

**EARLY DETECTION OF IMBALANCE IN LOAD AND MACHINE
IN FRONT LOAD WASHING MACHINES BY MONITORING
DRUM MOVEMENT**

by
HAMED MOHAMMADI

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabanci University
December 2020

**EARLY DETECTION OF IMBALANCE IN LOAD AND MACHINE
IN FRONT LOAD WASHING MACHINES BY MONITORING
DRUM MOVEMENT**

Approved by:

[Redacted signature area]

[Redacted signature area]

[Redacted signature area]

Date of Approval: December 24, 2020

HAMED MOHAMMADI 2020 ©

All Rights Reserved

ABSTRACT

EARLY DETECTION OF IMBALANCE IN LOAD AND MACHINE IN FRONT
LOAD WASHING MACHINES BY MONITORING DRUM MOVEMENT

HAMED MOHAMMADI

COMPUTER SCIENCE AND ENGINEERING M.S. THESIS, DECEMBER 2020

Thesis Supervisor: Asst. Prof. Dr. Öznur Taştan Okan

Keywords: imbalance detection, artificial intelligence, condition monitoring,
machine learning, predictive maintenance, vibration measurement, washing
machines

Balance issues in washing machines manifest themselves in the form of vibrations. These unwanted vibrations become more prominent at high spin speeds. They can be detrimental to the machine's performance and shorten lifespan by causing permanent physical damage. Detecting these vibrations early in the wash cycle and at spin speeds below the machine's resonant frequency is critical in devising proper measures to alleviate their effects. In this thesis, we focus on the two common balance issues observed in washing machines. The first one is machine imbalance, which stems from the improper adjustment of leveling legs. The second balance problem is the load imbalance, which is the result of an uneven distribution of the load inside the drum. We specifically investigate the possibility of detecting these imbalances as early as possible using models trained on sensory data collected from the drum. For this aim, we collect vibration data on the two types of imbalance scenarios throughout the wash cycle. Using these data, we build supervised classification models using different feature extraction techniques on the multivariate times series data and different machine learning models. We compare models that are trained with different partial data collected at different time segments early in the wash cycle. Our results show that we can attain a 95% F1-score with input as short as 500 ms of the wash cycle, indicating that early prediction of these two imbalances during the wash cycle is possible. The collected data are shared for the research community.

ÖZET

ÖN YÜKLÜ ÇAMAŞIR MAKİNELERİNDE TAMBUR HAREKETİNİN İZLENMESİYLE YÜK VE MAKİNEDE DENGESİZLİĞİN ERKEN TESPİTİ

HAMED MOHAMMADI

BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ YÜKSEK LİSANS TEZİ, ARALIK
2020

Tez Danışmanı: Dr. Öğr. Üyesi Öznur Taştan Okan

Anahtar Kelimeler: dengesizlik tespiti, yapay zeka, durum izleme, makine öğrenimi, öngörücü bakım, titreşim ölçümü, çamaşır makinesi

Çamaşır makinelerinde denge sorunları titreşim şeklinde kendini gösterir. Yüksek dönüş hızlarında daha belirgin hale gelen bu titreşimler, kalıcı fiziksel hasara neden olarak makinenin performansına ve ömrünü olumsuz şekilde etkileyebilir. Bu titreşimleri yıkama döngüsünün başlarında ve makinenin rezonans frekansının altında tespit edilmesi, uygun önlemlerin belirlenmesi için kritik öneme sahiptir. Bu tezde, çamaşır makinelerinde görülen iki genel denge sorununa odaklanıyoruz. İlki, makinenin dengesizliği, tesviye ayaklarının yanlış ayarlanması kaynaklanan makine dengesizliği; ikincisi ise tambur içindeki yükün eşit olmamasından kaynaklanan yük dengesizliği. Tamburdan toplanan sensör verileri ile eğitilmiş yapay öğrenme modelleri kullanarak özellikle bu dengesizliklerin olabildiğince erken tespit edilme imkanını araştırmaktayız. Bu amaçla, iki tür dengesizlik senaryosuna ilişkin veri toplamaktayız. Bu ardışık verilerden, farklı özellik çıkarma teknikleri ve farklı makine öğrenimi modelleri kullanarak denetimli ardışık veri sınıflandırma modelleri oluşturmaktayız. Yıkama döngüsünün farklı zamanlarından toplanan kısmi sensör verisi ile kurulan modelleri karşılaştırmaktayız. Sonuçlarımız, yıkama döngüsünün 500 ms'sinden toplanan veri ile %95 F1 skoruna ulaşabildiğini göstermektedir, bu da yıkama döngüsü sırasında bu iki dengesizliğin erken tespitinin mümkün olduğunu işaret eder. Toplanan veriler, araştırmacıların erişimine sunulmuştur.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Dr. Öznur Taştan Okan for her continuous guidance, valuable feedback, and constant support and understanding that helped me conduct this study. I also want to thank Dr. Ozan Biçen for his valuable feedback.

