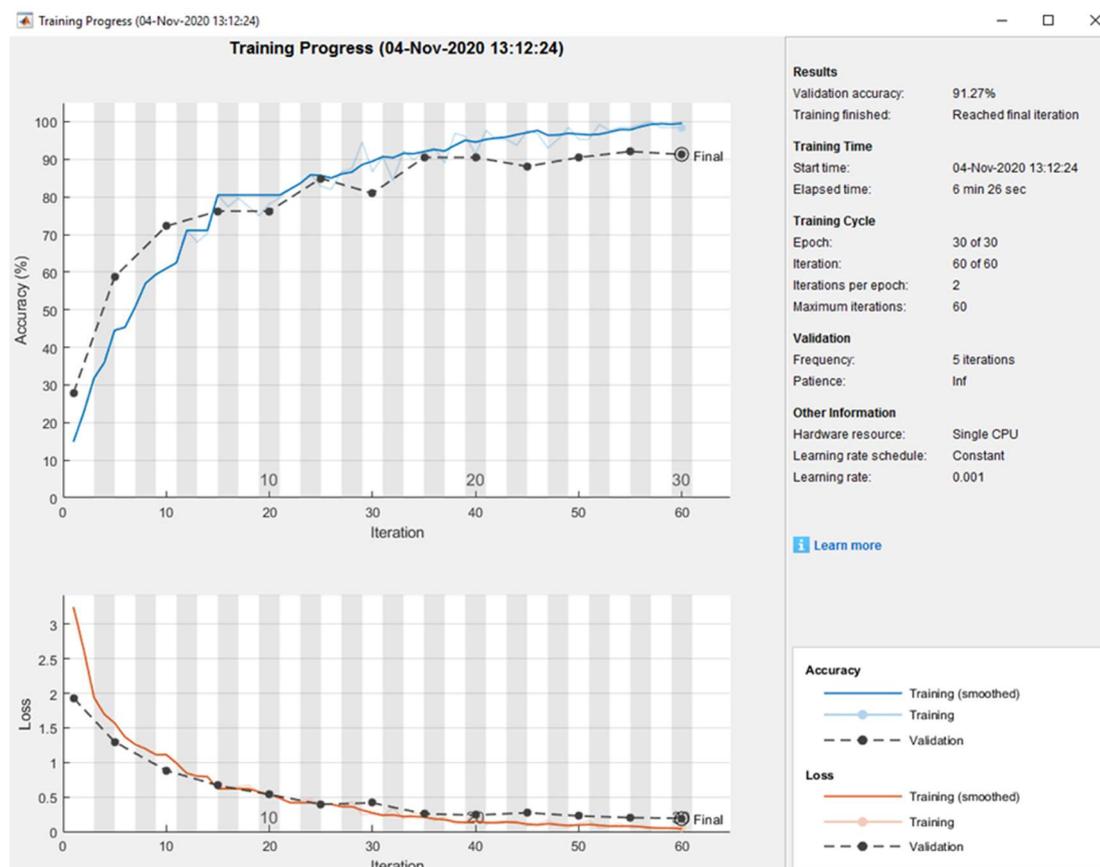
K2C: Training progress for 4:1 ratio on training and test data (no grid/axis)



K2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis)

10k	10	0	2	0	0	0	0
1k	0	12	0	0	0	0	0
2k5	0	1	7	0	3	0	1
3k	2	0	0	8	1	0	1
5k	0	0	0	0	12	0	0
7k5	0	0	0	0	0	12	0
9k	0	0	0	0	0	0	12
	10k	1k	2k5	3k	5k	7k5	9k

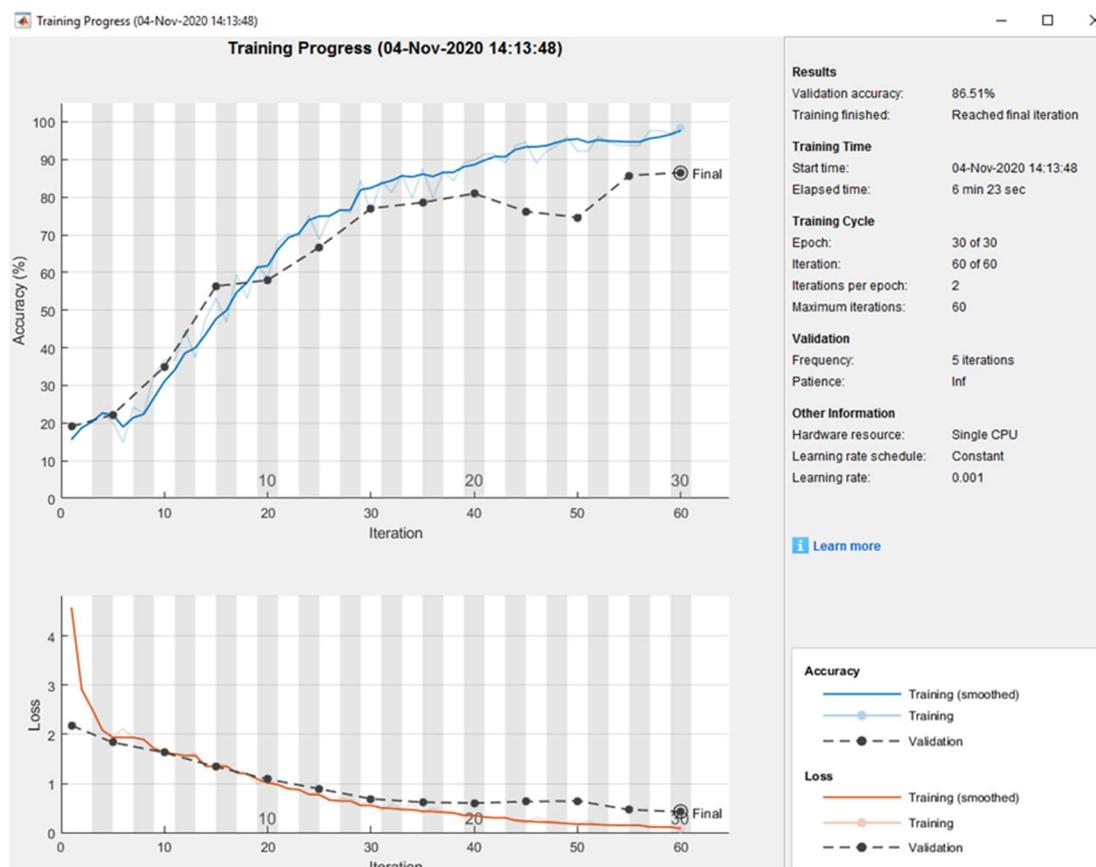
K3A: Training progress for 7:3 ratio on training and test data (with grid/axis)



K3B: Confusion matrix for 7:3 ratio on training and test data (with grid/axis)

10k	18	0	0	0	0	0	0
1k	3	13	2	0	0	0	0
2k5	0	1	17	0	0	0	0
3k	1	0	0	15	0	0	2
5k	1	0	0	0	17	0	0
7k5	0	0	0	0	0	18	0
9k	1	0	0	0	0	0	17
	10k	1k	2k5	3k	5k	7k5	9k

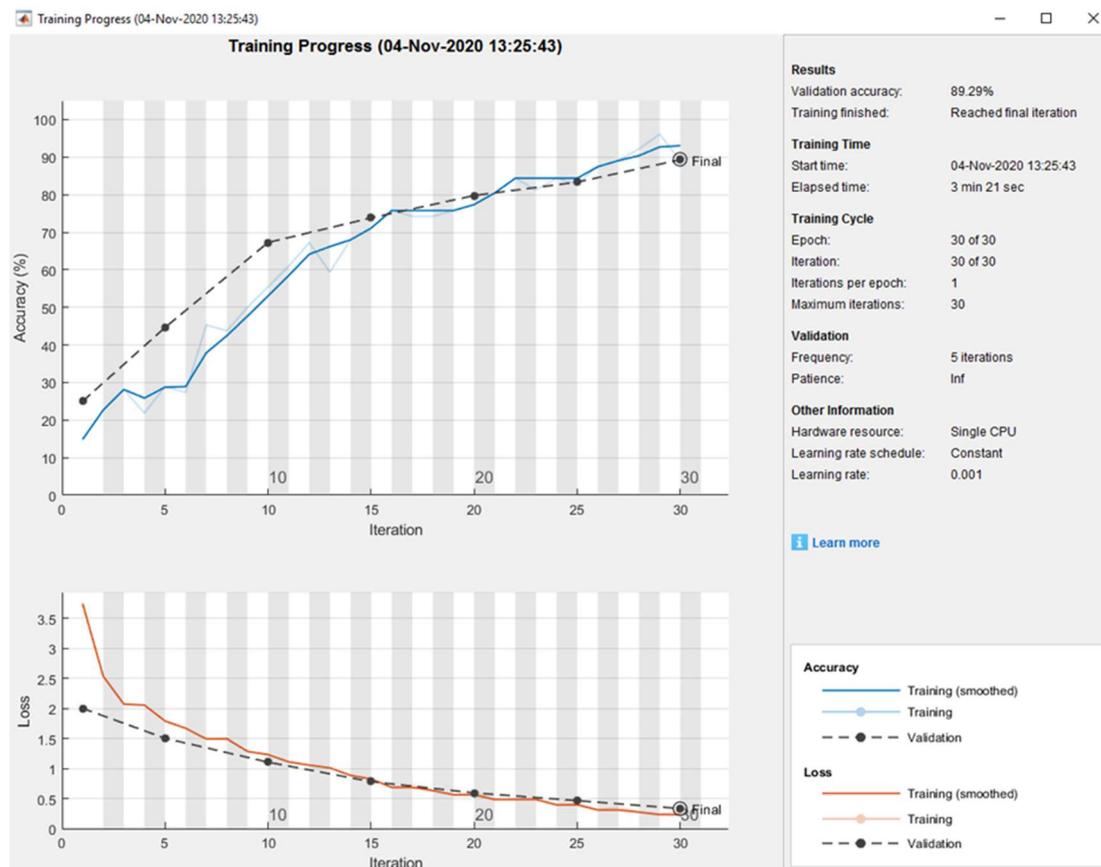
K3C: Training progress for 7:3 ratio on training and test data (no grid/axis)



K3D: Confusion matrix for 7:3 ratio on training and test data (no grid/axis)

10k	13	0	2	0	1	1	1
1k	0	15	2	1	0	0	0
2k5	0	0	14	1	2	0	1
3k	0	0	2	16	0	0	0
5k	0	0	0	0	18	0	0
7k5	0	0	0	0	0	18	0
9k	1	0	2	0	0	0	15
	10k	1k	2k5	3k	5k	7k5	9k

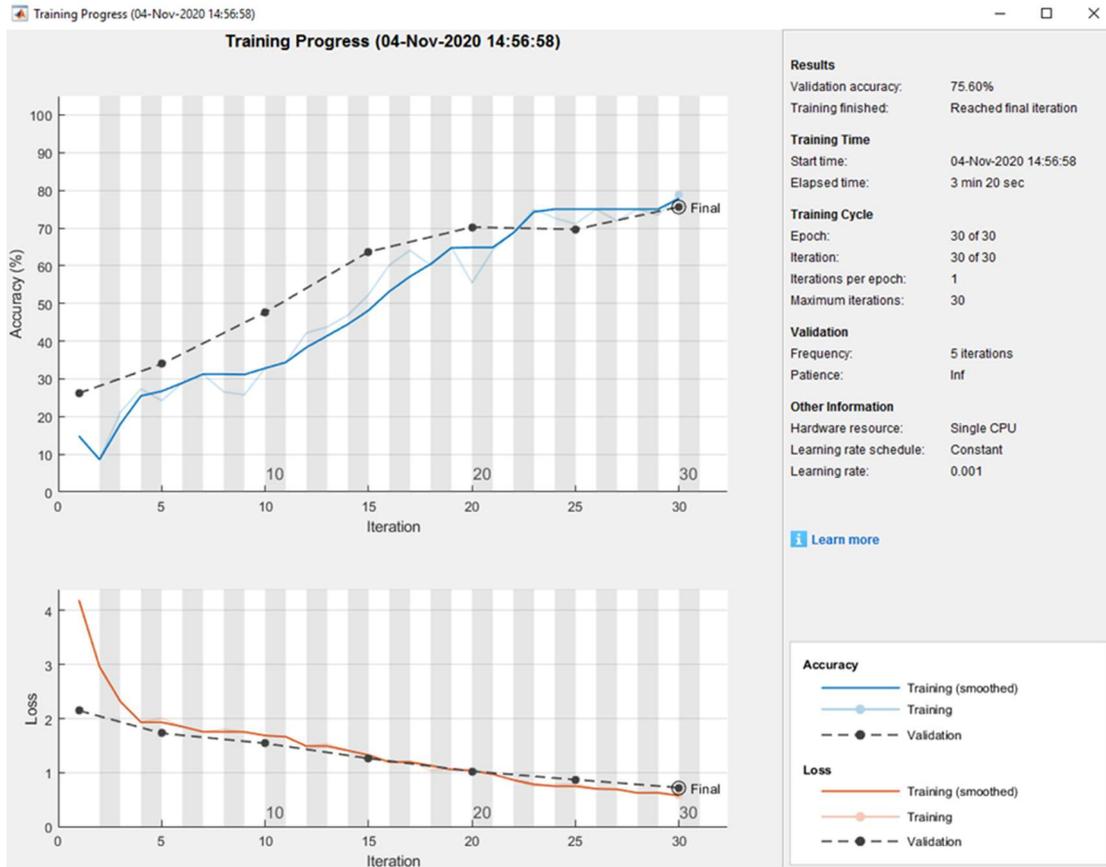
K4A: Training progress for 3:2 ratio on training and test data (with grid/axis)



K4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis)

10k	21	0	2	0	1	0	0
1k	0	19	5	0	0	0	0
2k5	0	3	17	0	1	1	2
3k	0	0	0	24	0	0	0
5k	0	0	0	0	24	0	0
7k5	0	0	0	0	0	24	0
9k	1	0	0	2	0	0	21
	10k	1k	2k5	3k	5k	7k5	9k

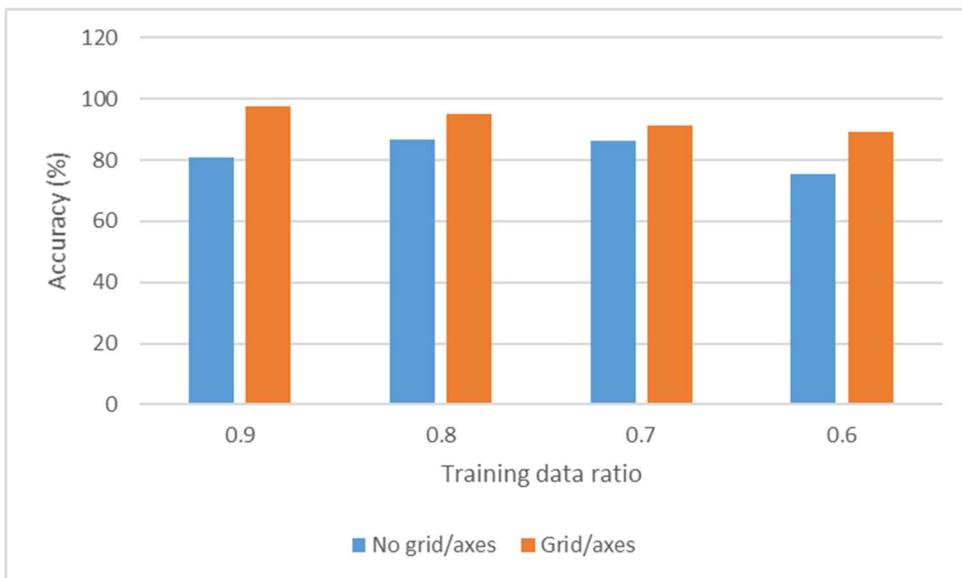
K4C: Training progress for 3:2 ratio on training and test data (no grid/axis)



K4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis)

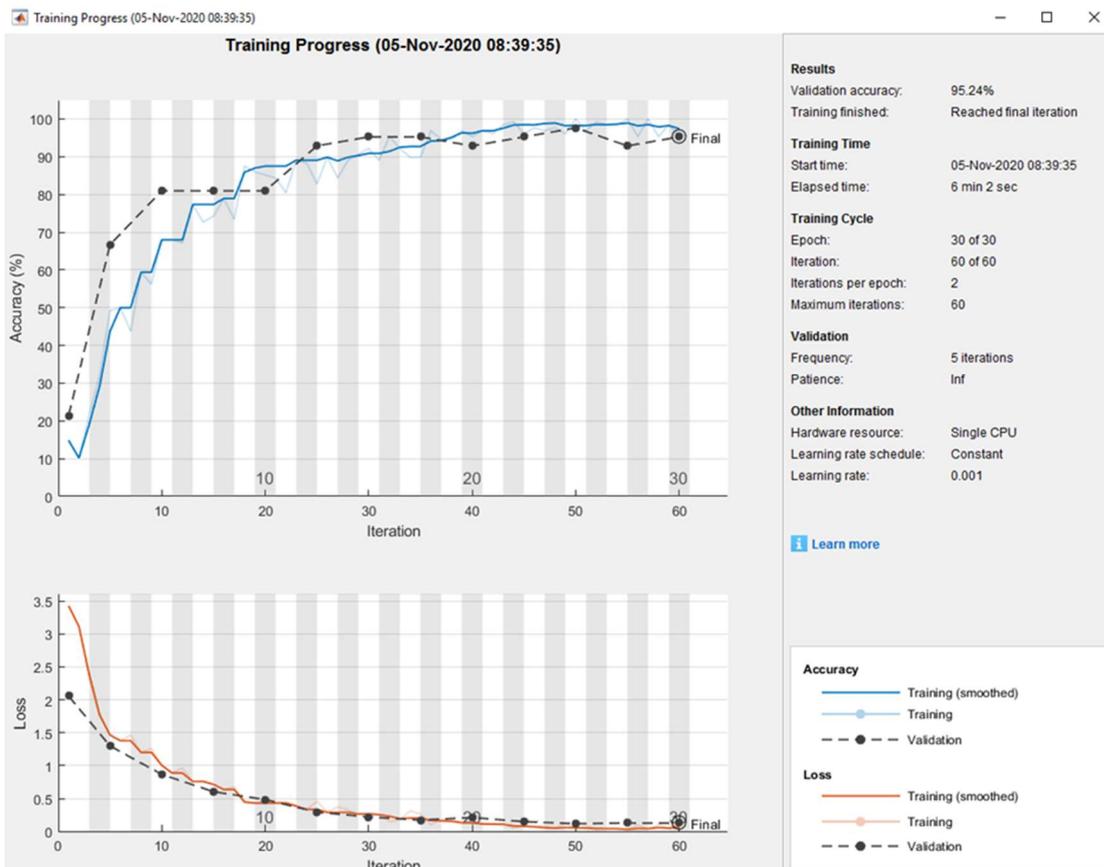
10k	21	0	1	1	0	1	0
1k	0	22	1	0	0	0	1
2k5	6	4	7	3	3	0	1
3k	7	1	4	10	1	0	1
5k	0	0	0	0	24	0	0
7k5	0	0	0	0	0	24	0
9k	2	0	0	0	3	0	19
	10k	1k	2k5	3k	5k	7k5	9k

K5: Comparing results between grid/axis and no grid/axis.



Appendix L: DCNN results on speed classification on misaligned sensor axis test

L1A: Training progress for 9:1 ratio on training and test data (with grid/axis)



L1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis)

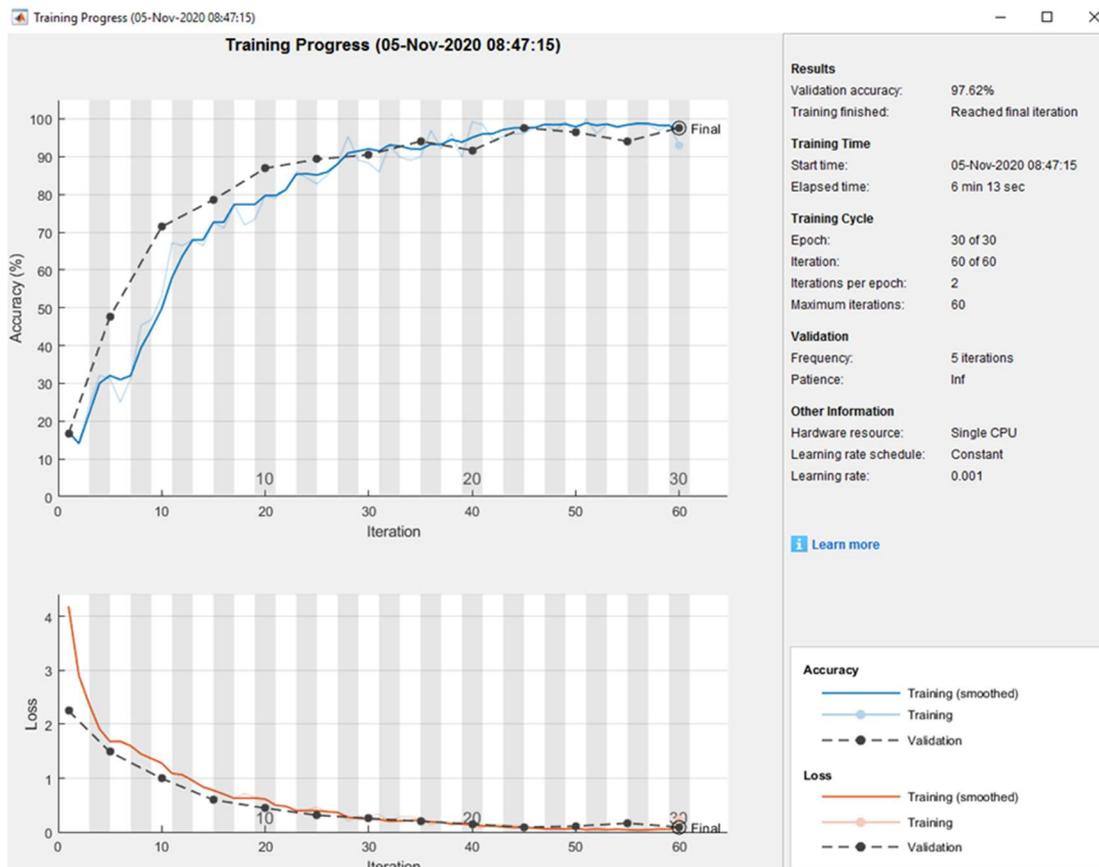
10k	6	0	0	0	0	0	0
1k	0	6	0	0	0	0	0
2k5	0	1	4	0	1	0	0
3k	0	0	0	6	0	0	0
5k	0	0	0	0	6	0	0
7k5	0	0	0	0	0	6	0
9k	0	0	0	0	0	0	6
	10k	1k	2k5	3k	5k	7k5	9k

L1C: Training progress for 9:1 ratio on training and test data (no grid/axis)

L1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis)

10k	6	0	0	0	0	0	0
1k	1	5	0	0	0	0	0
2k5	0	2	1	1	1	0	1
3k	1	2	0	3	0	0	0
5k	0	0	0	0	6	0	0
7k5	0	0	0	0	0	6	0
9k	1	0	0	0	0	0	5
	10k	1k	2k5	3k	5k	7k5	9k

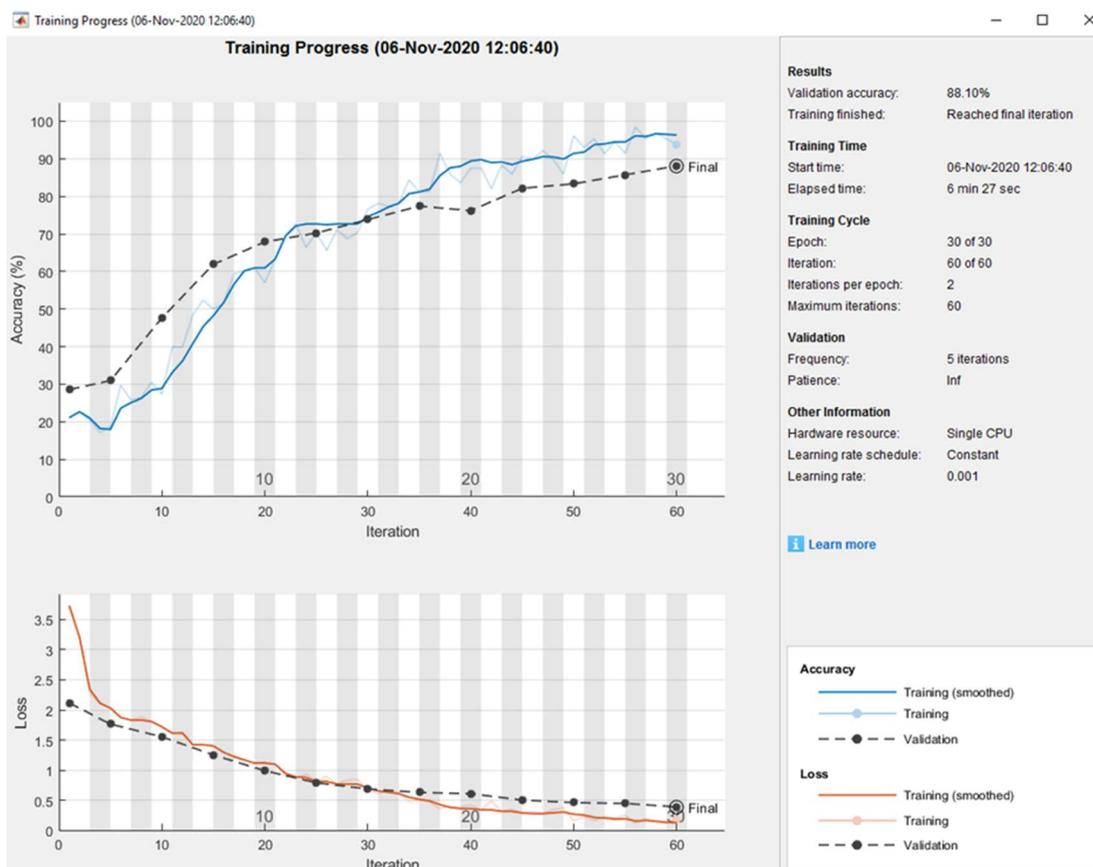
L2A: Training progress for 4:1 ratio on training and test data (with grid/axis)



L2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis)

10k	12	0	0	0	0	0	0
1k	0	12	0	0	0	0	0
2k5	0	0	10	0	1	0	1
3k	0	0	0	12	0	0	0
5k	0	0	0	0	12	0	0
7k5	0	0	0	0	0	12	0
9k	0	0	0	0	0	0	12
	10k	1k	2k5	3k	5k	7k5	9k

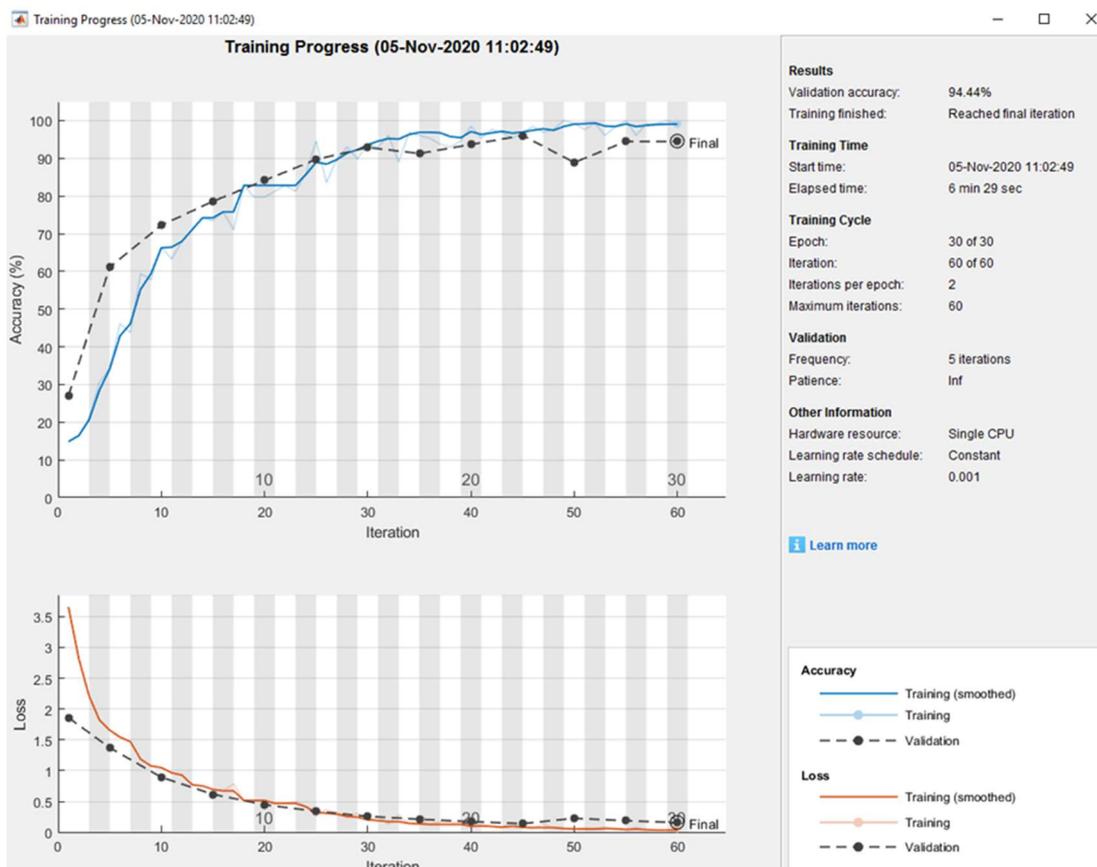
L2C: Training progress for 4:1 ratio on training and test data (no grid/axis)



L2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis)

10k	10	0	0	2	0	0	0
1k	0	12	0	0	0	0	0
2k5	0	0	8	2	1	0	1
3k	1	1	0	10	0	0	0
5k	0	0	0	0	12	0	0
7k5	0	0	0	0	0	12	0
9k	0	0	1	0	1	0	10
	10k	1k	2k5	3k	5k	7k5	9k

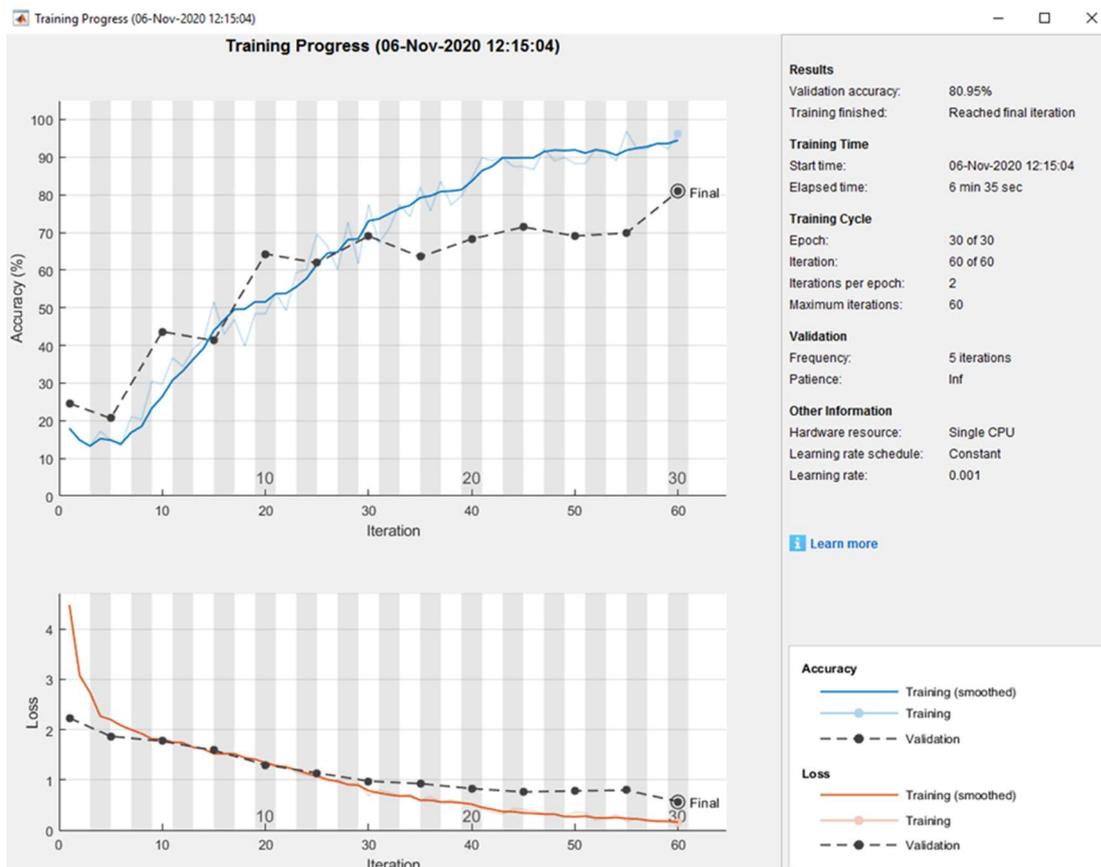
L3A: Training progress for 7:3 ratio on training and test data (with grid/axis)



L3B: Confusion matrix for 7:3 ratio on training and test data (with grid/axis)

10k	18	0	0	0	0	0	0
1k	1	17	0	0	0	0	0
2k5	0	0	16	2	0	0	0
3k	0	0	0	18	0	0	0
5k	0	0	0	0	18	0	0
7k5	0	0	0	0	0	18	0
9k	1	0	2	1	0	0	14
	10k	1k	2k5	3k	5k	7k5	9k

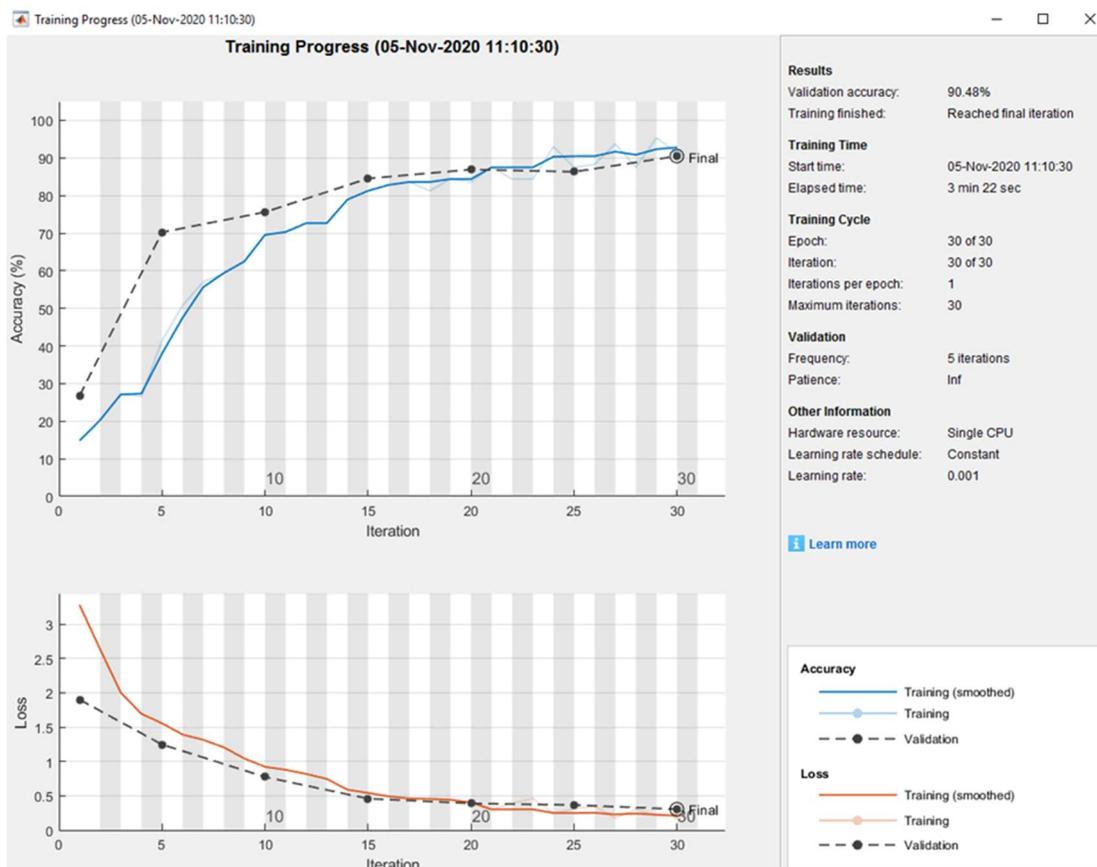
L3C: Training progress for 7:3 ratio on training and test data (no grid/axis)



L3D: Confusion matrix for 7:3 ratio on training and test data (no grid/axis)

10k	11	2	1	2	0	0	2
1k	0	16	2	0	0	0	0
2k5	0	1	10	2	2	0	3
3k	2	0	1	14	0	0	1
5k	0	0	0	0	18	0	0
7k5	0	0	0	0	0	18	0
9k	3	0	0	0	0	0	15
	10k	1k	2k5	3k	5k	7k5	9k

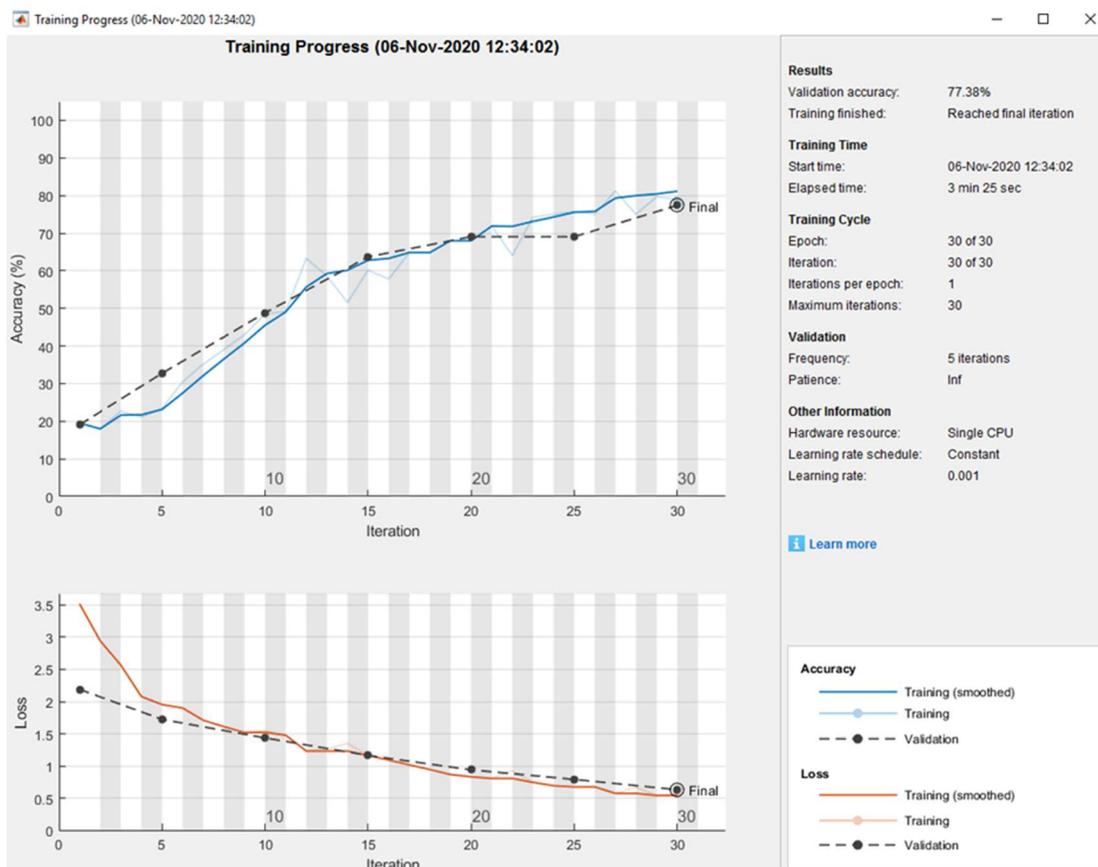
L4A: Training progress for 3:2 ratio on training and test data (with grid/axis)



L4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis)

10k	23	0	1	0	0	0	0
1k	0	24	0	0	0	0	0
2k5	0	1	18	1	0	0	4
3k	0	0	2	15	0	0	7
5k	0	0	0	0	24	0	0
7k5	0	0	0	0	0	24	0
9k	0	0	0	0	0	0	24
	10k	1k	2k5	3k	5k	7k5	9k

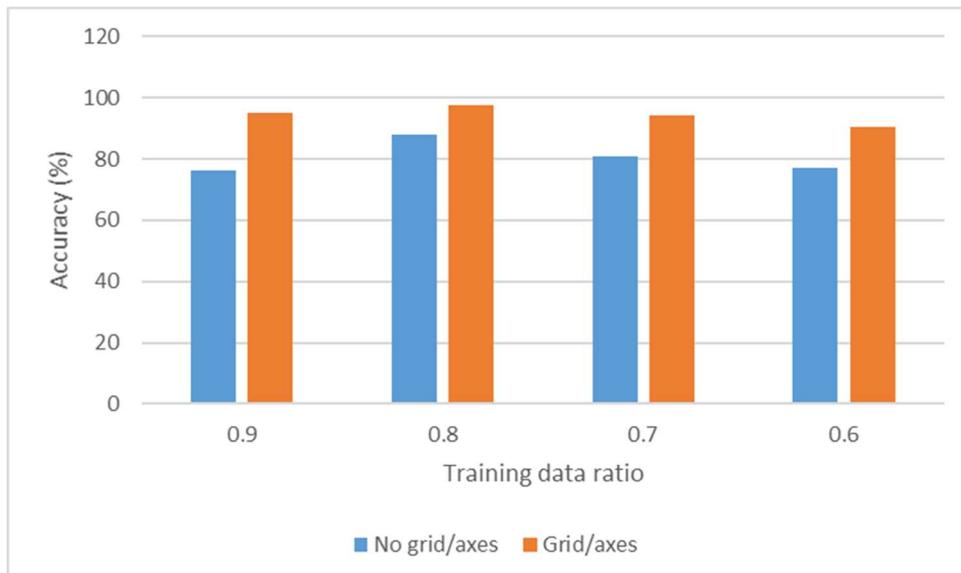
L4C: Training progress for 3:2 ratio on training and test data (no grid/axis)



L4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis)

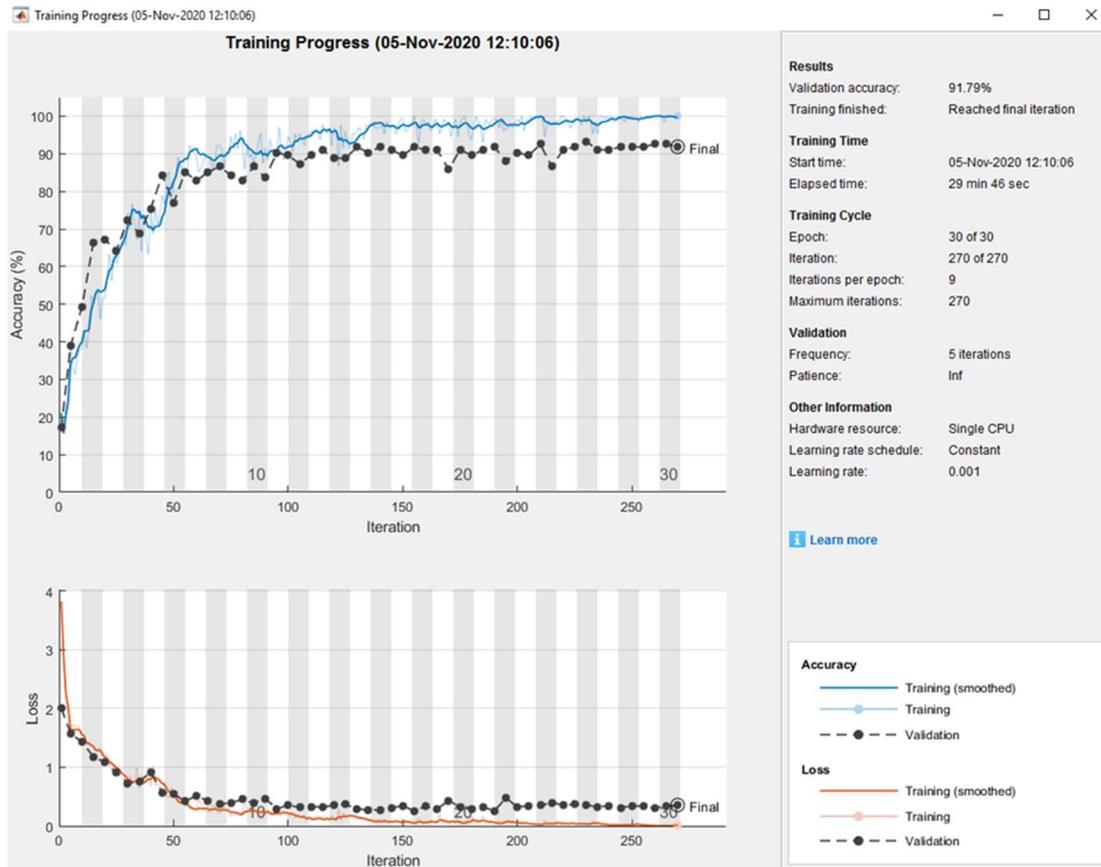
10k	12	0	6	4	1	0	1
1k	0	23	1	0	0	0	0
2k5	0	2	17	1	1	0	3
3k	0	3	7	12	1	0	1
5k	0	0	2	0	22	0	0
7k5	0	0	0	0	0	24	0
9k	0	0	4	0	0	0	20
	10k	1k	2k5	3k	5k	7k5	9k