I want to give my special thanks to my wife and my daughter for always being there for me during the hardest of times and giving me hope and motivation.

I also would like to thank Arçelik A.Ş., and specially my team leader Mr. Çağatay Büyüktopçu, for giving me the chance to continue my studies, and providing me with the time and equipment to undertake this research.

To My Beloved Wife & Dearest Daughter

TABLE OF CONTENTS

LIST OF TABLES	x
LIST OF FIGURES	xi
1. INTRODUCTION	1
2. LITERATURE REVIEW	3
2.1. Imbalance Detection Algorithms.....	3
2.1.1. Threshold-based Detection and Linear Modeling	3
2.1.2. Machine Learning and Neural Networks	4
2.2. Data Types Used in Imbalance Detection Algorithms	5
2.2.1. Mechanical Data	5
2.2.2. Motor Data	6
2.2.3. Hybrid Data.....	6
2.3. Feature Extraction	6
2.4. Early Detection in Sequential Data	7
3. DATA COLLECTION AND PREPROCESSING	8
3.1. Data Collection Setup	8
3.1.1. Unbalanced Machine	10
3.1.2. Unbalanced Load.....	11
3.2. Preprocessing	12
3.2.1. Reconstruction and Resampling.....	12
3.2.2. Trimming.....	12
3.2.3. Feature Extraction	13
3.2.3.1. Extrema	14
3.2.3.2. Mean and Variance	15
3.2.3.3. Kurtosis and Crest Factor	15
3.2.3.4. Histogram	16
3.2.3.5. Uniform Manifold Approximation and Projection....	17
3.2.3.6. Frequency Spectrum.....	18

3.2.3.7. Wasserstein Time Series Kernel	19
4. MODEL SELECTION AND TRAINING	21
4.1. Models	21
4.1.1. Support Vector Machines	21
4.1.2. Gradient Boosted Decision Trees.....	22
4.1.3. Neural Networks	22
4.1.3.1. Convolutional Neural Networks	23
4.1.3.2. Long Short-Term Memory.....	23
4.2. Hyperparameter Tuning	24
4.3. Performance Metrics	24
4.3.1. Precision	25
4.3.2. Recall	26
4.3.3. F_1 Score.....	26
4.3.4. Average Precision	26
4.4. Earliness	27
5. RESULTS.....	28
6. CONCLUSION	33
BIBLIOGRAPHY.....	35
APPENDIX A	37
APPENDIX B	42
APPENDIX C	47

LIST OF TABLES

Table 5.1. Average precision test score of best-performing models with respect to sample lengths for machine imbalance detection with difficult cases	29
Table 5.2. F_1 test score of best-performing models with respect to sample lengths for load imbalance detection with difficult cases.....	29
Table 5.3. Average precision test score of best-performing models with respect to sample lengths for machine imbalance detection with noisy data	30
Table 5.4. F_1 test score of best-performing models with respect to sample lengths for load imbalance detection with noisy data.....	30
Table A.1. Average precision test score of different models with respect to sample lengths for machine imbalance detection.....	37
Table A.2. F_1 test score of different models with respect to sample lengths for load imbalance detection	38
Table C.1. Hyperparameters for best-performing machine imbalance detection models - SVM models as (C, γ) , and XGBoost models as $(\text{depth}_{max}, w_{min}, \gamma)$	47
Table C.2. Hyperparameters for best-performing load imbalance classification models - SVM models as (C, γ) , and XGBoost models as $(\text{depth}_{max}, w_{min}, \gamma)$	47

LIST OF FIGURES

Figure 3.1. Data collection sensor board (a) mounted on top of the concrete block (b) attached to the washing machine drum (c)	9
Figure 3.2. The x -, y -, and z -axis of accelerometer data from (a) the daily program with balanced machine, and (b) the drum rotating at 100 RPM with balanced load	10
Figure 3.3. Different load combinations attached to the drum for load imbalance data collection	11
Figure 3.4. Data pipeline from washing machine to machine learning algorithm	12
Figure 3.5. Generating DFT representation from 5 seconds of sensor data from the drum rotating at 100 RPM	14
Figure 3.6. Kurtosis (K) and crest factor (C) values of three sample functions	16
Figure 3.7. Histogram comparison of x -, y -, and z -axis of accelerometer data from normal wash with balanced machine (top) and unbalanced machine (bottom).....	17
Figure 3.8. Raw data and DFT of x -axis of (a) accelerometer and (b) gyroscope data from a balanced machine during normal wash	18
Figure 3.9. Comparing DFT of x -axis of gyroscope data from drum rotating at 100 RPM with balanced load and 350 grams of unbalanced load	19
Figure 4.1. CNN model for load imbalance detection and classification with 1 second of input data.....	23
Figure 4.2. Grid search results with SVM model for γ and C parameters with 10-bin histogram as feature vector and samples of length 2 seconds	25
Figure 5.1. Comparison of performance of the models with difficult cases and noisy training data for machine imbalance detection task	31
Figure 5.2. Comparison of performance of the models with difficult cases and noisy training data for load imbalance detection task.....	31