L5: Comparing results between grids/axis and no grid/axis



Appendix M: DCNN results on speed classification on baseline test database

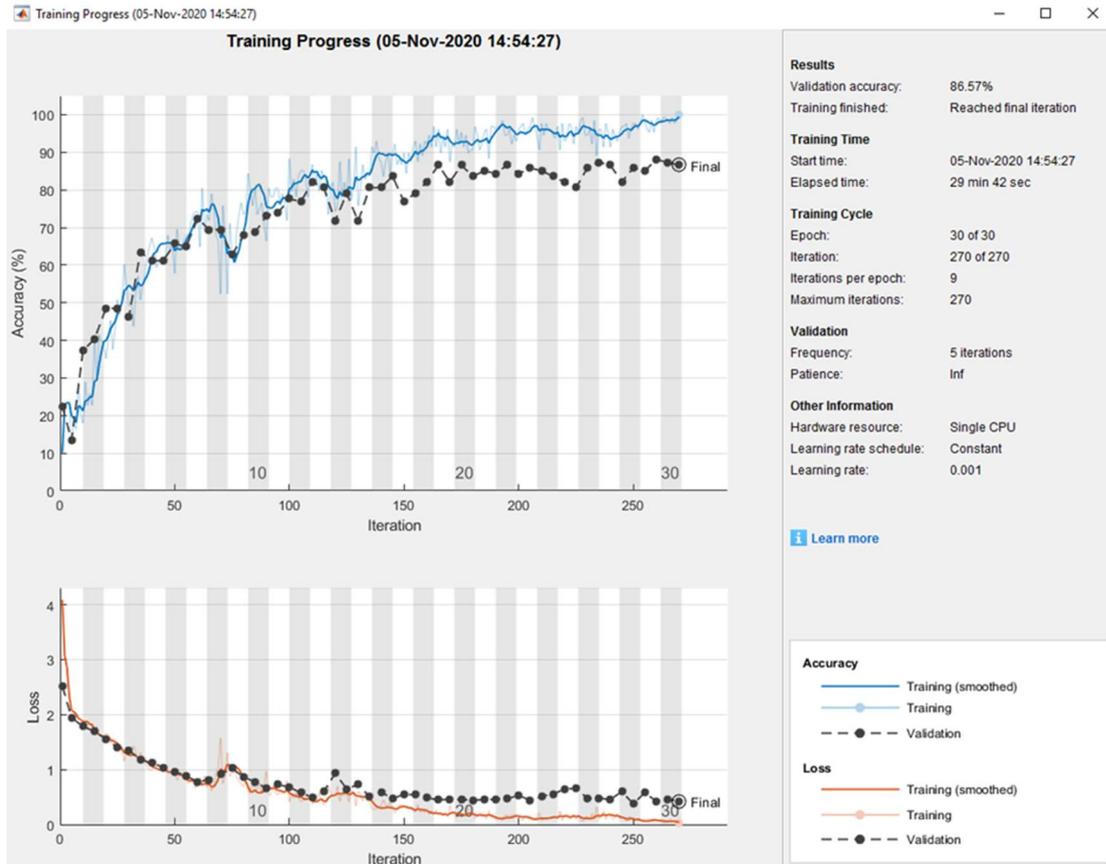
M1A: Training progress for 9:1 ratio on training and test data (with grid/axis)



M1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis)

10k	18	0	0	0	0	0	0
1k	0	12	0	4	2	0	0
2k5	0	1	17	2	0	0	0
3k	0	0	0	17	0	1	0
5k	0	0	0	0	20	0	0
7k5	0	0	0	0	0	20	0
9k	0	0	0	1	0	0	19
	10k	1k	2k5	3k	5k	7k5	9k

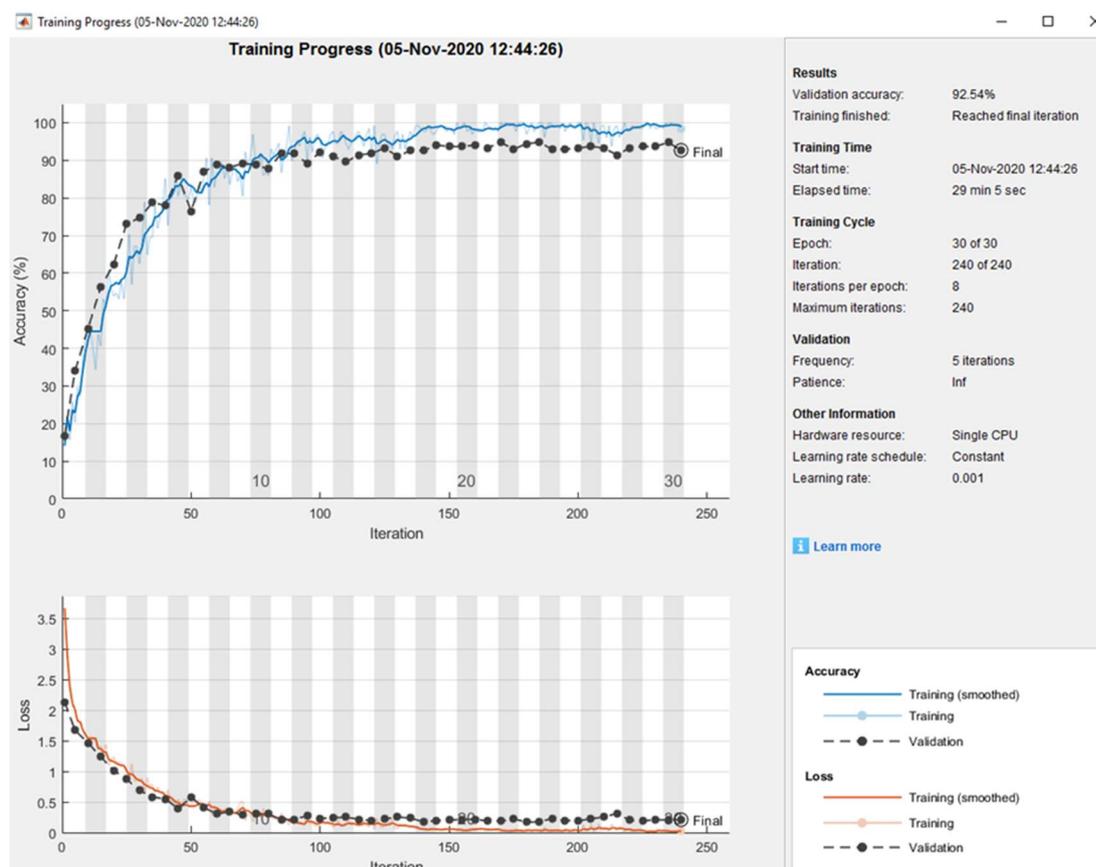
M1C: Training progress for 9:1 ratio on training and test data (no grid/axis)



M1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis)

10k	18	0	0	0	0	0	0
1k	0	17	0	1	0	0	0
2k5	1	0	17	0	2	0	0
3k	2	2	1	12	0	0	1
5k	4	0	0	1	15	0	0
7k5	0	0	0	0	1	19	0
9k	0	0	0	0	2	0	18
	10k	1k	2k5	3k	5k	7k5	9k

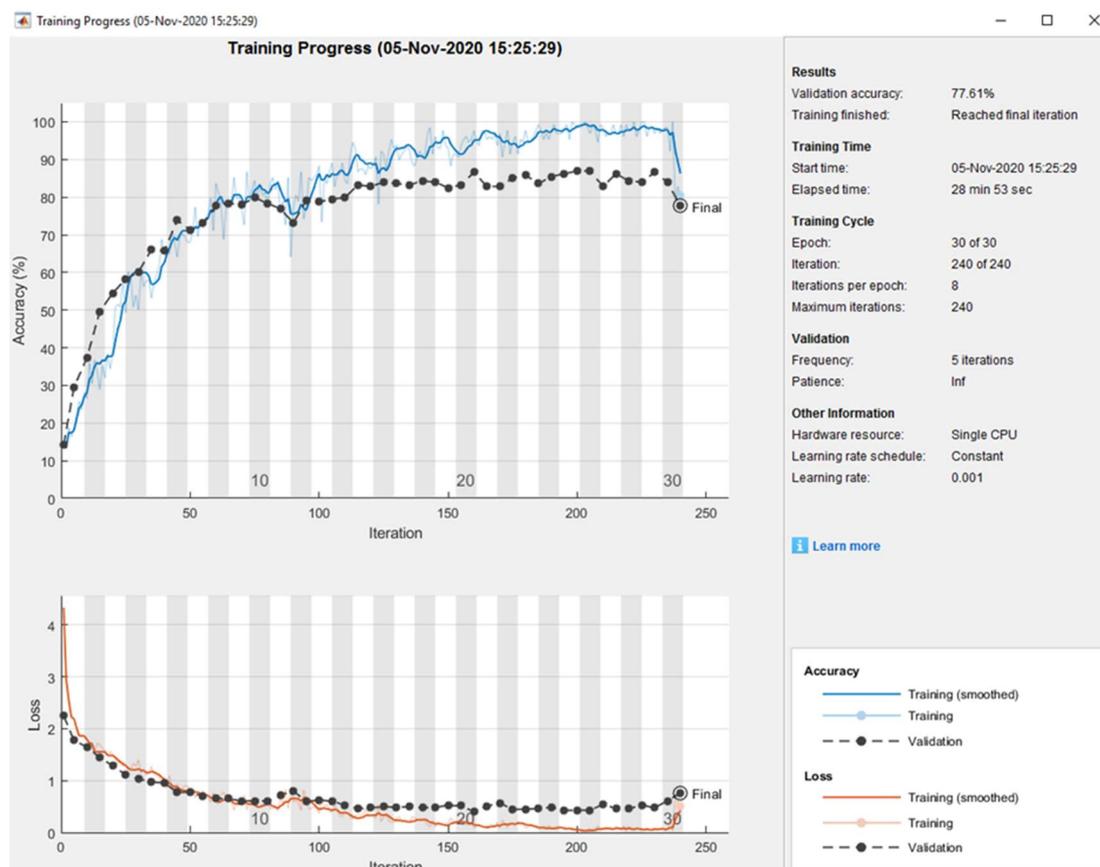
M2A: Training progress for 4:1 ratio on training and test data (with grid/axis)



M2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis)

10k	36	0	0	0	0	0	0
1k	0	31	0	4	1	0	0
2k5	0	2	34	3	0	1	0
3k	2	3	0	28	2	0	1
5k	0	0	0	0	39	1	0
7k5	0	0	0	0	0	40	0
9k	0	0	0	0	0	0	40
	10k	1k	2k5	3k	5k	7k5	9k

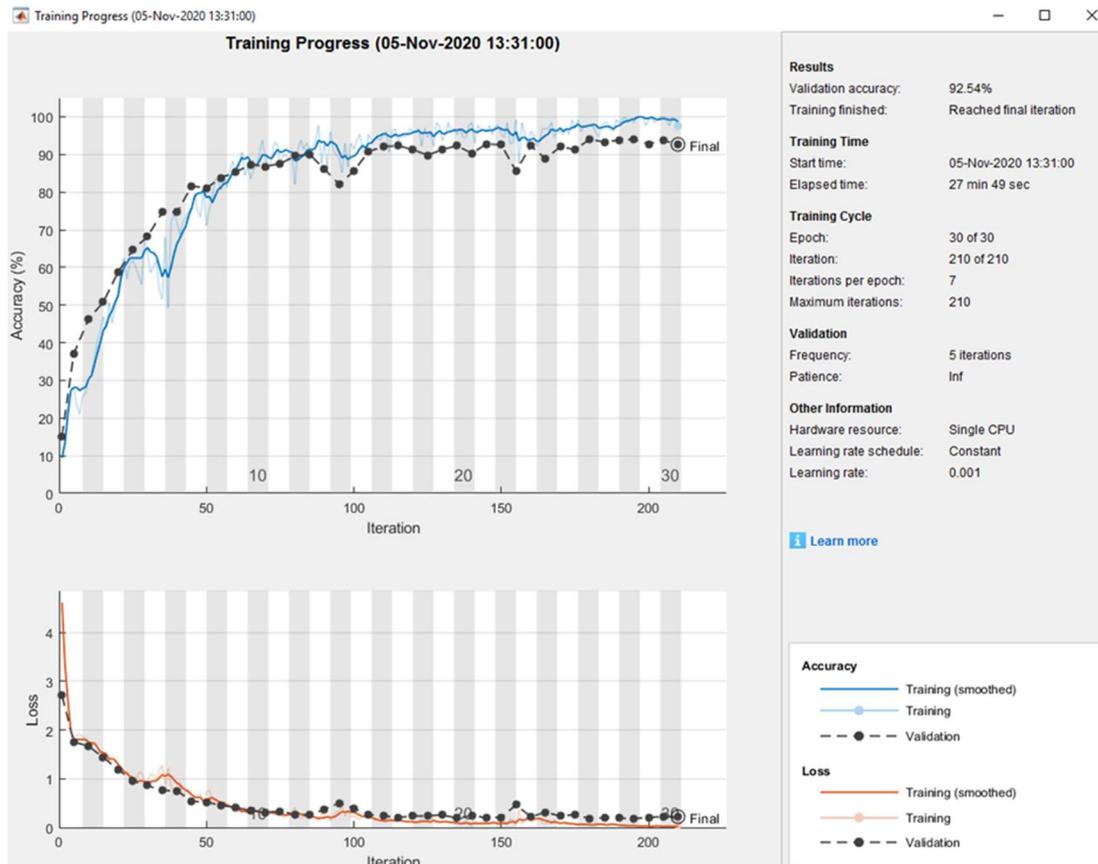
M2C: Training progress for 4:1 ratio on training and test data (no grid/axis)



M2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis)

10k	32	0	1	0	3	0	0
1k	1	14	9	0	8	2	2
2k5	0	0	36	0	4	0	0
3k	0	2	5	23	4	2	0
5k	2	0	7	0	28	3	0
7k5	0	0	0	0	0	40	0
9k	1	0	0	0	3	1	35
	10k	1k	2k5	3k	5k	7k5	9k

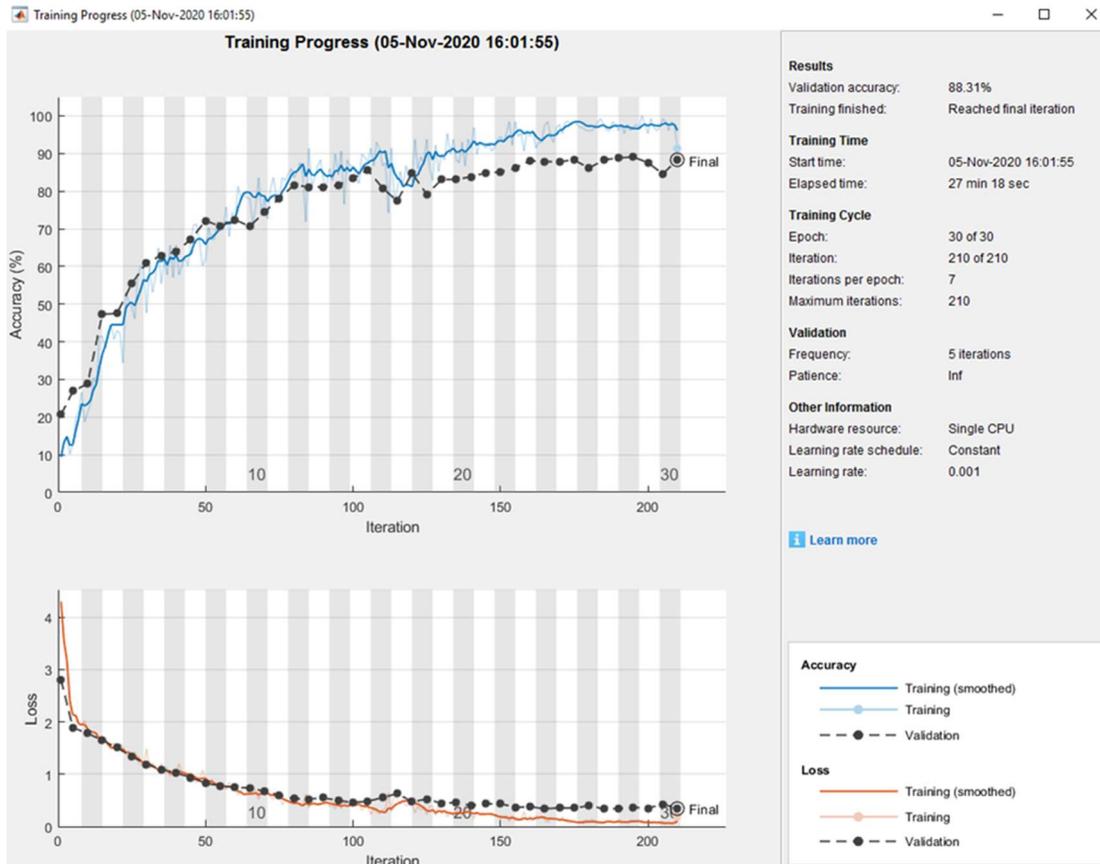
M3A: Training progress for 7:3 ratio on training and test data (with grid/axis)



M3B: Confusion matrix for 7:3 ratio on training and test data (with grid/axis)

10k	52	0	1	1	0	0	0
1k	1	42	1	8	2	0	0
2k5	1	1	56	1	0	1	0
3k	0	4	2	46	1	0	1
5k	0	1	0	2	57	0	0
7k5	0	0	0	1	0	59	0
9k	0	0	0	0	0	0	60
	10k	1k	2k5	3k	5k	7k5	9k

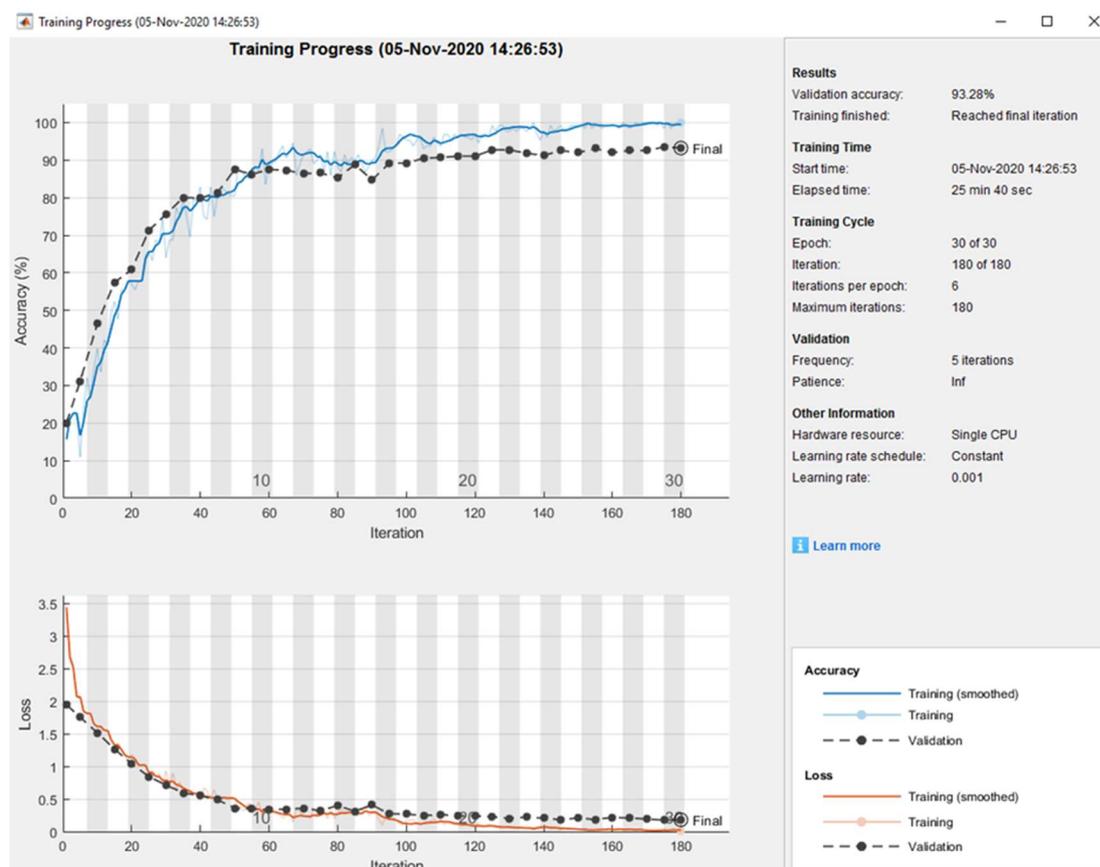
M3C: Training progress for 7:3 ratio on training and test data (no grid/axis)



M3D: Confusion matrix for 7:3 ratio on training and test data (no grid/axis)

10k	49	0	1	0	2	0	2
1k	0	41	3	5	5	0	0
2k5	0	0	55	1	4	0	0
3k	0	5	6	41	2	0	0
5k	1	2	2	0	54	0	1
7k5	0	0	0	0	0	59	1
9k	0	0	1	0	3	0	56
	10k	1k	2k5	3k	5k	7k5	9k

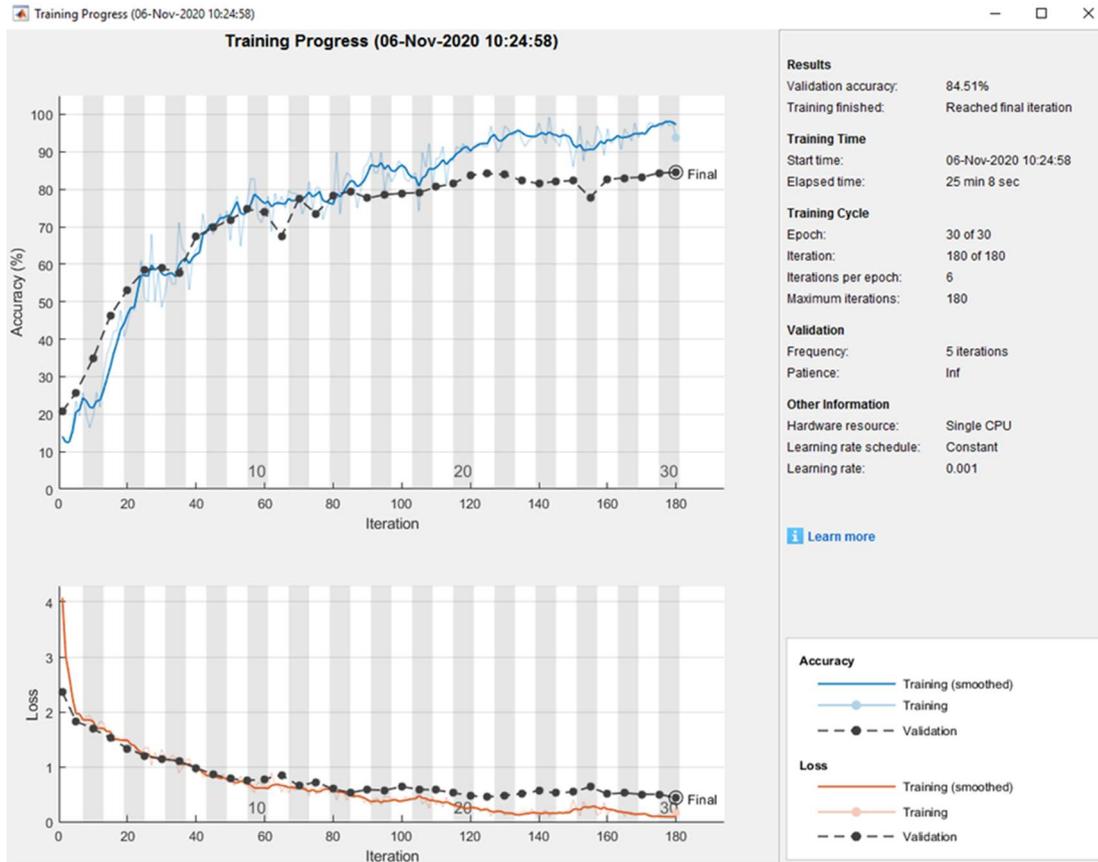
M4A: Training progress for 3:2 ratio on training and test data (with grid/axis)



M4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis)

10k	72	0	0	0	0	0	0
1k	0	60	1	5	5	0	1
2k5	1	1	76	1	0	0	1
3k	0	6	4	59	0	3	0
5k	1	1	1	0	75	0	2
7k5	0	0	2	0	0	78	0
9k	0	0	0	0	0	0	80
	10k	1k	2k5	3k	5k	7k5	9k

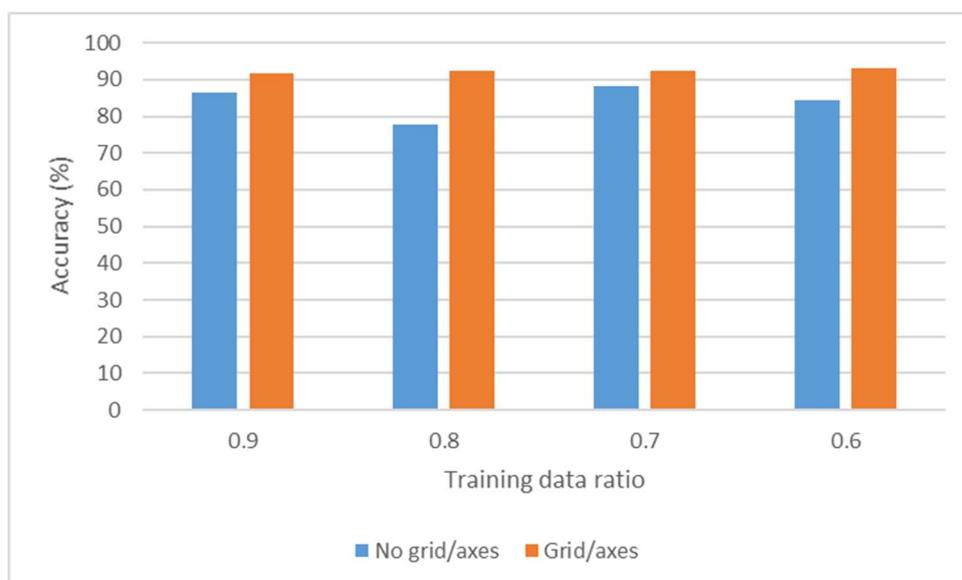
M4C: Training progress for 3:2 ratio on training and test data (no grid/axis)



M4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis)

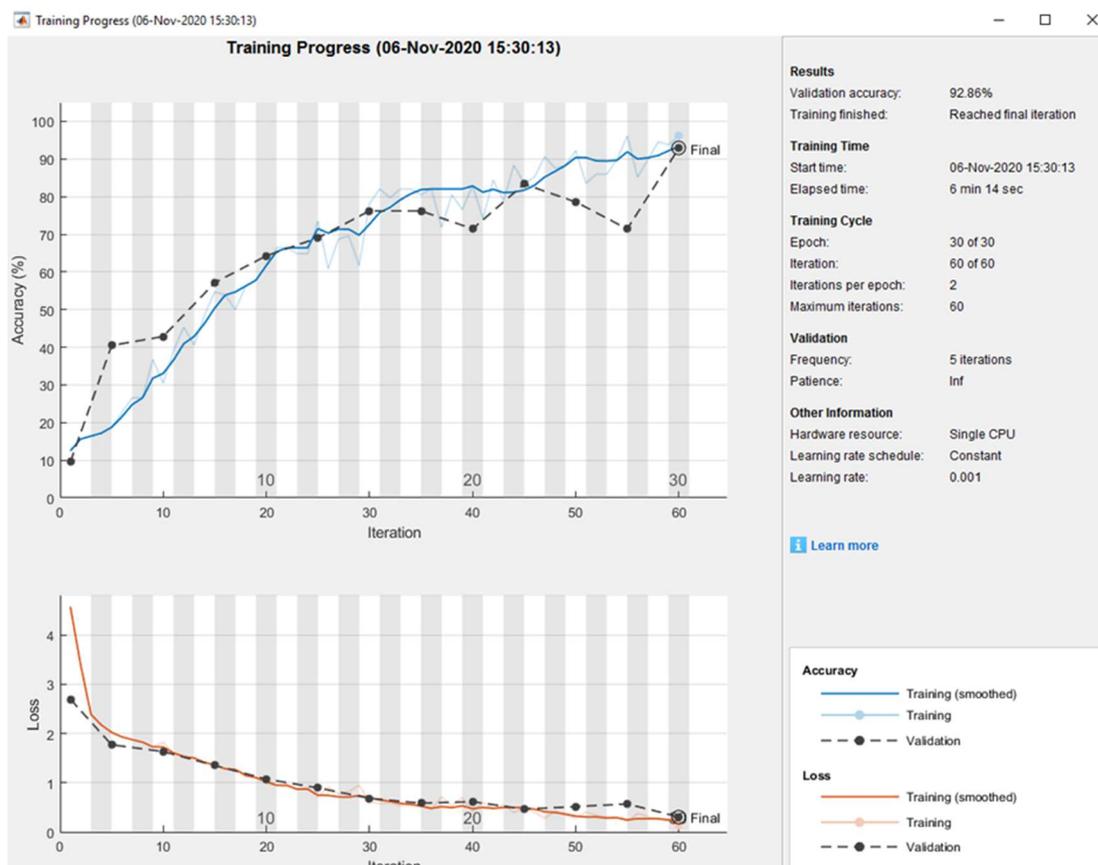
10k	69	0	0	1	1	0	1
1k	0	66	3	2	0	1	0
2k5	2	5	67	1	4	1	0
3k	1	16	8	42	3	1	1
5k	9	8	3	5	54	0	1
7k5	0	0	0	0	0	80	0
9k	1	0	0	0	3	1	75
	10k	1k	2k5	3k	5k	7k5	9k

M5: Comparing results between grid/axis and no grid/axis



Appendix N: DCNN results on speed classification on 5 °C thermal perturbation test

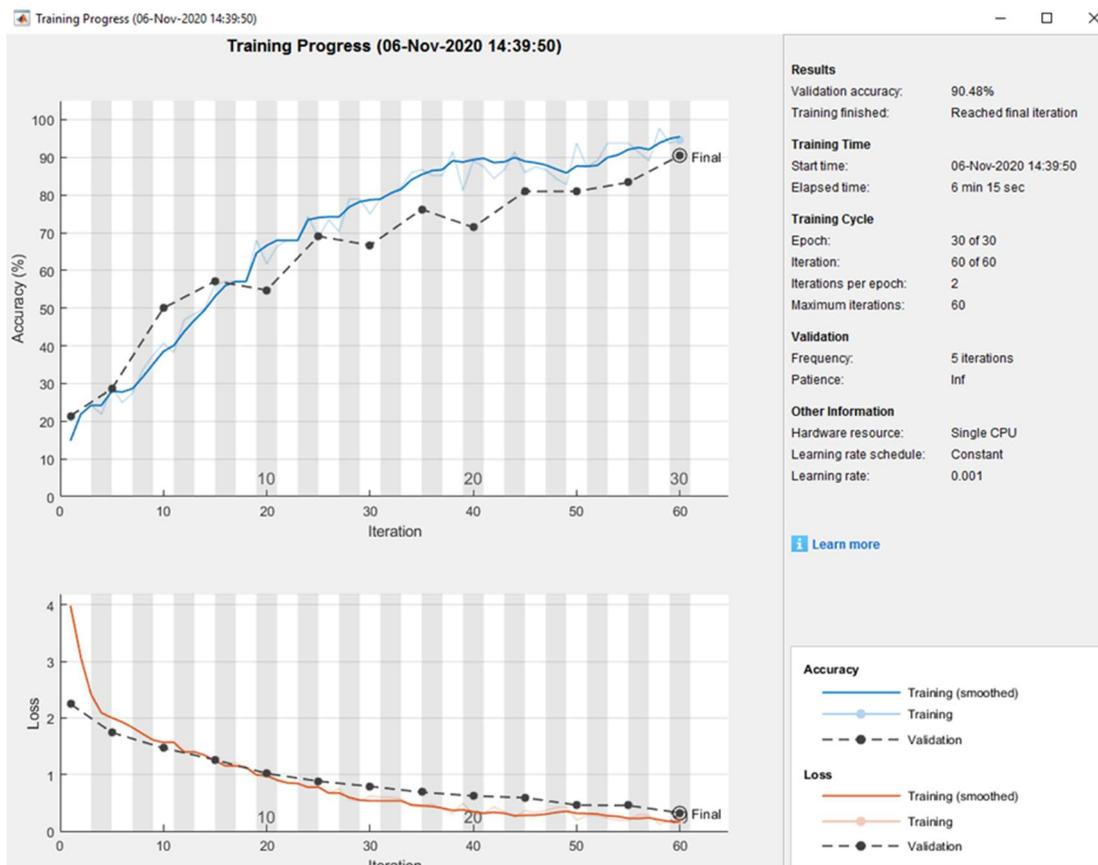
N1A: Training progress for 9:1 ratio on training and test data (with grid/axis)



N1B: Confusion matrix for 9:1 ratio on training and test data (with grid/axis)

10k	5	0	0	0	0	0	1
1k	0	6	0	0	0	0	0
2k5	0	0	5	0	0	0	1
3k	0	0	0	6	0	0	0
5k	0	0	0	0	6	0	0
7k5	0	0	0	0	0	6	0
9k	1	0	0	0	0	0	5
	10k	1k	2k5	3k	5k	7k5	9k

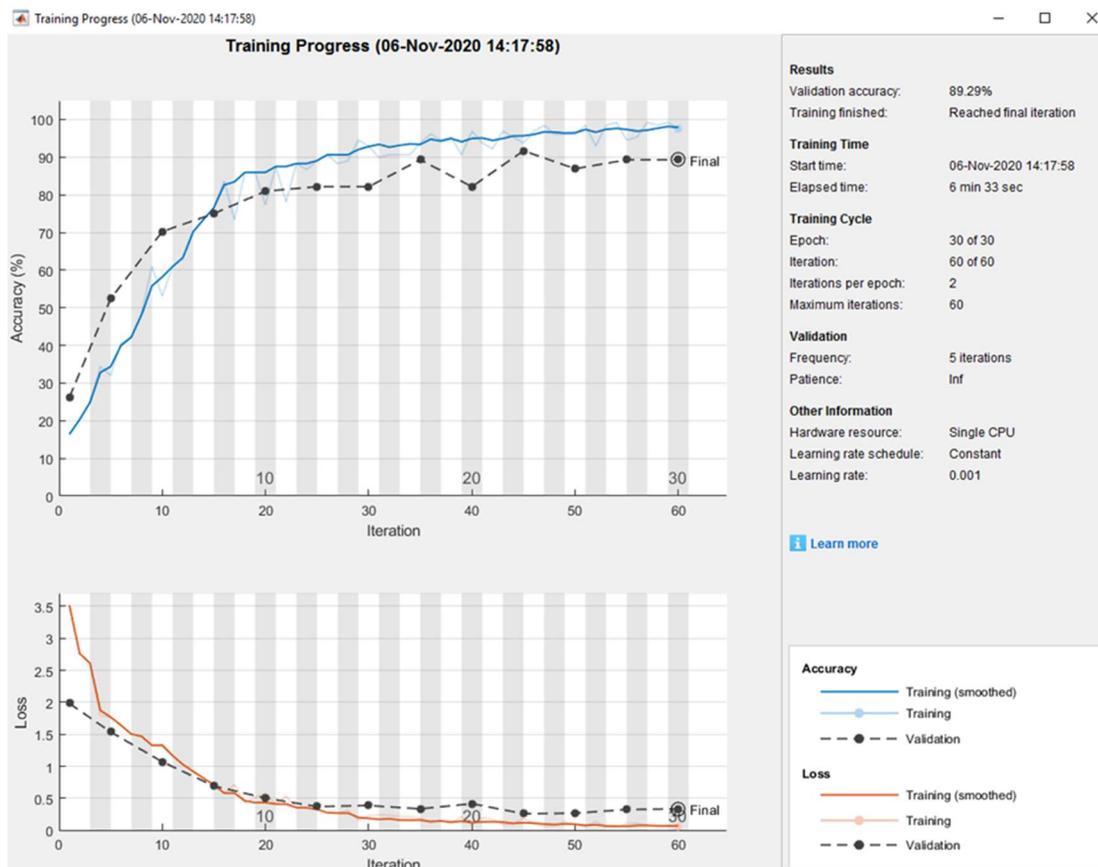
N1C: Training progress for 9:1 ratio on training and test data (no grid/axis)



N1D: Confusion matrix for 9:1 ratio on training and test data (no grid/axis)

10k	6	0	0	0	0	0	0
1k	1	5	0	0	0	0	0
2k5	0	0	5	0	0	0	1
3k	0	2	0	4	0	0	0
5k	0	0	0	0	6	0	0
7k5	0	0	0	0	0	6	0
9k	0	0	0	0	0	0	6
	10k	1k	2k5	3k	5k	7k5	9k

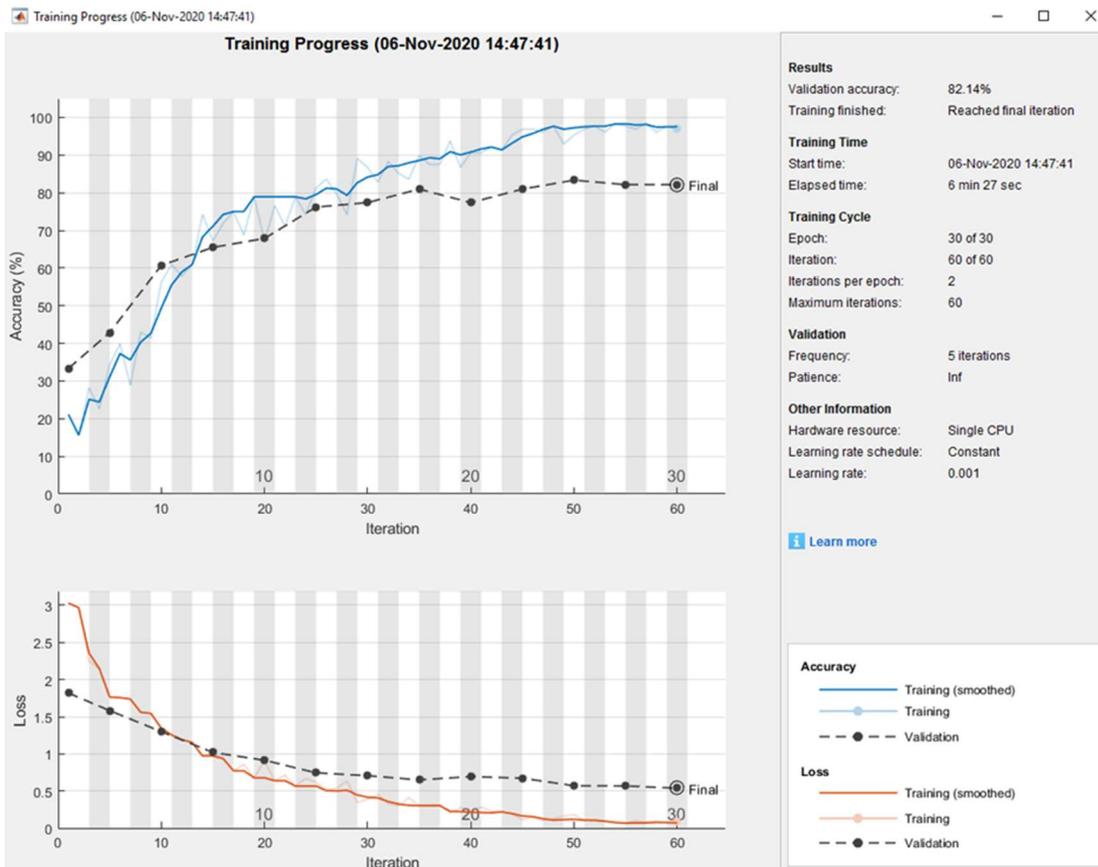
N2A: Training progress for 4:1 ratio on training and test data (with grid/axis)



N2B: Confusion matrix for 4:1 ratio on training and test data (with grid/axis)

10k	12	0	0	0	0	0	0
1k	1	9	2	0	0	0	0
2k5	0	0	12	0	0	0	0
3k	0	1	0	10	1	0	0
5k	0	0	0	0	12	0	0
7k5	0	0	0	0	0	12	0
9k	4	0	0	0	0	0	8
	10k	1k	2k5	3k	5k	7k5	9k

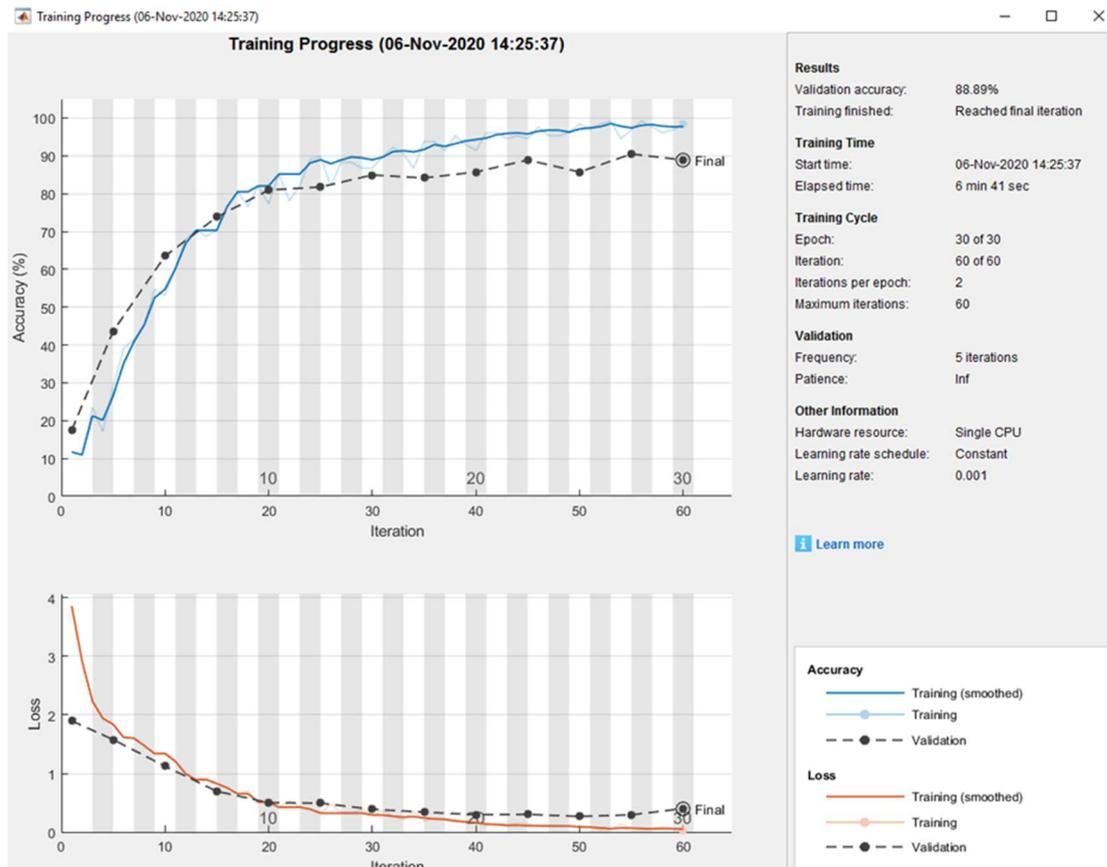
N2C: Training progress for 4:1 ratio on training and test data (no grid/axis)



N2D: Confusion matrix for 4:1 ratio on training and test data (no grid/axis)

10k	8	0	0	1	0	0	3
1k	0	9	2	0	1	0	0
2k5	0	1	10	0	0	0	1
3k	3	2	0	7	0	0	0
5k	0	0	0	0	12	0	0
7k5	0	0	0	0	0	12	0
9k	1	0	0	0	0	0	11
	10k	1k	2k5	3k	5k	7k5	9k

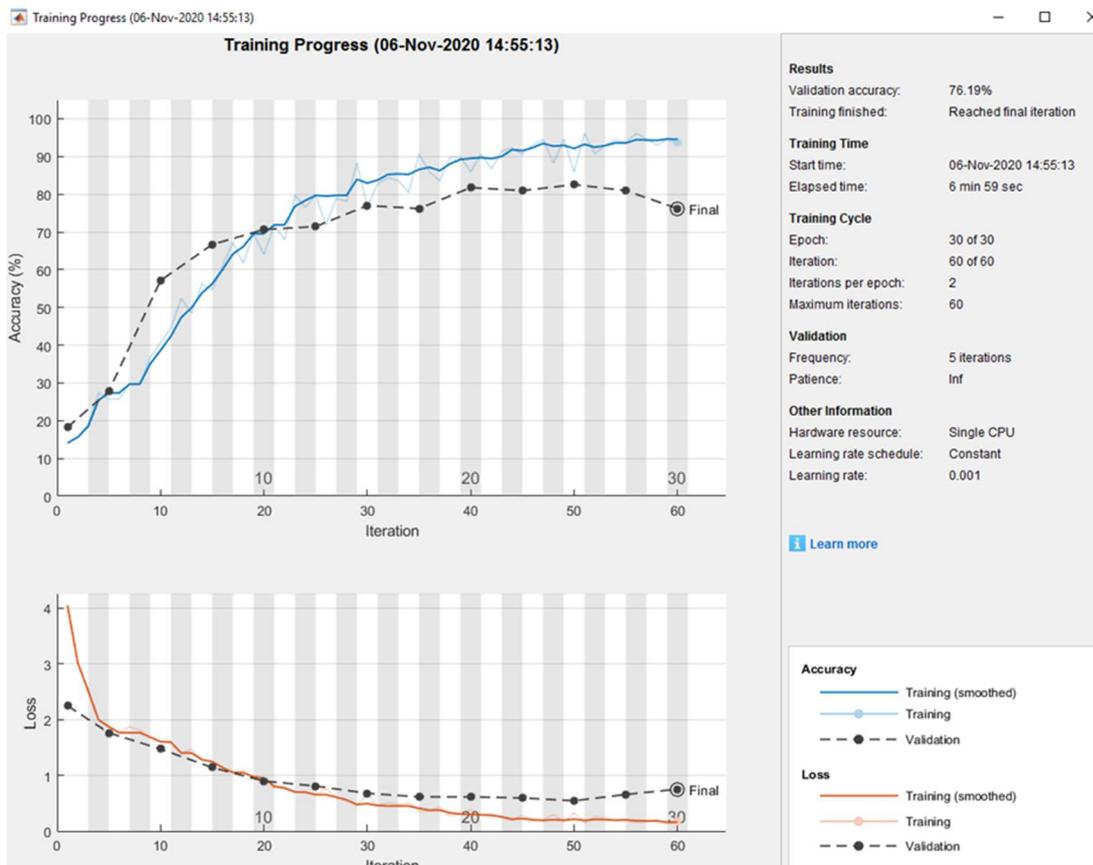
N3A: Training progress for 7:1 ratio on training and test data (with grid/axis)



N3B: Confusion matrix for 7:1 ratio on training and test data (with grid/axis)

10k	15	0	0	0	0	0	3
1k	2	16	0	0	0	0	0
2k5	0	0	18	0	0	0	0
3k	2	6	0	10	0	0	0
5k	0	0	0	0	18	0	0
7k5	0	0	0	0	0	18	0
9k	1	0	0	0	0	0	17
	10k	1k	2k5	3k	5k	7k5	9k