Figure A.1. Average precision of SVM models on test data to detect machine imbalance with different input lengths	39
Figure A.2. Average precision of XGBoost models on test data to detect machine imbalance with different input lengths	39
Figure A.3. F_1 score of neural network models on test data to detect load imbalance with different input lengths.....	40
Figure A.4. F_1 score of SVM models on test data to detect load imbalance with different input lengths.....	40
Figure A.5. F_1 score of XGBoost models on test data to detect load imbalance with different input lengths	41
Figure B.1. Confusion matrix of XGBoost model with DFT features to detect machine imbalance with input length of 0.5 second	42
Figure B.2. Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 1 second ..	42
Figure B.3. Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 2 seconds .	42
Figure B.4. Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 5 seconds .	43
Figure B.5. Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 10 seconds	43
Figure B.6. Confusion matrix of SVM model with mean and variance features to detect machine imbalance with input length of 20 seconds ...	43
Figure B.7. Confusion matrix of SVM model with mean and variance features to detect machine imbalance with input length of 30 seconds ...	44
Figure B.8. Confusion matrix of XGBoost model with DFT features to detect load imbalance with input length of 0.5 second	44
Figure B.9. Confusion matrix of XGBoost model with DFT features to detect load imbalance with input length of 1 second	44
Figure B.10. Confusion matrix of SVM model with UMAP features to detect load imbalance with input length of 2 seconds.....	45
Figure B.11. Confusion matrix of SVM model with 10-bin histogram features to detect load imbalance with input length of 5 seconds	45
Figure B.12. Confusion matrix of SVM model with 10-bin histogram features to detect load imbalance with input length of 10 seconds	45
Figure B.13. Confusion matrix of SVM model with 20-bin histogram features to detect load imbalance with input length of 20 seconds	46
Figure B.14. Confusion matrix of SVM model with 10-bin histogram features to detect load imbalance with input length of 30 seconds	46

1. INTRODUCTION

Washing machines are among the most essential durable goods found in every modern household. Front-loading (or horizontal-axis) washing machines are becoming more popular due to their energy efficiency and considerably less water consumption compared to their top-loading counterparts (Ramasubramanian & Tiruthani, 2009). These machines are produced to be durable and last for many years. However, minor anomalies in their working conditions, such as imbalances and improper belt tension, that happen repeatedly over a prolonged duration of usage can adversely affect their performance and lifespan (Yörükoğlu & Altuğ, 2012).

Among the imbalance issues, some are caused by the incorrect installation of the machine, while others can be the result of an undesirable distribution pattern of the laundry items put inside the drum. The two most common types of imbalances that can be prevented, or their consequences can be alleviated, are machine imbalance and load imbalance. Machine imbalance is caused by improper adjustment of the leveling legs of the washing machine during installation, while the load imbalance is the result of an unbalanced distribution of the laundry item inside the drum at high spin speeds.

In order to get a grasp of the extent of damage that these imbalances can cause, let us assume a washing machine drum rotating at 1200 rounds per minute (RPM) with 1 kilogram of laundry tangled and concentrated on one side of it. The centrifugal force produced by this laundry can be calculated using the equation $\vec{F} = m\omega^2r$, in which m is the mass of the load, ω is its angular velocity, and r is the radius of the drum. Assuming the radius to be 25 centimeters, the force exerted by this unbalanced load on the drum is 3948 N, or 402.6 kilograms perpendicular to the drum. Although the spring and damper mechanism in modern washing machines can absorb some of this force, it still can permanently damage the drum - and the washing machine - if the spin is not interrupted immediately. While certain precautions can prevent the formation of these imbalances, the user might be unaware of them. In this thesis, we investigate the possibility of detecting them early in the wash cycle so that required measures can be taken and physical damage to the machine can be avoided.

Most of the previous work on imbalance detection in washing machines focus on load imbalance detection, and formulate it as a classification problem (Lee & Kim, 2010; Murray, Henderson, Marcetic, Marcinkiewicz, Sadasivam & Rajarathnam, 2011; Ramasubramanian & Tiruthani, 2009; Yörükoğlu & Altuğ, 2012; Yuan, 2008; Zhang, Xie, Garstecki, Xie, Slabbekoorn & Buendia, 2011). The main limitation with most of the proposed methods, however, is that they either rely on complex mathematical models of the machine or use data from multiple sensors located at different positions in the machine to achieve the task. The former approach requires extensive expertise and experience in the physics and design of the machine, while the latter imposes extra production costs. In addition, to the best of our knowledge, there is no previous work that focuses on the early detection of these balance issues.

In this thesis, we address these problems by using an inexpensive inertial measurement unit (IMU) attached to the drum of a washing machine and utilize the data obtained from it to detect both load imbalance and machine imbalance as early as possible in the wash cycle. We also aim to classify the amount of unbalanced load when load imbalance is detected. The developed method does not require any mathematical modeling of the machine, nor any prior knowledge and expertise in its design and physical properties. Another contribution of this work is that we collect data and make it available for the other researchers working on similar problems ¹.

This thesis is organized as follows. In Chapter 2, we present an overview of relevant literature and review different methods for data collection, feature extraction, and imbalance detection. Chapter 3 explains the data collection set up, and preprocessing steps to be used for the classification algorithms. In Chapter 4, we talk about the different models used for imbalance detection and load imbalance classification. Chapter 5 shows the results obtained from all the implemented models, together with the best models selected for each imbalance detection task. Finally, Chapter 6 concludes the thesis, and future research directions in this area are discussed.

¹To download the datasets, please visit <https://bit.ly/3i9ST5W>

2. LITERATURE REVIEW

In this chapter, we discuss different approaches to solving the problem of imbalance detection in washing machines among researchers. First, we review imbalance detection algorithms with a focus on industrial applications. We then discuss different data collection strategies used in previous works. We continue by examining different early detection approaches. In the end, we go over the different feature extraction methods that are used to address this family of problems.

2.1 Imbalance Detection Algorithms

Algorithms used from imbalance detection range from the most basic threshold-based decision making, to the more sophisticated neural network approaches. These methods can be classified as follows:

2.1.1 Threshold-based Detection and Linear Modeling

Many of the older solutions to the problem of imbalance detection using vibrations and displacement rely on a threshold or a defined linear relation over the data read from the sensors for decision making. Unless proper precautions are taken, this type of approach is susceptible to noise and measurement errors.

Lee & Kim (2010) use this approach and classify the position of the unbalanced load inside the drum into the front, center, rear, and diagonal. They monitor the vibration amount of the drum at different positions using acceleration sensors, as well as the spin speed of the drum. The phase differences between the vibration

signals are then obtained and divided by the spin speed to obtain a value. This value is then compared to a reference value to detect the position of the unbalanced load.

Ramasubramanian & Tiruthani (2009) derive a mathematical model of the machine and use it to associate the amplitude of drum movements with the mass of the unbalanced load. They use a simplified model of the drum as a one-dimensional spring-mass system and use this model to predict the movement of the drum with unbalanced load at different locations. The predictions are then compared with the real data collected from the drum using a custom capacitive displacement sensor to detect the location of the unbalanced load.

In another work, Murray et al. (2011) use a modeling approach with data obtained from the motor to determine unbalanced mass. They construct a mathematical model of the machine. The torque and speed information of the motor is monitored and the amount of ripple in these variables is used to calculate the mass of the unbalanced load inside the drum.

2.1.2 Machine Learning and Neural Networks

More recent works conducted in the area of imbalance detection, or more broadly anomaly detection, use machine learning models to detect different types of anomalies in rotary machinery, and more specifically washing machines. These methods are more robust to noise and measurement errors and produce better results compared to the ones discussed in Section 2.1.1.

Xing, Pei & Philip (2009) use a 1-nearest neighbor (1NN) model to perform early detection on time series. They define minimum prediction length as the shortest length of the input sequence for which the classifier predicts the same label as the full-length input, and the predicted label does not change as longer data sequences are provided to the model.

Kadous (1999) extracts events from the training data using parametrized event primitives (PEPs) and combines them with features such as global maxima and minima to use them for classification. They use naïve Bayes or C4.5 (Quinlan, 1993) as the learner, and k-means to cluster the data.

Yuan (2008) implements a support vector machine and a neural network model to estimate the mass and position of the unbalanced load and compare their perfor-

mance. Principal component analysis (PCA) is used for dimensionality reduction of the data collected from multiple sensors attached at different locations of the machine, and the obtained features are fed to the neural network and support vector machine to classify the input data.