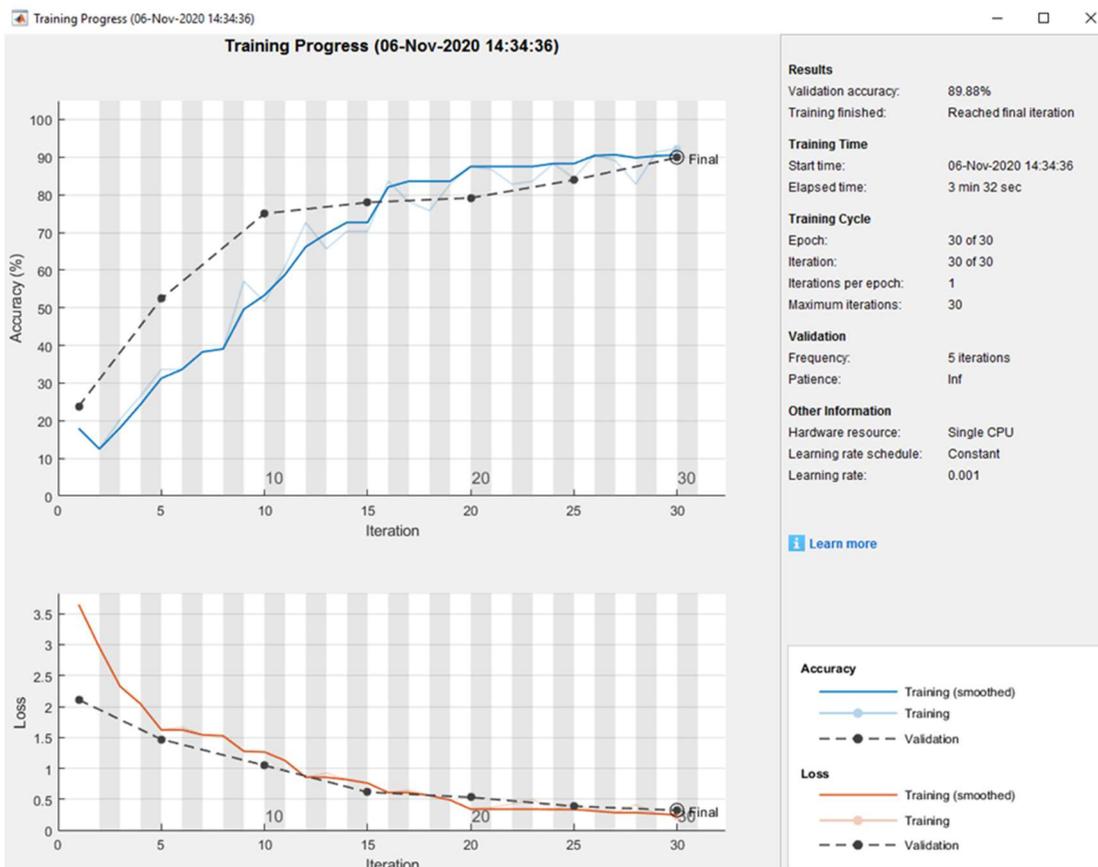
N3C: Training progress for 7:1 ratio on training and test data (no grid/axis)



N3D: Confusion matrix for 7:1 ratio on training and test data (no grid/axis)

10k	11	0	0	0	1	0	6
1k	6	8	4	0	0	0	0
2k5	3	1	11	0	0	0	3
3k	1	3	0	14	0	0	0
5k	1	0	0	0	17	0	0
7k5	0	0	0	0	0	18	0
9k	1	0	0	0	0	0	17
	10k	1k	2k5	3k	5k	7k5	9k

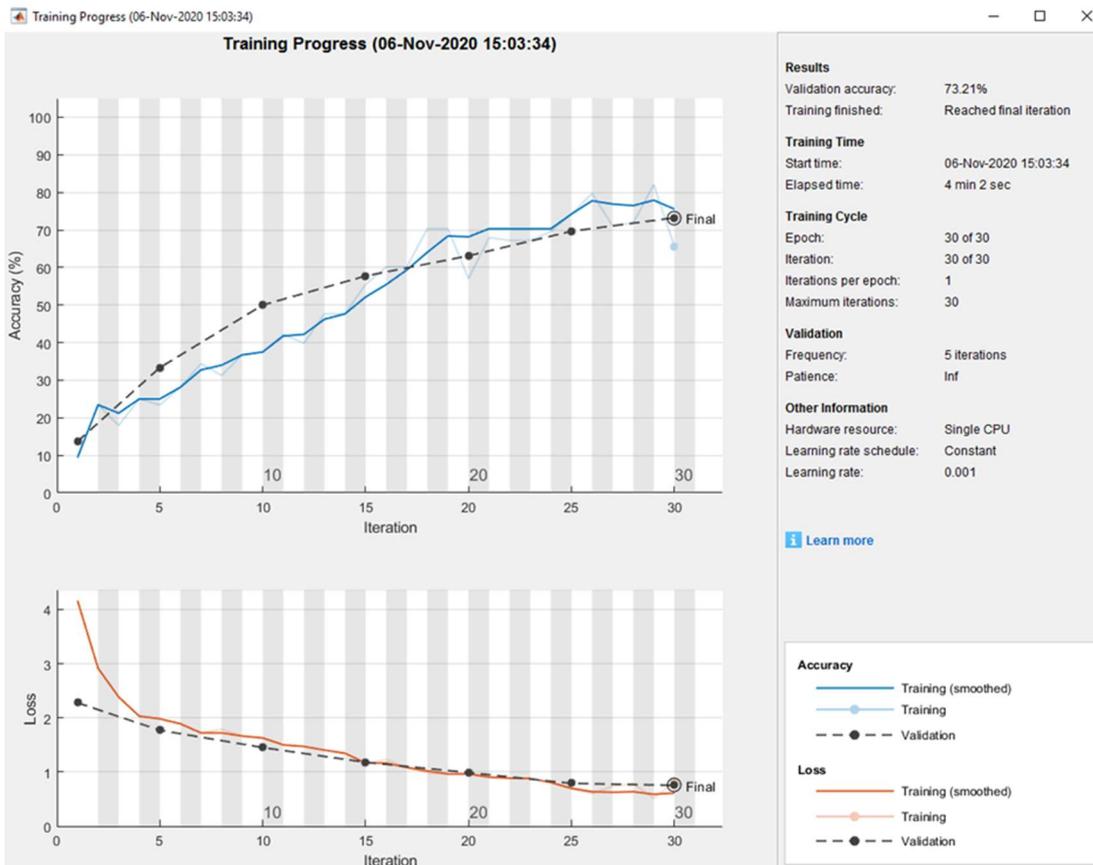
N4A: Training progress for 3:2 ratio on training and test data (with grid/axis)



N4B: Confusion matrix for 3:2 ratio on training and test data (with grid/axis)

10k	15	1	0	3	0	0	5
1k	1	23	0	0	0	0	0
2k5	0	1	22	1	0	0	0
3k	0	3	0	21	0	0	0
5k	0	0	0	0	24	0	0
7k5	0	0	0	0	0	24	0
9k	2	0	0	0	0	0	22
	10k	1k	2k5	3k	5k	7k5	9k

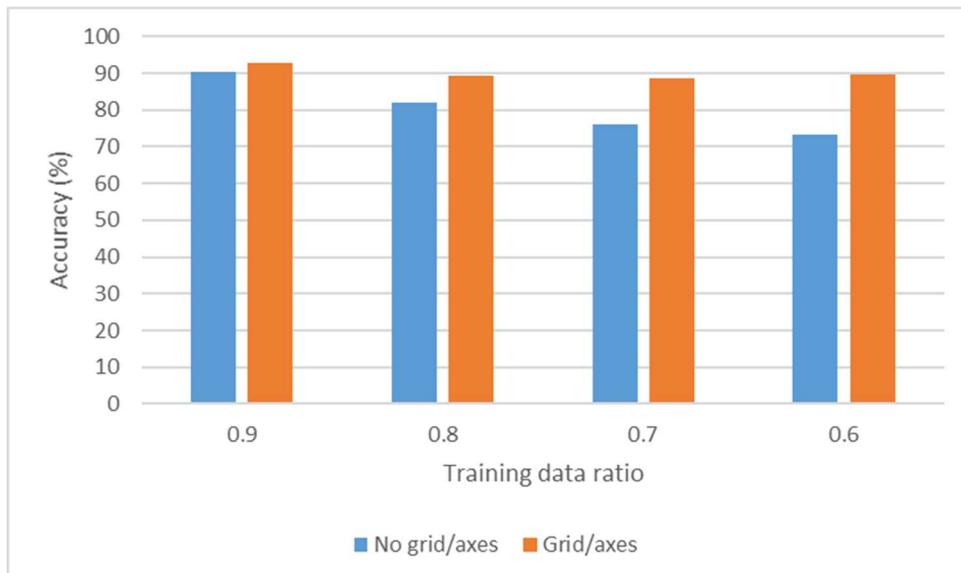
N4C: Training progress for 3:2 ratio on training and test data (no grid/axis)

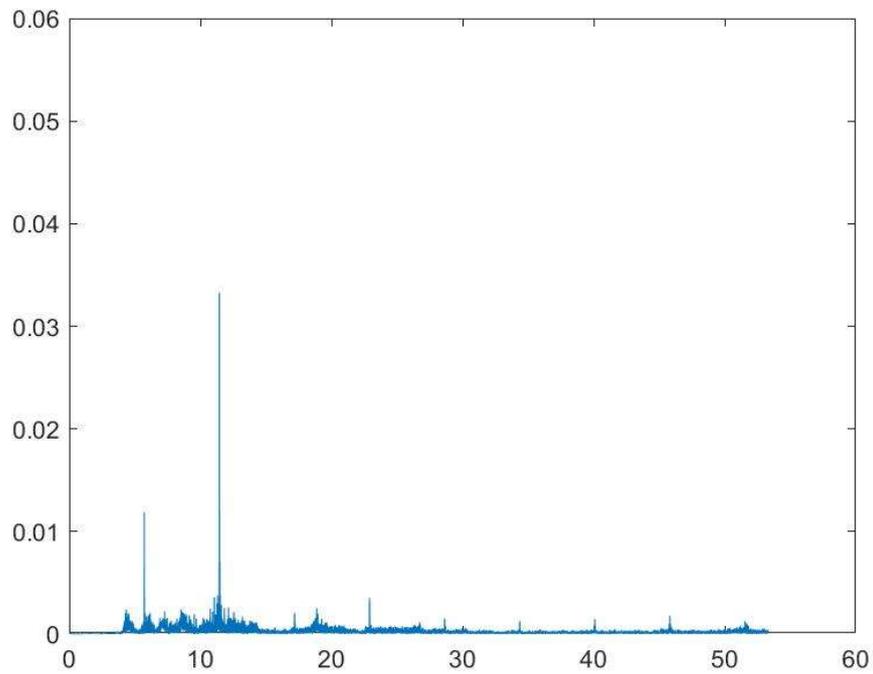
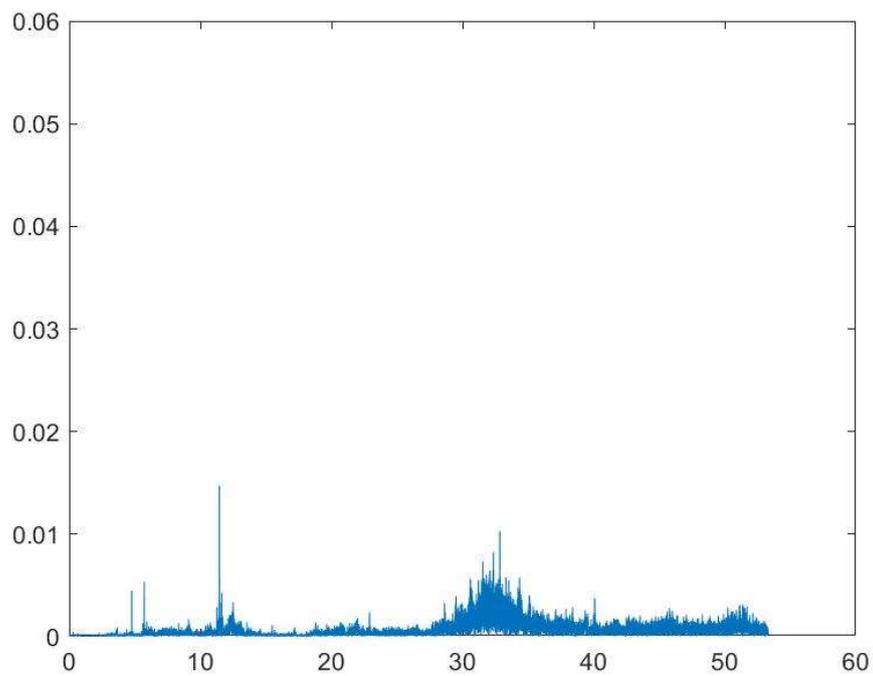


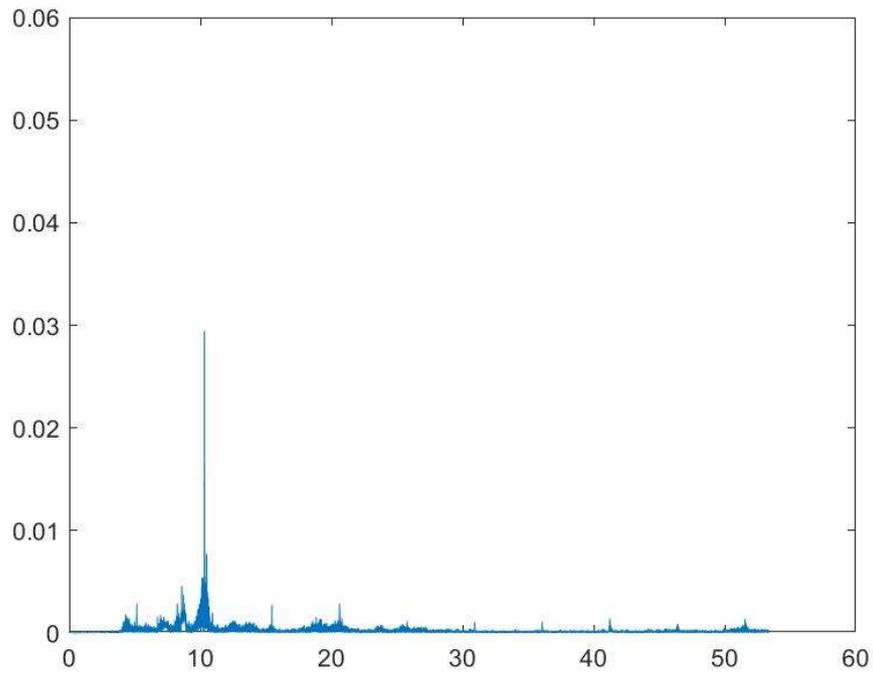
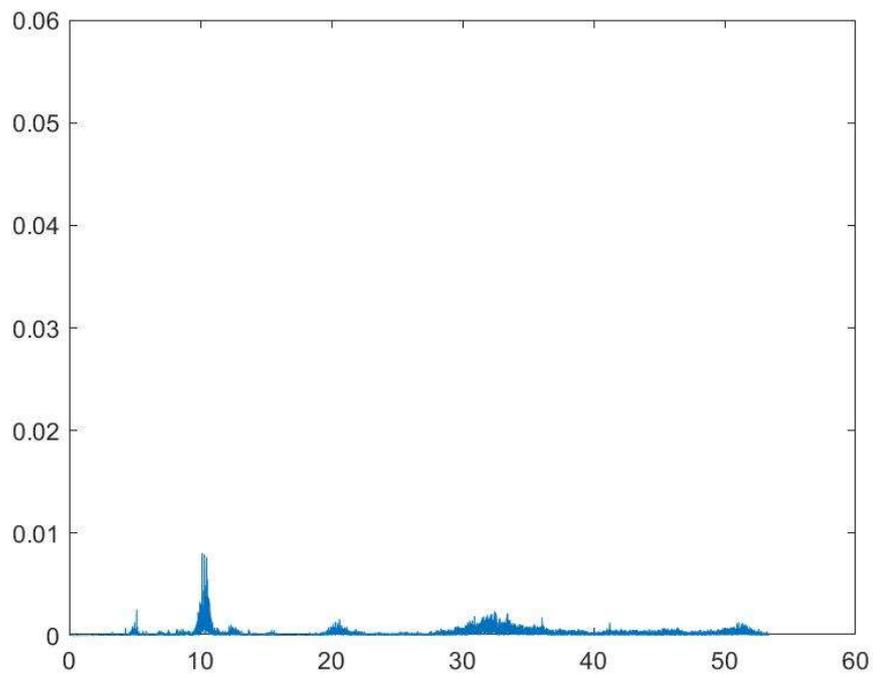
N4D: Confusion matrix for 3:2 ratio on training and test data (no grid/axis)

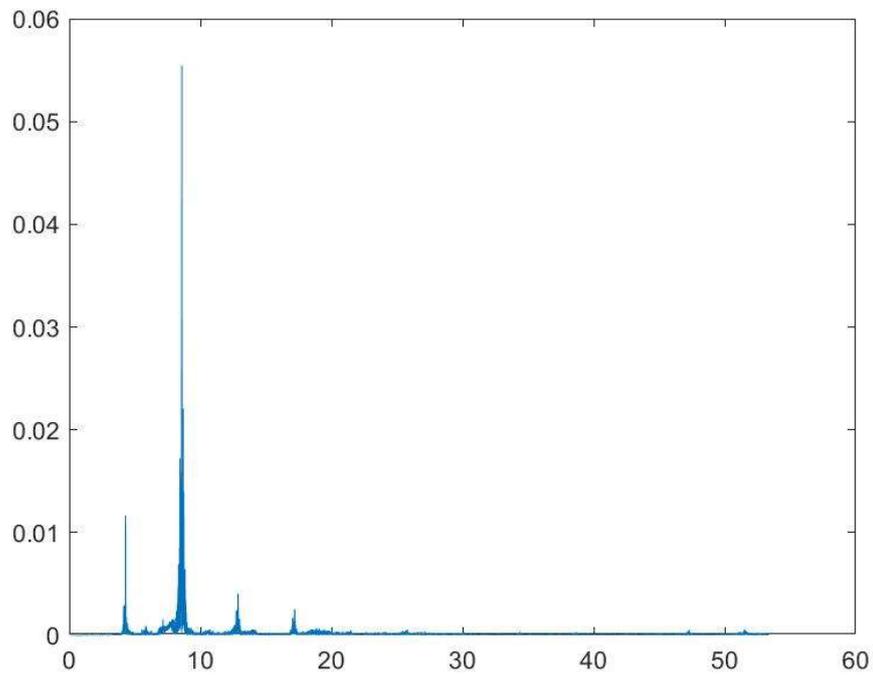
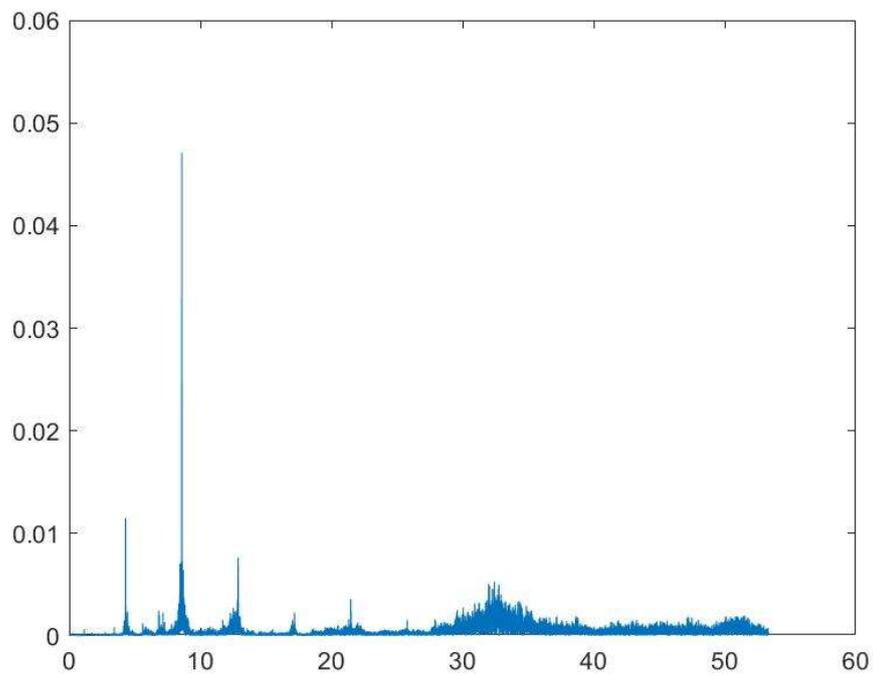
10k	15	3	0	2	1	0	3
1k	2	21	0	1	0	0	0
2k5	5	3	13	0	0	0	3
3k	5	5	0	11	2	0	1
5k	1	0	0	0	23	0	0
7k5	0	0	1	0	0	23	0
9k	5	0	0	0	2	0	17
	10k	1k	2k5	3k	5k	7k5	9k

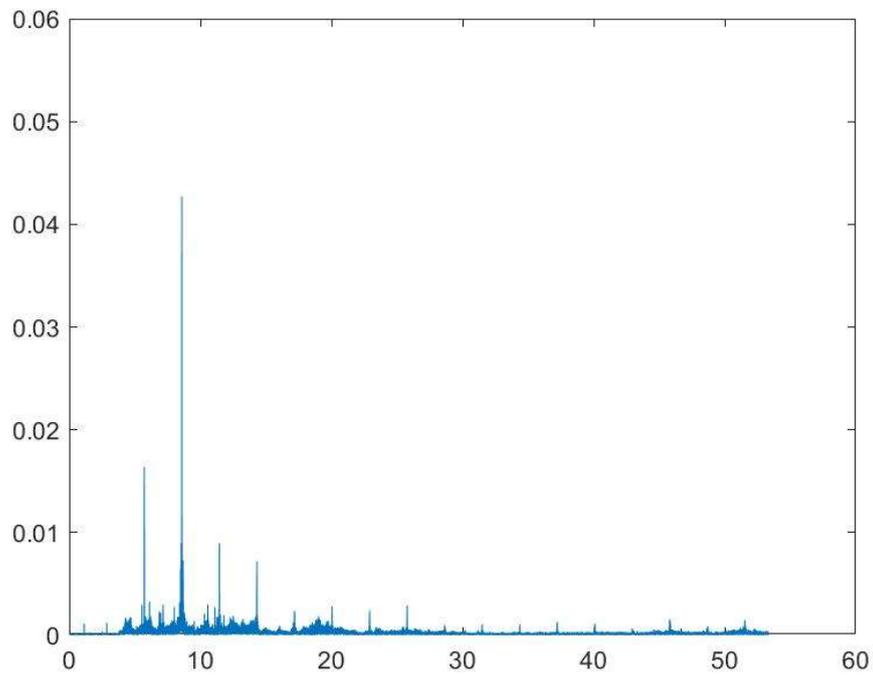
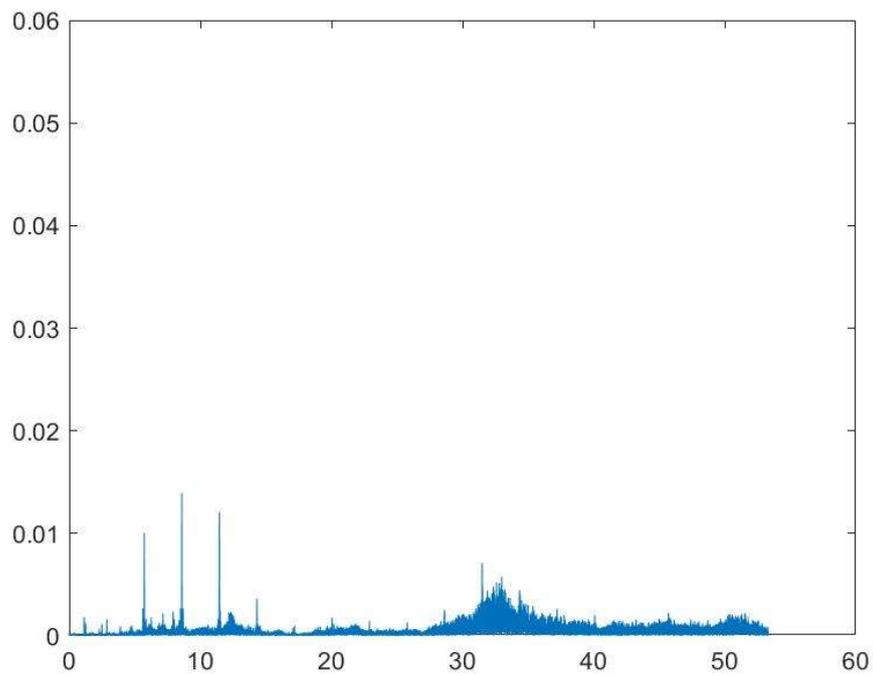
N5: Comparing results between grid/axis and no grid/axis