Yörükoğlu & Altuğ (2012) use a fuzzy neural network (FNN) to estimate the mass and location of the unbalanced load inside the drum. They use the collected data to develop the rules and membership functions for a fuzzy logic-based estimator and tune it experimentally. This estimator is then fed with the data collected from multiple sensors mounted on the drum to predict the mass and position of the unbalanced load.

2.2 Data Types Used in Imbalance Detection Algorithms

The type of data collected from the machine plays an important role in the performance of the imbalance detection algorithms. There are three general approaches in data collection for imbalance detection in washing machines. These approaches are as follows:

2.2.1 Mechanical Data

The first approach consists of measuring mechanical data, such as vibration and displacement of the drum, from the machine and using them to detect the imbalance. This method is used by Yuan (2008), Yörükoğlu & Altuğ (2012), and Ramasubramanian & Tiruthani (2009) to detect the imbalance and estimate the mass and location of the unbalanced load. Yuan (2008) implements a multi-sensor solution composed of two laser sensors and an accelerometer to estimate the mass of the unbalanced load. Yörükoğlu & Altuğ (2012) achieve the same goal by using two accelerometers and a Hall effect sensor. In another study, Ramasubramanian & Tiruthani (2009) develops a capacitive displacement sensor to measure drum movements along a single axis and use the data to estimate the unbalanced load.

2.2.2 Motor Data

In another data collection approach, motor-related data such as spin speed, torque, current, and power are used to detect balance issues in the washing machine. Murray et al. (2011) use motor torque ripples and motor speed and feed them into a digital signal processing (DSP) unit to carry out the unbalanced load detection task. Zhang et al. (2011) determine the load imbalance condition by operating the motor in three different speed profiles, i.e., constant speed, acceleration, and deceleration, and measuring the average power output from the motor during these phases. This value is then used to calculate the power fluctuation integral, which in turn is used to detect the amount of unbalanced load.

2.2.3 Hybrid Data

There has been previous work that combines the two approaches mentioned in Section 2.2.1 and Section 2.2.2 into a hybrid one. This approach uses both mechanical and motor data to detect the imbalance in load. For example, Lee & Kim (2010) classify the type of load imbalance in the washing machines by using a threshold-based method. The authors use the data from a multi-axis acceleration sensor, along with the fluctuations in motor spin speed subjected to unbalanced loads, to detect unbalanced load in the machine.

2.3 Feature Extraction

Extraction of meaningful features from the raw data is required to prepare the input data for machine learning models. Xing, Pei, Yu & Wang (2011) propose a method to extract local shapelets from the signal to manifest a target class in a distinct manner. In another work, Baydogan, Runger & Tuv (2013) use a bag-of-features representation of the signal by choosing subsequences of arbitrary length from random locations in the signal and dividing them into shorter partitions to capture the local information. A more recent work deploys a kernel, namely Wasserstein time series kernel (WTK), to measure the similarity between two subsequence distributions (Bock, Togninalli, Ghisu, Gumbsch, Rieck & Borgwardt, 2019). To detect

the bearing fault in rotating machinery, Janssens, Slavkovikj, Vervisch, Stockman, Loccufier, Verstockt, Van de Walle & Van Hoecke (2016) use signal shape representatives such as root mean square (RMS), Kurtosis, and Crest Factor as the features for the machine learning algorithms. Multi-scale fractal dimension (MFD) and Mel Frequency Cepstral Coefficients are among the other feature extraction methods that are used to classify anomalies in multivariate time series (MTS) (Nelwamondo, Marwala & Mahola, 2006).

2.4 Early Detection in Sequential Data

Early classification of temporal data has been addressed as a general machine learning problem in various previous studies (Alonso González & Diez, 2004; Anderson, Parrish & Gupta, 2012; Dachraoui, Bondu & Cornuejols, 2013; Ghalwash, Radosavljevic & Obradovic, 2014). The problem is to make a decision with partial temporal input. Xing et al. (2009) uses a 1-nearest neighbor classifier to achieve reliable predictions with minimal temporal input lengths. They do not make any assumptions about the form of the underlying distributions on the input. In another work, the cost of deferring decision is incorporated in the cost function and early prediction is attained by extracting local patterns called multivariate shapelets and classifying the time series by probing the earliest pattern that is closest to the training shapelets (Ghalwash & Obradovic, 2012). A similar approach is used by He, Duan, Peng, Jing, Qian & Wang (2015) to extract distinctive shapelets from a multivariate time series, and use methods such as query by committee (QBC) to classify the samples. Achenchabe, Bondu, Cornuéjols & Dachraoui (2020) introduce an optimization criterion by considering misclassification and decision postponement costs and use it in algorithms that seek to predict future information gain by considering the waiting cost. In another work, Hatami & Chira (2013) put forward a classifier structure including a reject option. This architecture is able to make online decisions without waiting for the entire length of the input data.

In the existing literature, the aim is to decide whether to make a classification at a given time or to defer the prediction at a later time step. Thus, the previous work is an online learning setup. In our work, we conduct a comparative study to determine the sufficient portion of the data to attain a good prediction error. However, we do not formulate the problem as an online learning problem.

3. DATA COLLECTION AND PREPROCESSING

In this chapter, we first describe the data collection procedure from washing machines for machine imbalance and load imbalance prediction. Next, we describe the steps for processing the raw data and prepare it for the imbalance detection models.

We collected the data using a sensor board attached to the concrete block on the drum of a washing machine. This board encapsulates an inertial measurement unit, including an accelerometer and a gyroscope sensor, and transfers the collected data to a computer where the data is stored, and later used for model training and testing.

3.1 Data Collection Setup

To be able to detect the presence of machine imbalance and load imbalance, we use two different strategies to collect the required data. The reason for this is that in the machine imbalance detection problem, the earliest drum spin is the one at 52 RPM during the first washing cycle, which happens right after the washing machine takes the required amount of water. However, this rotation speed is not enough for detecting the load imbalance. In order to detect this type of imbalance, the drum needs to rotate at a speed high enough to cause centrifugal forces to push the laundry to the drum and make them stay there, but lower than the resonant frequency of the machine to avoid extreme movements and physical damage to the machine. This speed, known as satellization speed, is the angular velocity at which the velocity of the drum and the laundry become equal ($\omega_{drum} = \omega_{laundry}$). Satellization speed can be calculated by mathematically modeling the machine and the laundry inside it (Janke, Richmond & Zasowski, 2015). For the purpose of this work, we take the satellization speed to be 100 RPM. As a result, the data required for detecting machine imbalance was collected during a normal 30-minute wash cycle, while load imbalance data was collected with the drum rotating at a fixed speed of 100 RPM.

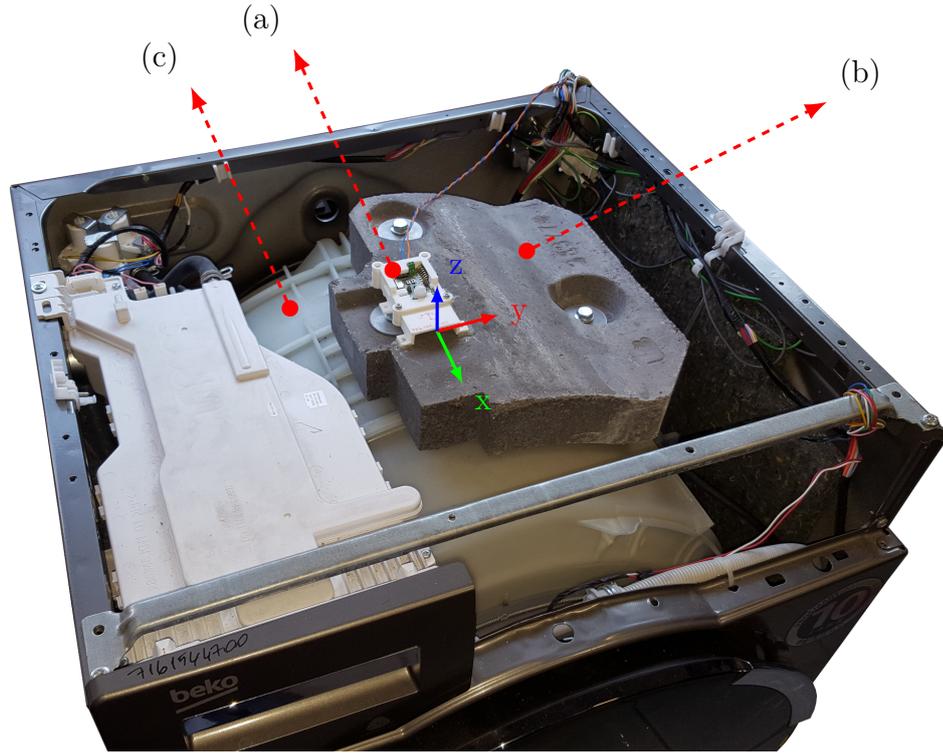


Figure 3.1 Data collection sensor board (a) mounted on top of the concrete block (b) attached to the washing machine drum (c)

We collect the data using the setup shown in Figure 3.1. In this setup, a sensor board with a gyroscope and an accelerometer is fixed on the concrete block attached to the drum of a washing machine. The sensors measure the linear acceleration and angular velocity of the drum along and around x , y , and z axes, respectively. The board then transfers the measurements to a computer over WiFi, where the data is recorded. Figure 3.2 illustrates some samples of the raw data collected from the accelerometer during a full wash cycle, and with the drum rotating at 100 RPM.