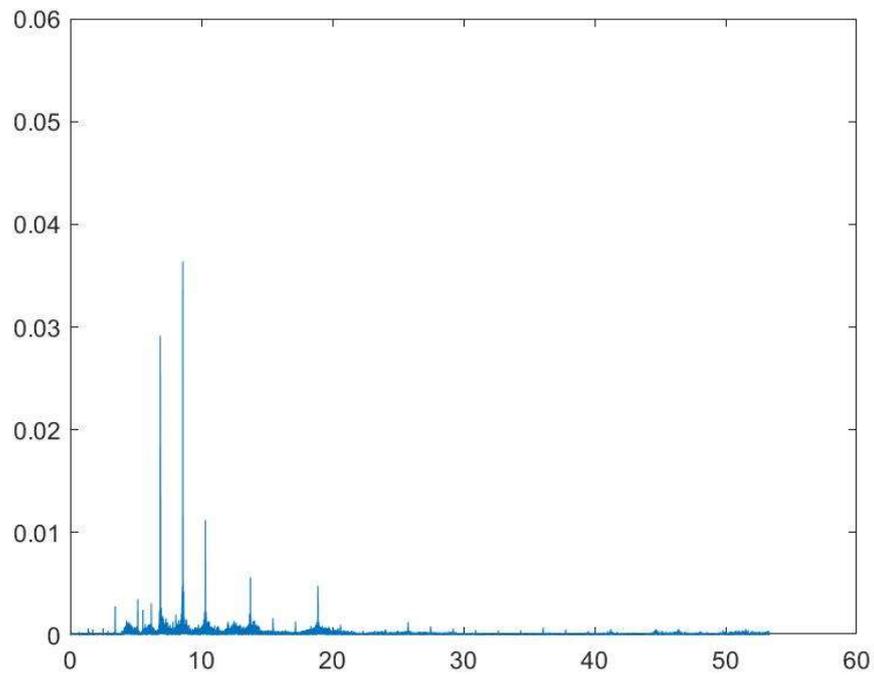
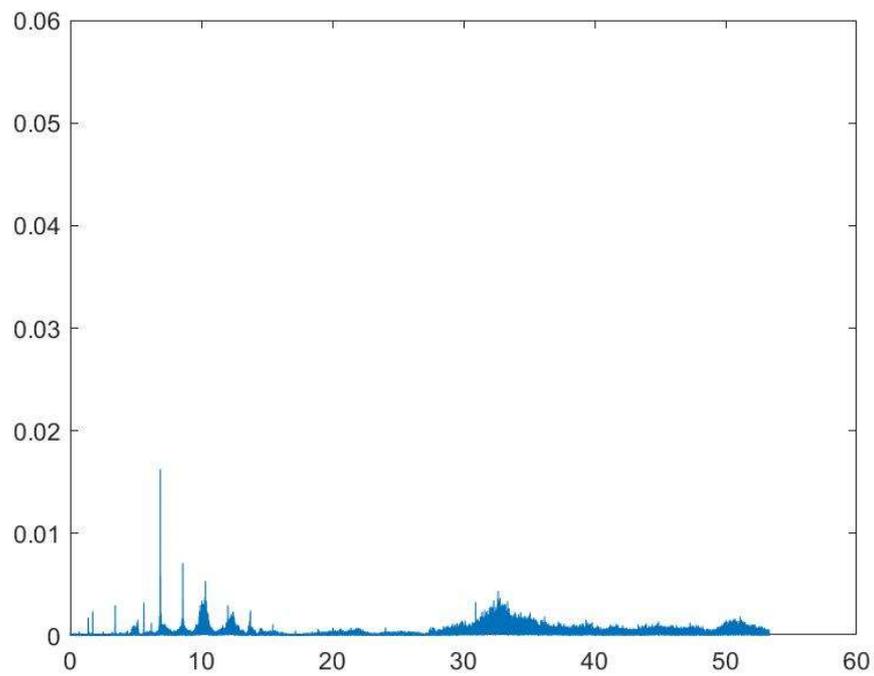


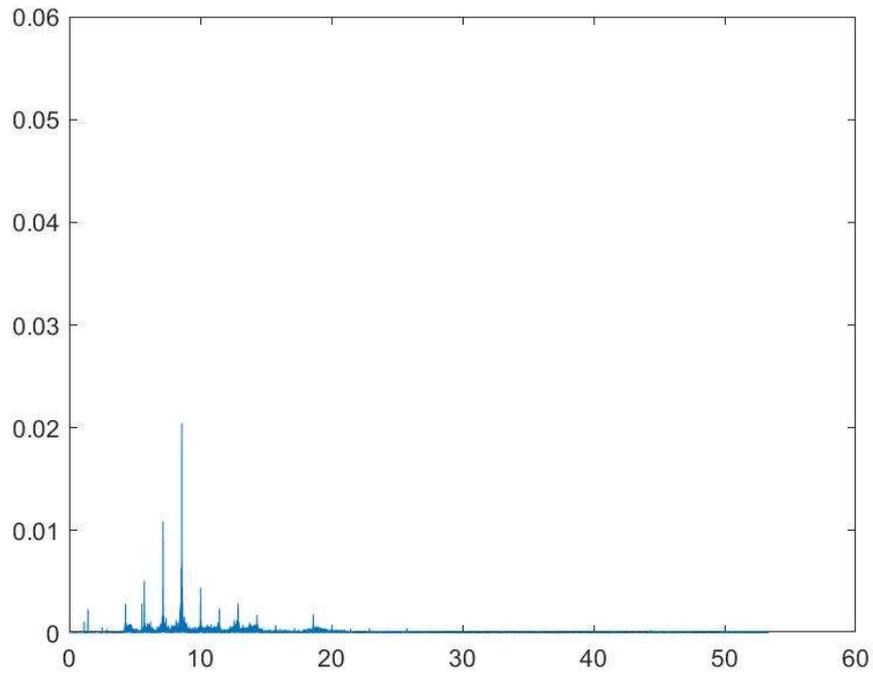
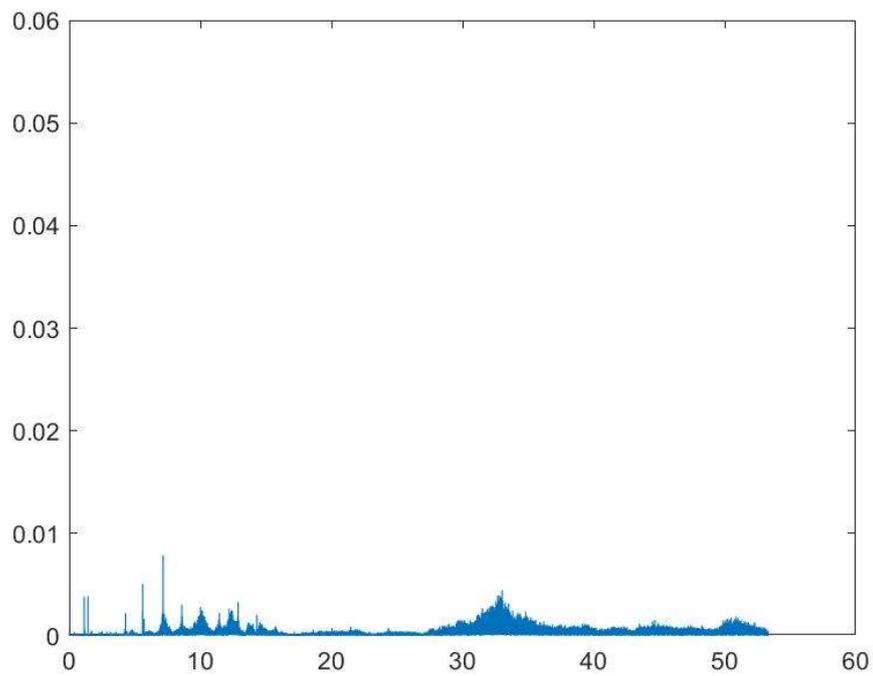
Appendix O: Normalized FFT data for baseline test with fixed axis**O1A: Normalized FFT at 10000 mm/min for good ball screw****O1B: Normalized FFT at 10000 mm/min for bad ball screw**

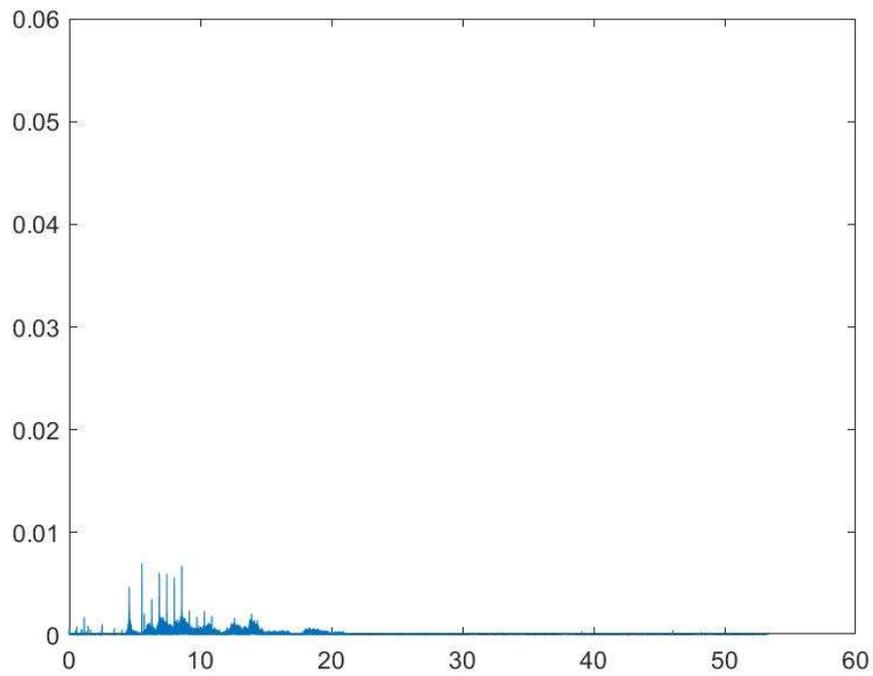
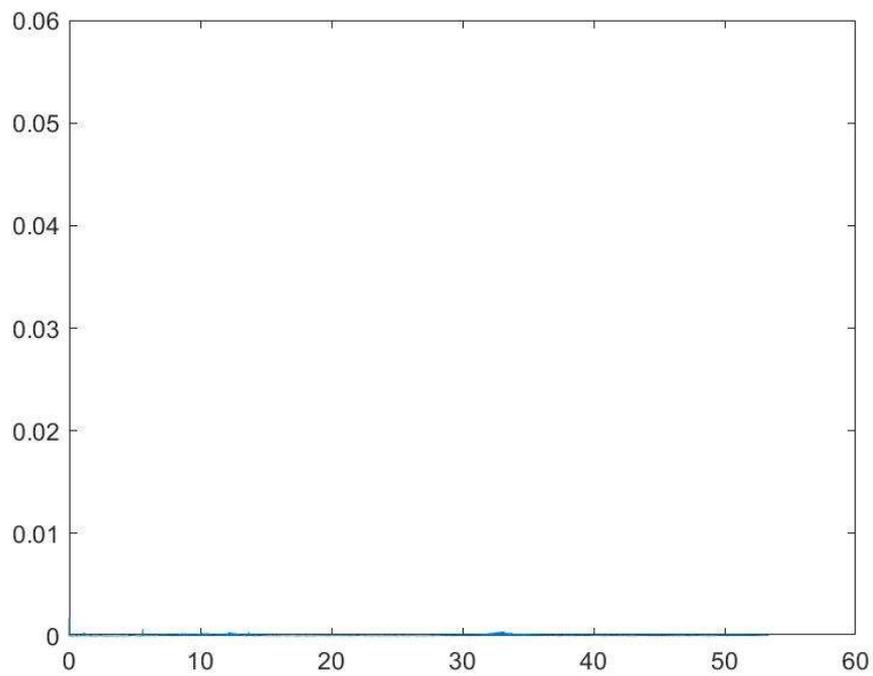
O2A: Normalized FFT at 9000 mm/min for good ball screw**O2B: Normalized FFT at 9000 mm/min for bad ball screw**

O3A: Normalized FFT at 7500 mm/min for good ball screw**O3B: Normalized FFT at 7500 mm/min for bad ball screw**

O4A: Normalized FFT at 5000 mm/min for good ball screw**O4B: Normalized FFT at 5000 mm/min for bad ball screw**

O5A: Normalized FFT at 3000 mm/min for good ball screw**O5B: Normalized FFT at 3000 mm/min for bad ball screw**

O6A: Normalized FFT at 2500 mm/min for good ball screw**O6B: Normalized FFT at 2500 mm/min for bad ball screw**

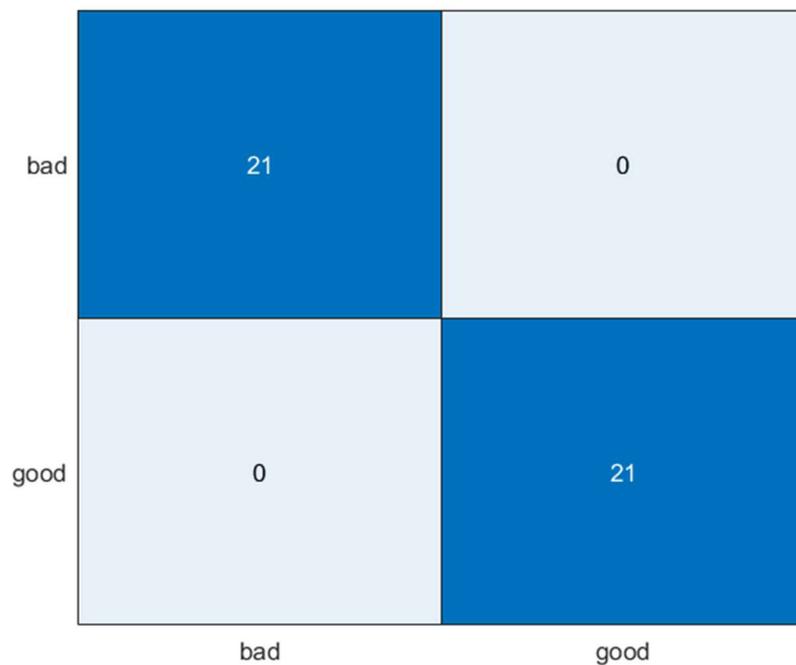
O7A: Normalized FFT at 1000 mm/min for good ball screw**O7B: Normalized FFT at 1000 mm/min for bad ball screw**

Appendix P: DCNN results for normalized FFT of baseline test

P1: Training progress for 9:1 ratio on training and test set



P2: Confusion matrix for 9:1 ratio on training and test set



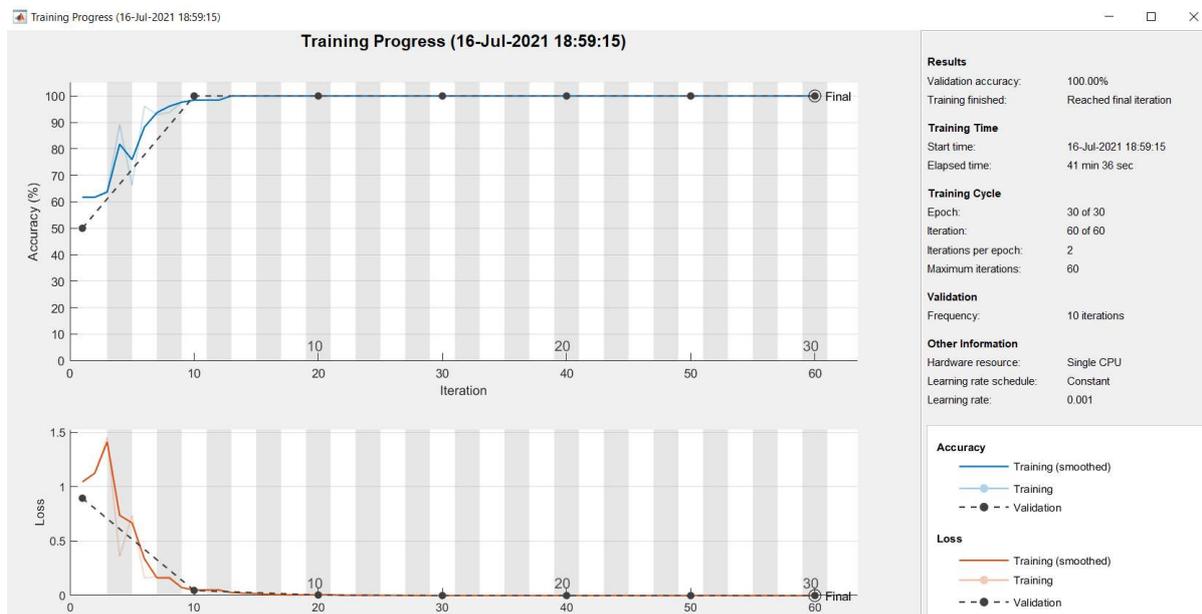
P3: Training progress for 4:1 ratio on training and test set



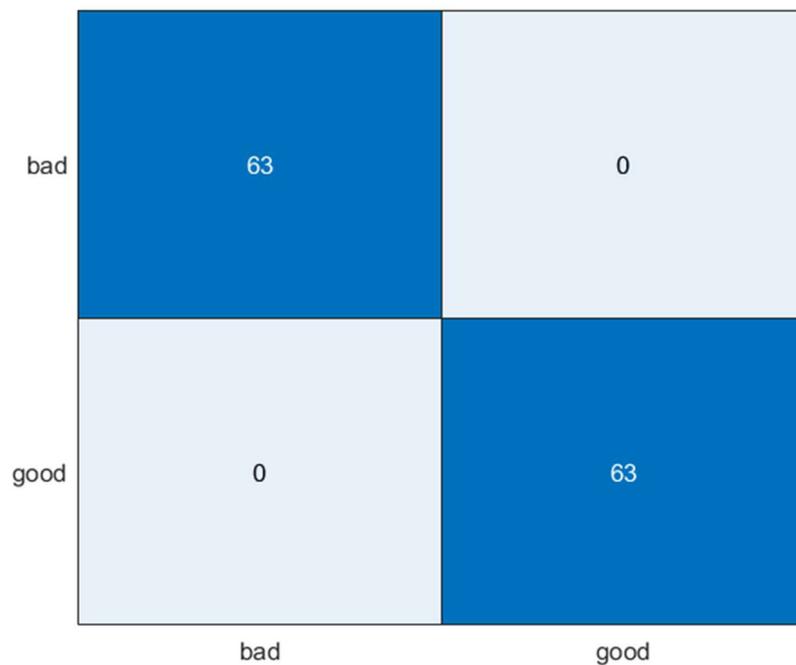
P4: Confusion matrix for 4:1 ratio on training and test set



P5: Training progress for 7:3 ratio on training and test set



P6: Confusion matrix for 7:3 ratio on training and test set



P7: Training progress for 3:2 ratio on training and test set

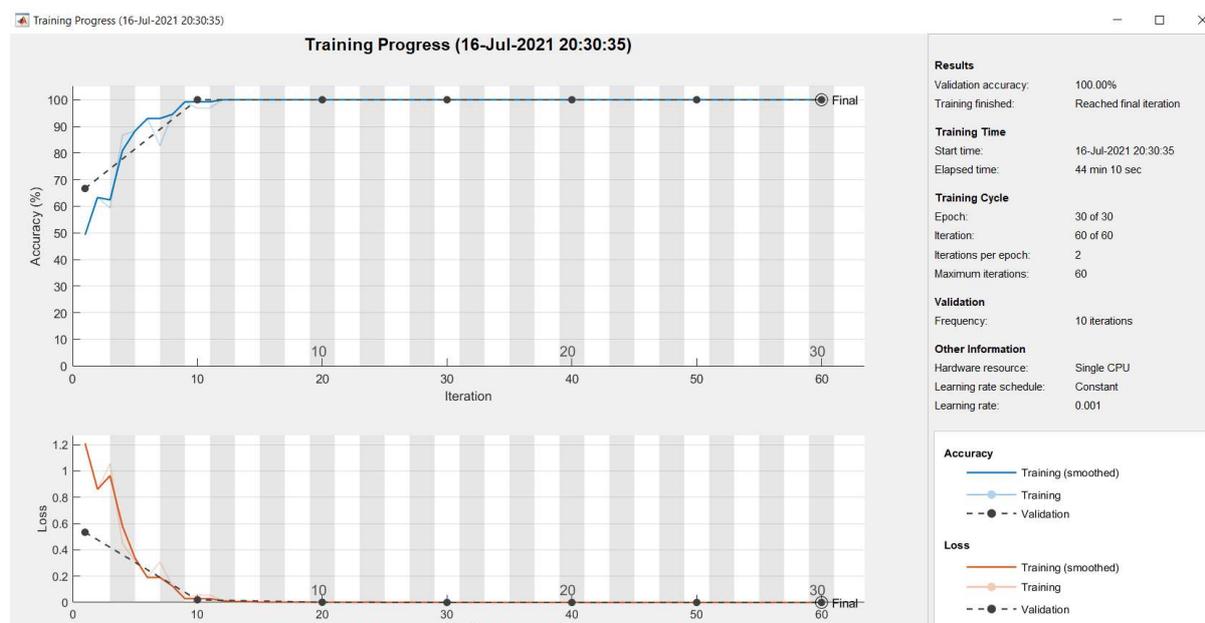


P8: Confusion matrix for 3:2 ratio on training and test set



Appendix Q: DCNN results for normalized FFT of misaligned sensor axis test

Q1: Training progress for 9:1 ratio on training and test set



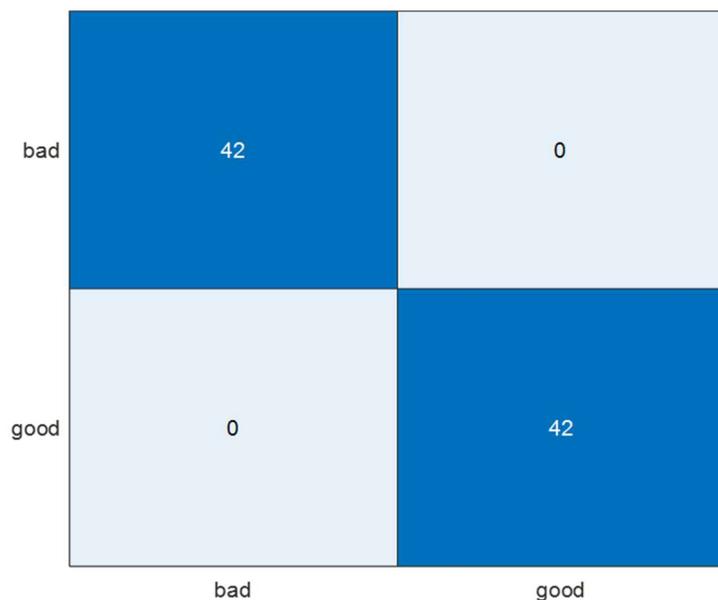
Q2: Confusion matrix for 9:1 ratio on training and test set