In order to choose the sampling rate for the sensors, we use the Nyquist-Shannon Theorem which states that a digitally sampled signal can be fully reconstructed if the sampling frequency is at least twice as large as the highest frequency component in the signal (Shannon, 1949). The highest selectable spin speed of the machine that is used for data collection is 1000 RPM (16.67 Hz). As a result, a sampling frequency of 33.3 Hz is sufficient to reconstruct the signal. However, since the lowest sampling frequency that the sensor board supports is 50 Hz, this rate was chosen to collect the data from the two sensors. As the detection needs to happen in the early stages of the wash cycle and even below the machine's resonant frequency, the chosen sampling rate is well above the minimum required frequency to fully reconstruct the signal.

The data is collected for both the normal and unbalanced behavior of the machine.

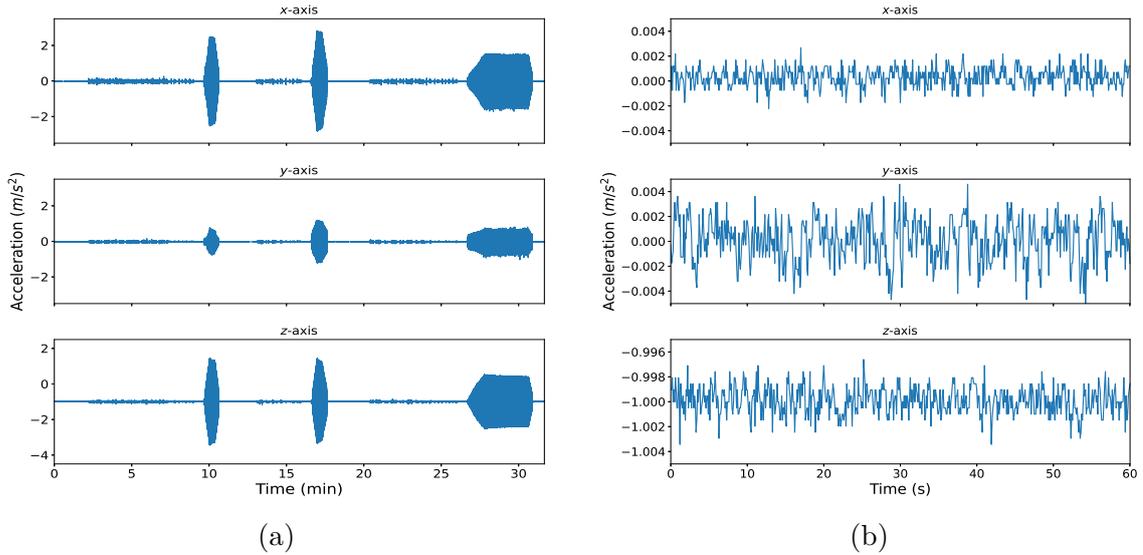


Figure 3.2 The x -, y -, and z -axis of accelerometer data from (a) the daily program with balanced machine, and (b) the drum rotating at 100 RPM with balanced load

The data for the normal behavior of the machine is collected by operating it under normal conditions, i.e. fully balanced for machine imbalance data and with balanced loads for load imbalance data. In order to collect the abnormal data, we inflicted the two types of imbalances on the machine separately. We set up experiments that simulate these imbalance cases as follows:

3.1.1 Unbalanced Machine

To produce this type of imbalance, the machine was perfectly leveled, and then one of the leveling legs of the machine was deliberately made shorter than the others. By doing this, the washing machine could swing by as much as 0.3° along its diagonal axis. The machine was then run by choosing the *Daily Wash* 30-minute program with the following load and spin speed combinations: laundry load of 2.5 and 5 kilograms, and rinse spin speeds of 400, 600, 800, and 1000 RPM. Ten different samples were taken from each run, resulting in a total of 80 samples labeled as abnormal. The same procedure was repeated with the fully balanced machine and 80 samples were recorded and labeled as normal runs.