Q3: Training progress for 4:1 ratio on training and test set



Q4: Confusion matrix for 4:1 ratio on training and test set



Q5: Training progress for 7:3 ratio on training and test set



Q6: Confusion matrix for 7:3 ratio on training and test set



Q7: Training progress for 3:2 ratio on training and test set

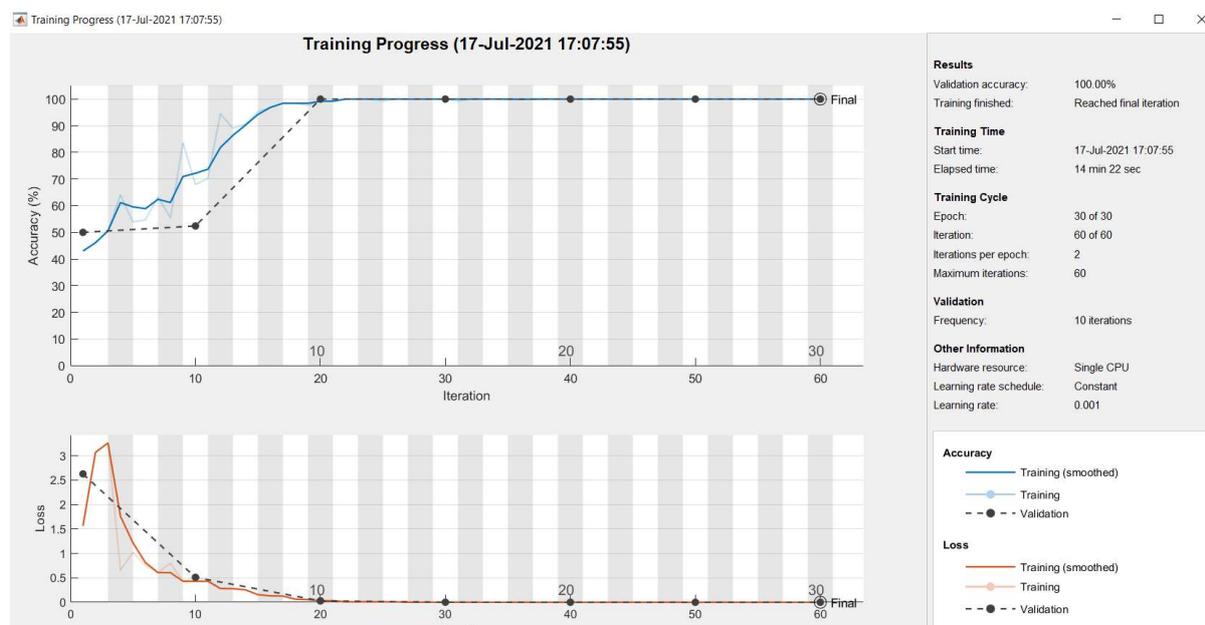


Q8: Confusion matrix for 3:2 ratio on training and test set



Appendix R: DCNN results for normalized FFT of 5 °C thermal perturbation test

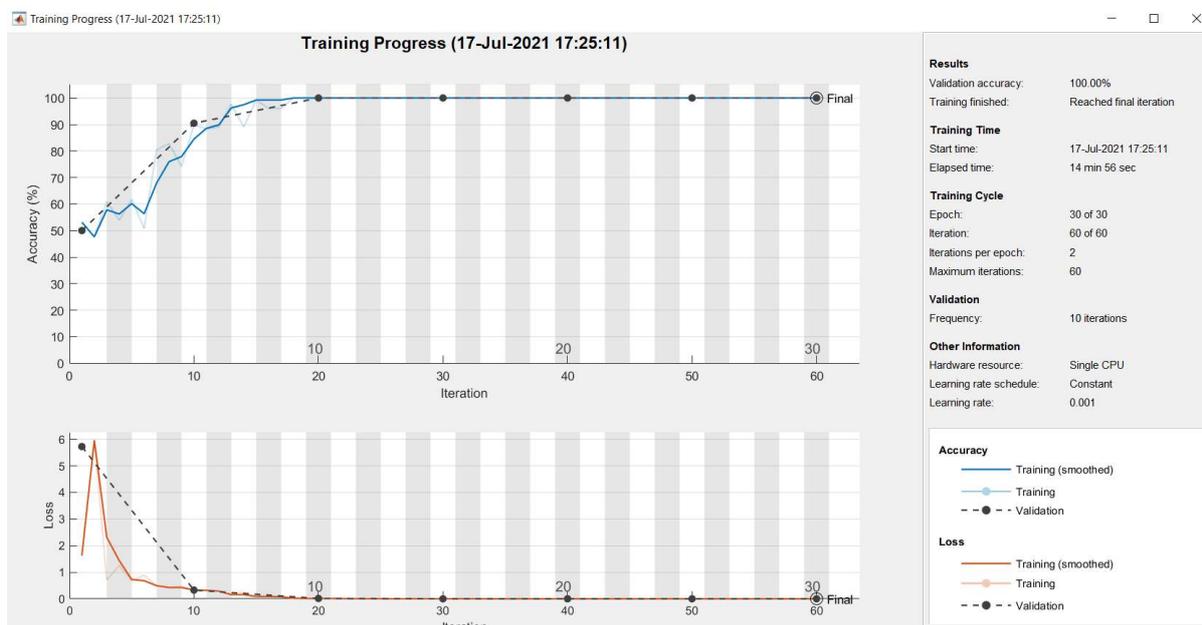
R1: Training progress for 9:1 ratio on training and test set



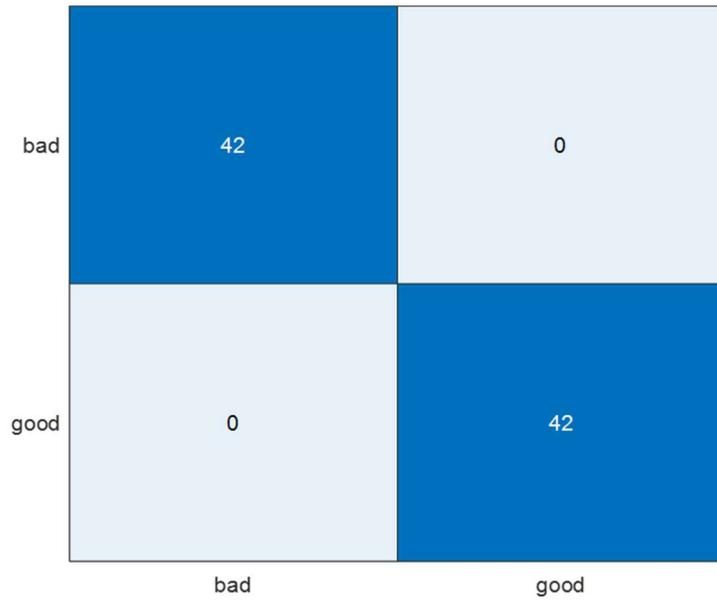
R2: Confusion matrix for 9:1 ratio on training and test set



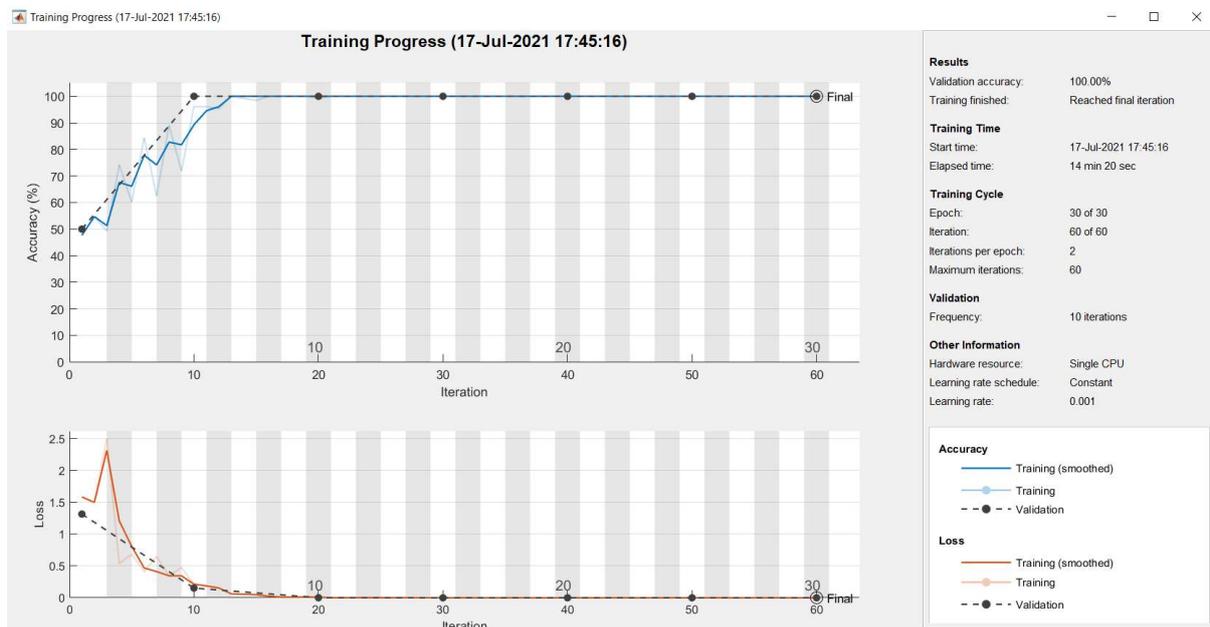
R3: Training progress for 4:1 ratio on training and test set



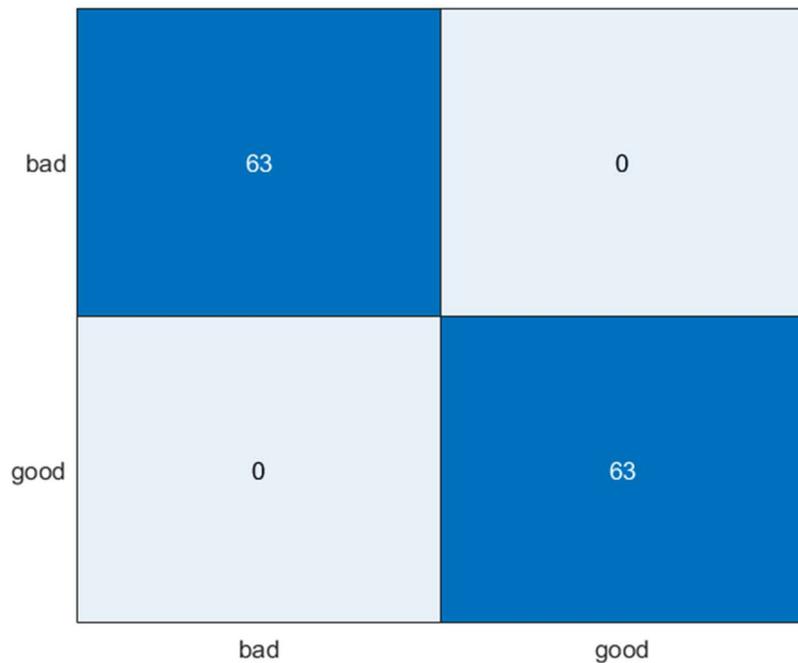
R4: Confusion matrix for 4:1 ratio on training and test set



R5: Training progress for 7:3 ratio on training and test set



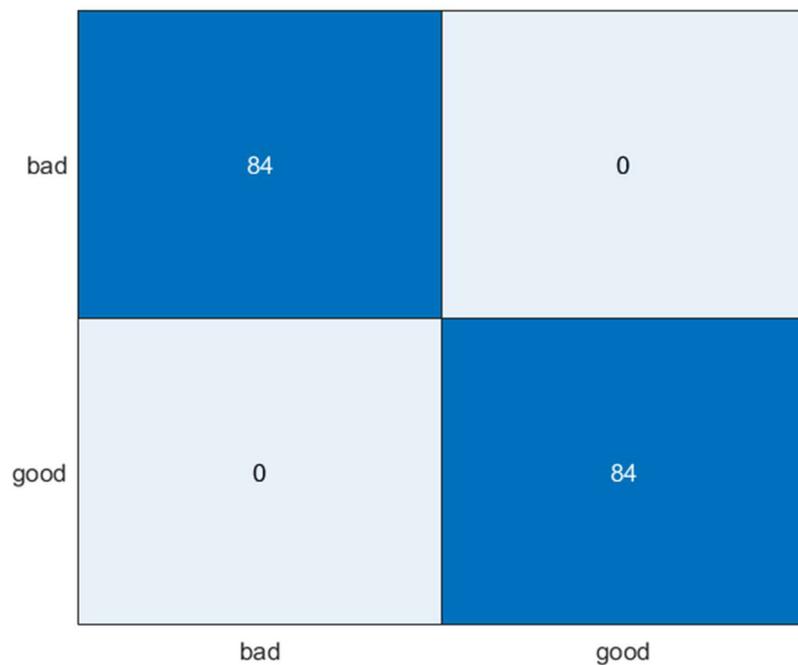
R6: Confusion matrix for 7:3 ratio on training and test set



R7: Training progress for 3:2 ratio on training and test set

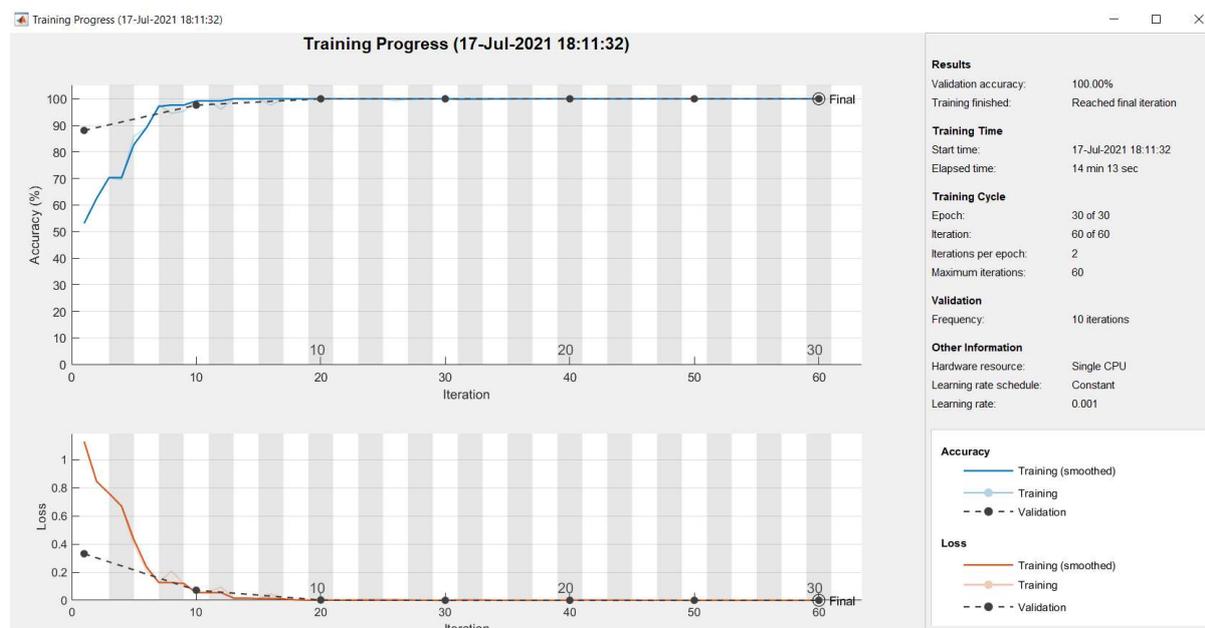


R8: Confusion matrix for 3:2 ratio on training and test set

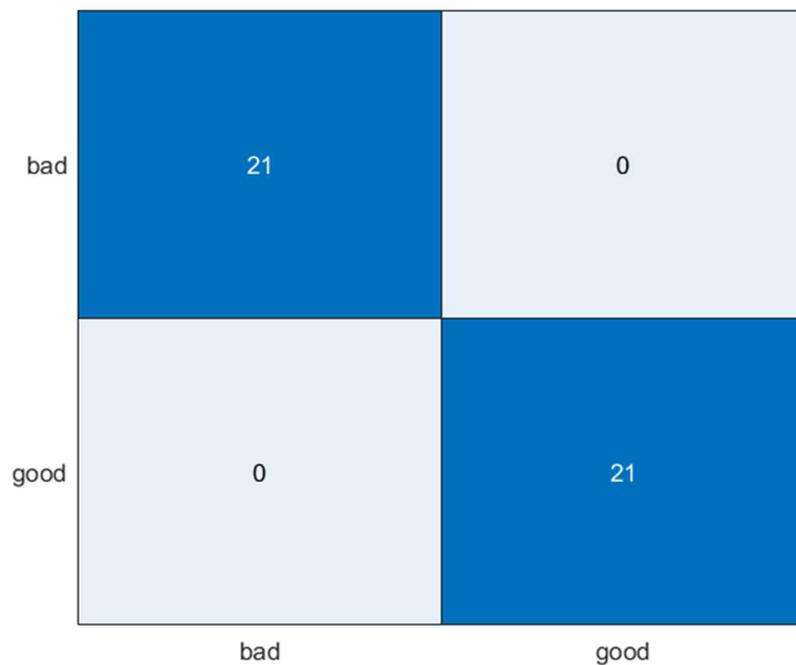


Appendix S: DCNN results for normalized FFT of 10 °C thermal perturbation test

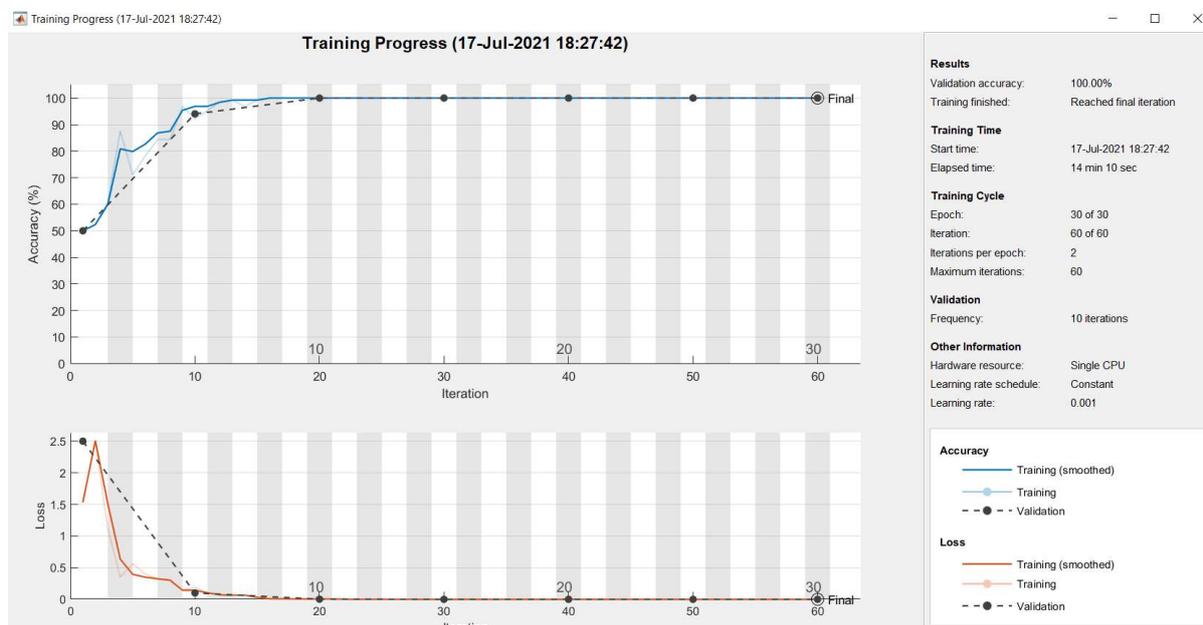
S1: Training progress for 9:1 ratio on training and test set



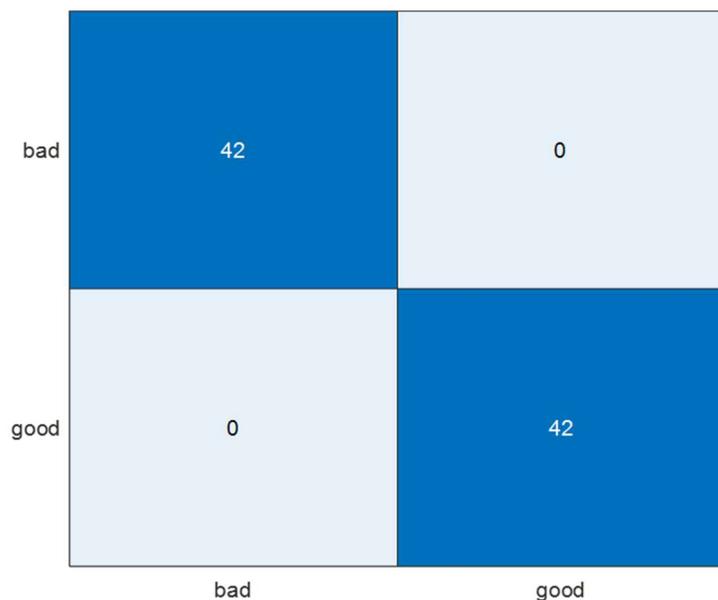
S2: Confusion matrix for 9:1 ratio on training and test set



S3: Training progress for 4:1 ratio on training and test set



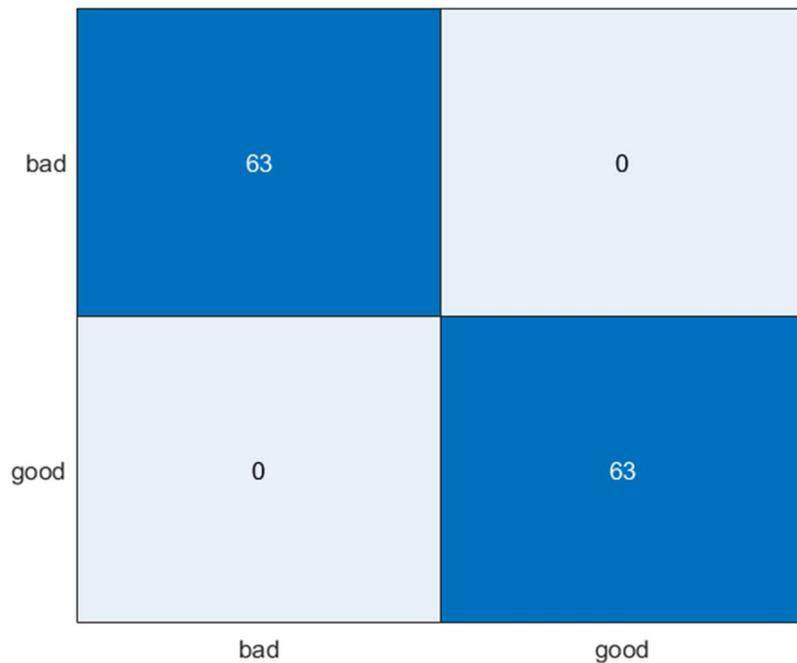
S4: Confusion matrix for 4:1 ratio on training and test set



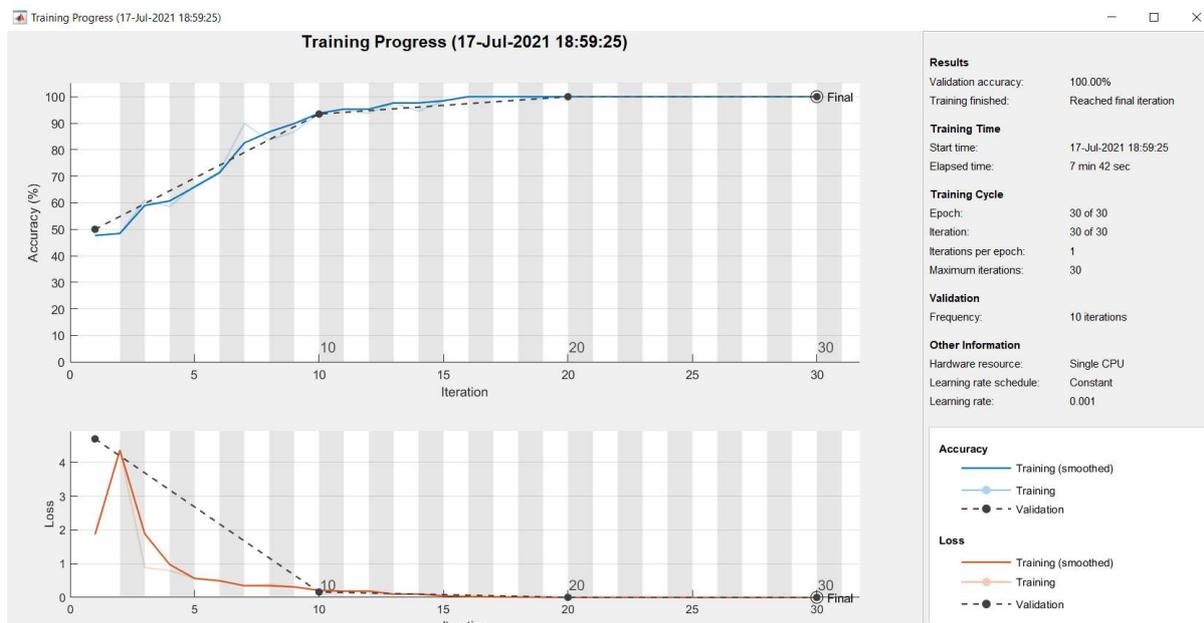
S5: Training progress for 7:3 ratio on training and test set



S6: Confusion matrix for 7:3 ratio on training and test set



S7: Training progress for 3:2 ratio on training and test set

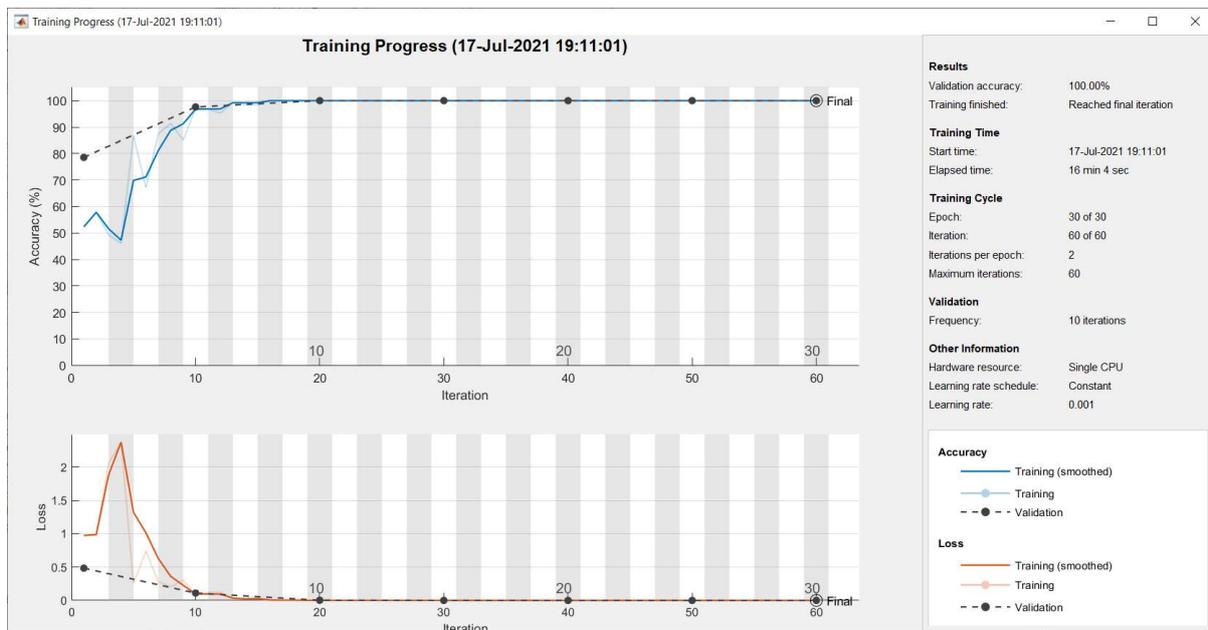


S8: Confusion matrix for 3:2 ratio on training and test set

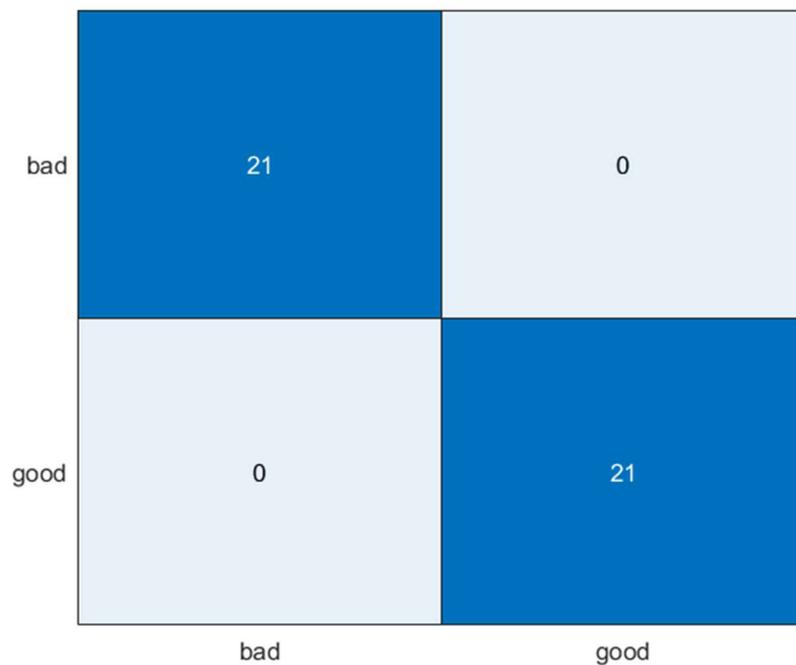


Appendix T: DCNN results for normalized FFT of 15 °C thermal perturbation test

T1: Training progress for 9:1 ratio on training and test set



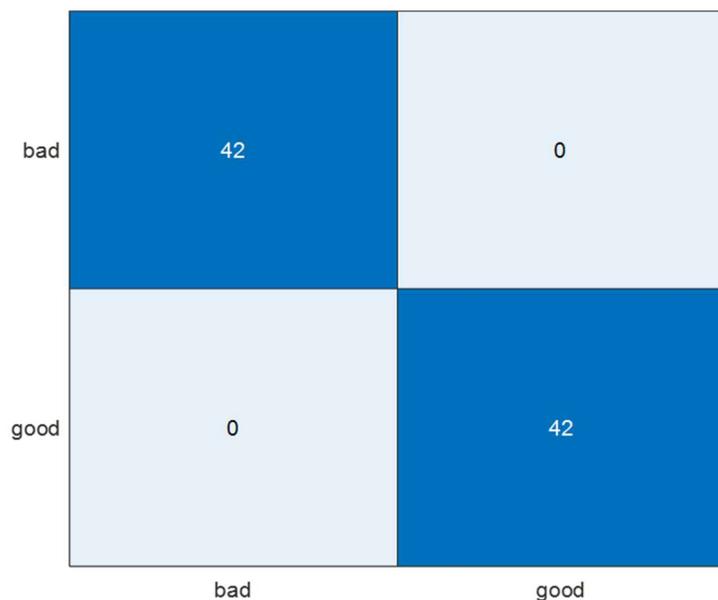
T2: Confusion matrix for 9:1 ratio on training and test set



T3: Training progress for 4:1 ratio on training and test set



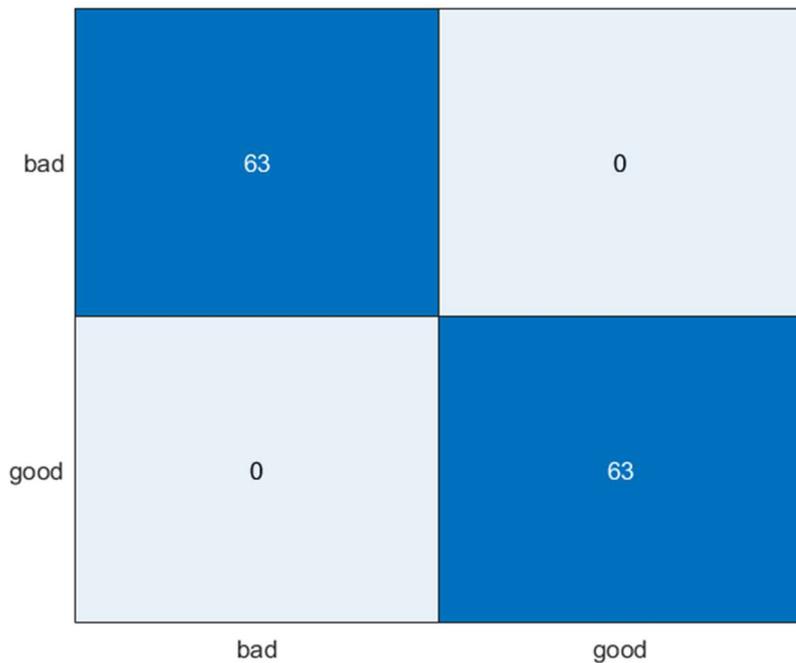
T4: Confusion matrix for 4:1 ratio on training and test set



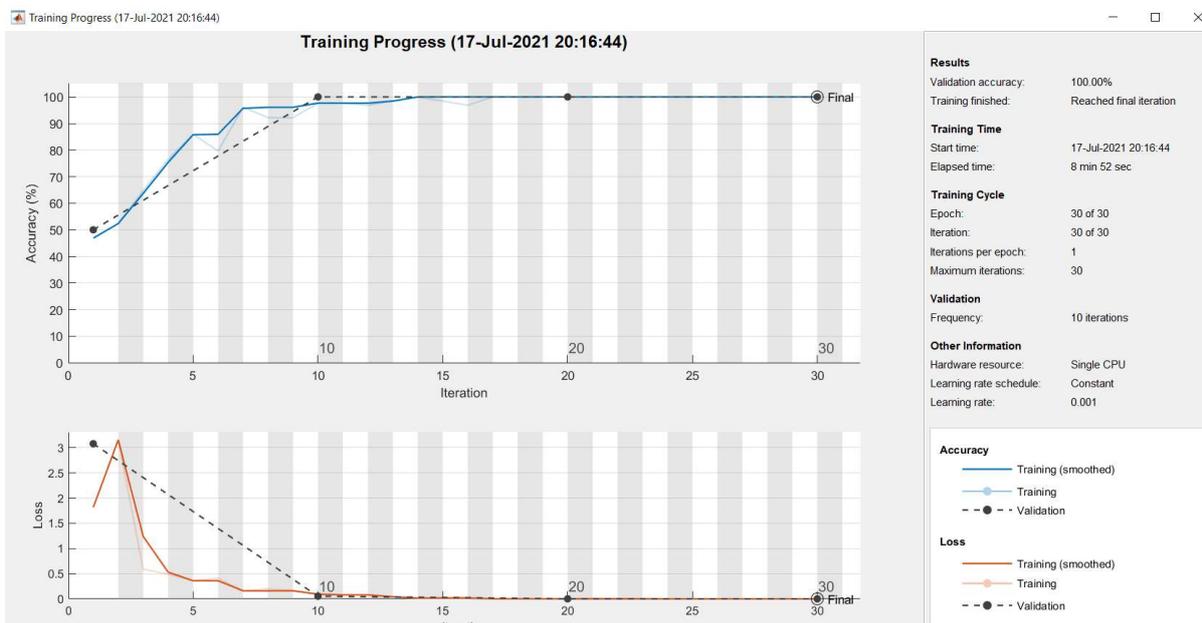
T5: Training progress for 7:3 ratio on training and test set



T6: Confusion matrix for 7:3 ratio on training and test set



T7: Training progress for 3:2 ratio on training and test set



T8: Confusion matrix for 3:2 ratio on training and test set



Appendix U: DCNN results for normalized FFT of 20 °C thermal perturbation test

U1: Training progress for 9:1 ratio on training and test set



U2: Confusion matrix for 9:1 ratio on training and test set



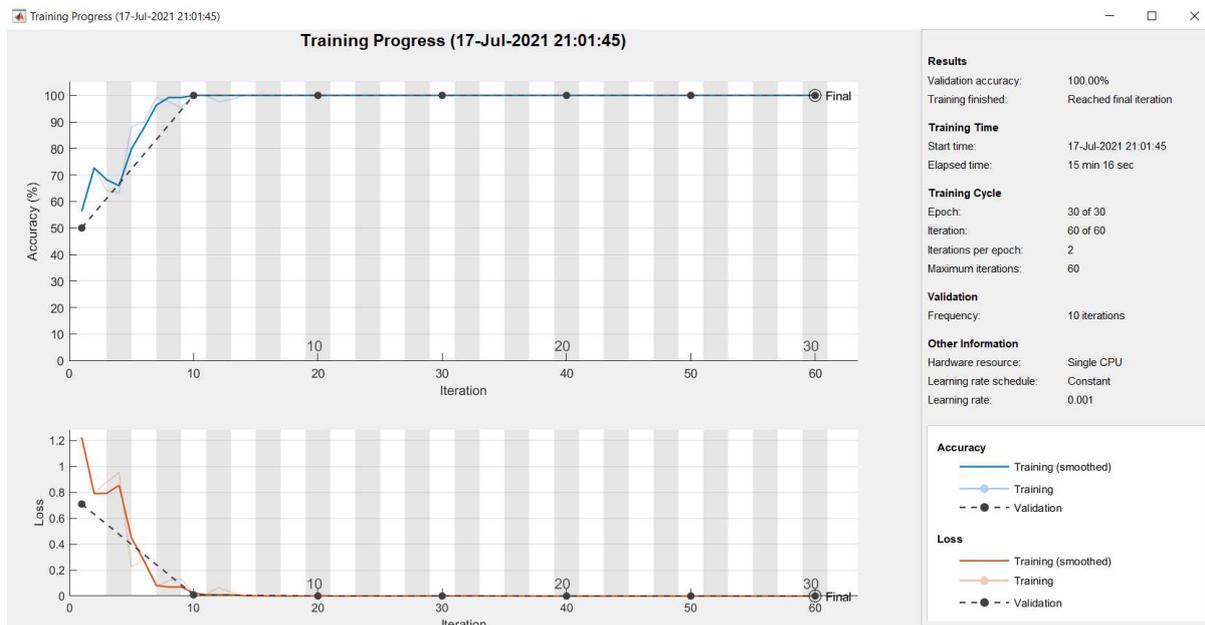
U3: Training progress for 4:1 ratio on training and test set



U4: Confusion matrix for 4:1 ratio on training and test set



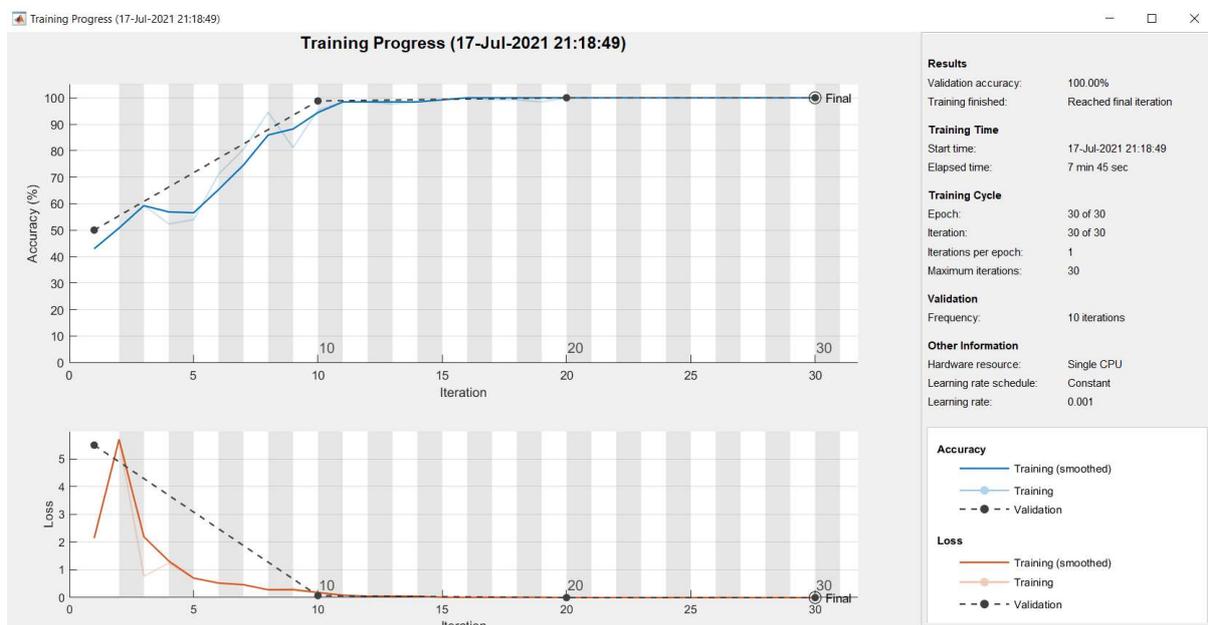
U5: Training progress for 7:3 ratio on training and test set

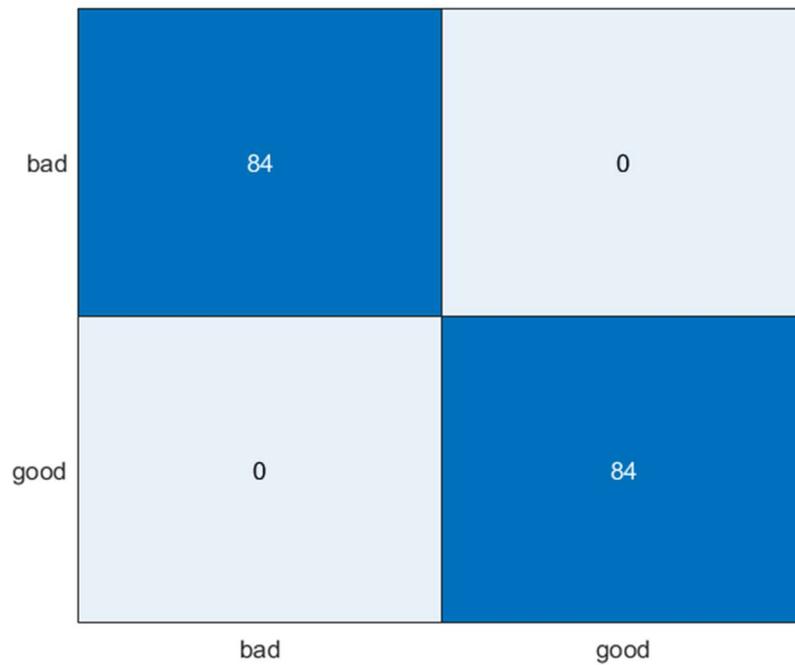
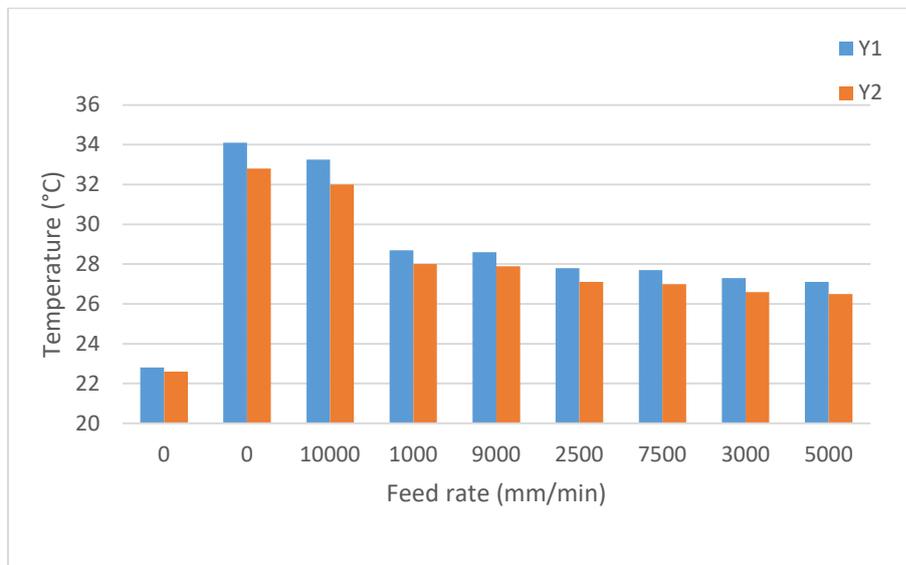


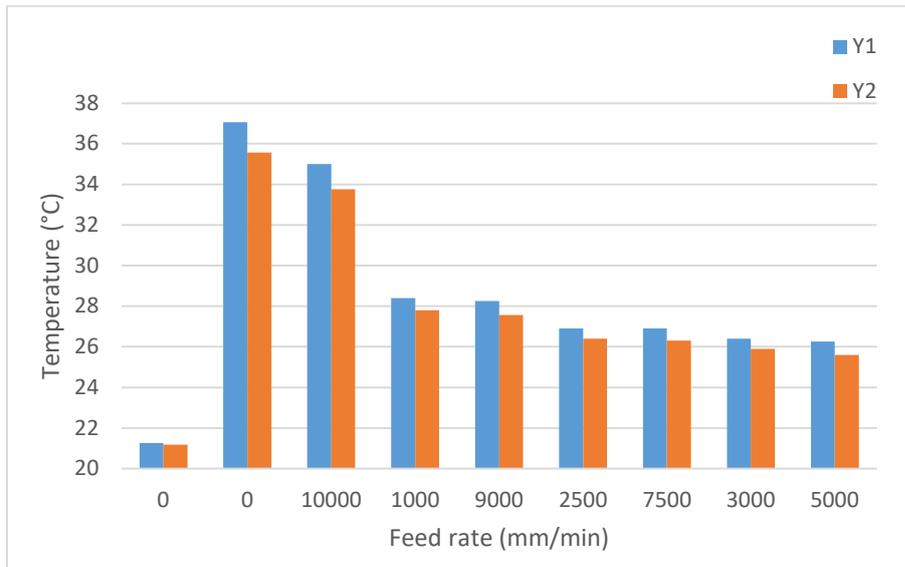
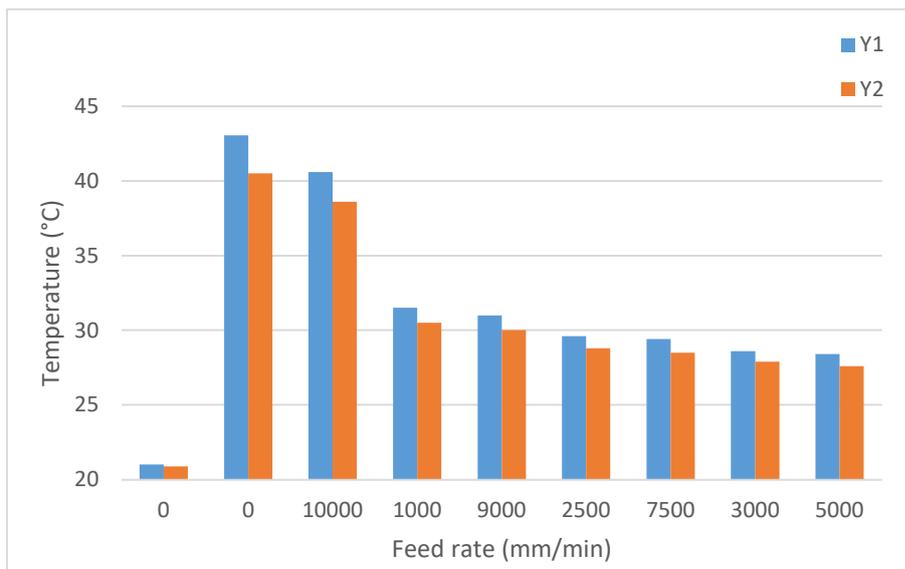
U6: Confusion matrix for 7:3 ratio on training and test set



U7: Training progress for 3:2 ratio on training and test set



U8: Confusion matrix for 3:2 ratio on training and test set**Appendix V: Temperature chart for thermal perturbation test****V1: Temperature chart for 10 °C thermal perturbation test**

V2: Temperature chart for 15 °C thermal perturbation test**V3: Temperature chart for 20 °C thermal perturbation test**

V4: Temperature chart for 25 °C thermal perturbation test