3.1.2 Unbalanced Load



Figure 3.3 Different load combinations attached to the drum for load imbalance data collection

Since the behavior of the laundry inside the drum cannot be reliably controlled, we simulated this type of imbalance using custom-sized metal loads. As Figure 3.3 illustrates, these loads are evenly distributed among the three regions between the drum paddles, with one of them being heavier than the others to produce the imbalance. We collected data using the following balanced and unbalanced load combinations: 0, 1200, 1950, and 2550 grams of balanced, and 0, 350, 650, and 1000 grams of an unbalanced load. The samples with no unbalanced load are labeled as normal, and the rest as imbalanced load. Data labels also include the amount of unbalanced load, with 1, 2, and 3 representing 350, 650, and 1000 grams of such load, respectively. The machine was operated at 100 RPM with each of these load configurations, and one hour of data was collected at each run, resulting in a total of 16 hours of load imbalance data.

3.2 Preprocessing

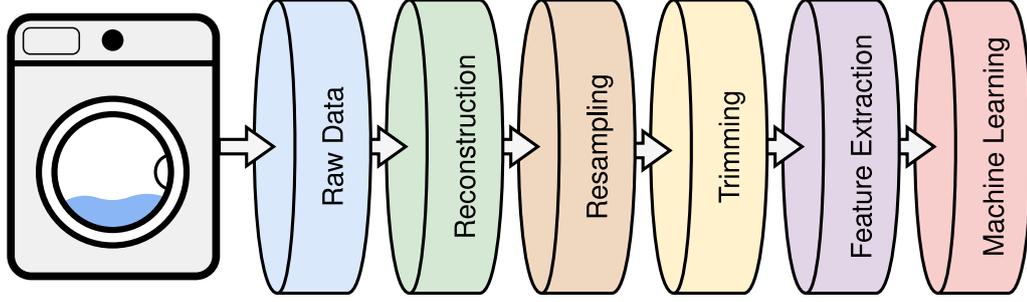


Figure 3.4 Data pipeline from washing machine to machine learning algorithm

The data collected from the sensors is not ready to be used for the purpose of imbalance detection and classification. To prepare the data, we process the sensory data from the inertial measurement unit as shown in Figure 3.4. These steps are described in detail as follows:

3.2.1 Reconstruction and Resampling

Due to the internal working mechanism of the used sensor board and the delay caused by wireless communication, some of the data points obtained from the two sensors were not sampled at exactly the same time, and their timestamps were not matching. To overcome this problem, the signals from the accelerometer and gyroscope were reconstructed and resampled at the original sampling frequency of 50 Hz. The missing values were filled in using linear interpolation. Assuming we have two points (x_0, y_0) and (x_1, y_1) , to find the missing value y at a point x in the interval (x_0, x_1) we can use Equation 3.1.

$$(3.1) \quad y = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0}$$

All the data obtained for both imbalance detection tasks were reconstructed and resampled before other processing steps can be applied on them.

3.2.2 Trimming

The data collected from the *Daily Wash* program includes an idle part before the drum starts spinning, and another one after the program is finished and before the data collection is stopped, as shown in Figure 3.2(a). These parts of the data

do not have any significance in the detection of imbalances and can cause hidden confounding effects. To trim these parts, a moving window with a threshold was implemented. The window calculates the mean of all sample points inside it, and if the distance from the mean to either the minimum or maximum value inside the window exceeds a given threshold, we use the window location as the starting position of the signal. The same window moving backward from the end of the signal is used to detect its ending position. In this work, a window of size 25 and a threshold value of 0.1 is used to trim the samples. Since load imbalance data was collected at a fixed spin speed, it does not need trimming and can be directly used in the next step.

3.2.3 Feature Extraction

Extraction of meaningful features from the raw data is required to prepare the data for machine learning models. To do so, non-overlapping moving windows of predetermined sizes were implemented. The windows move across the data and extract the required features from within them, producing a single sample either for training or testing. Figure 3.5 shows a sample window of size 5 seconds extracting discrete Fourier transform features from the sensor data used for the load imbalance detection task. The strategy for selecting the size of the window will be discussed later in Section 4.4. Since the data collected during a full washing cycle include the high-speed spin cycles where the anomalous behavior becomes evident, only the parts during which the drum is rotating at 52 RPM were utilized for the detection task.

In order to get the best results from machine learning models, the extracted features need to be properly scaled. To do so, a min-max normalization strategy was used to scale the feature vectors to range $[0, 1]$. Equation 3.2 can be used to normalize a single point $x_i^{(j)}$ in the x_i component of a multivariate time series.

$$(3.2) \quad x_{i_{norm}}^{(j)} = \frac{x_i^{(j)} - \min(x_i)}{\max(x_i) - \min(x_i)}$$

In this work, we use the following feature extraction strategies, the obtained features are scaled, and the performance of the models using these extraction strategies are compared to decide on the best feature representation. We also devise several simple baselines to compare the methods against.

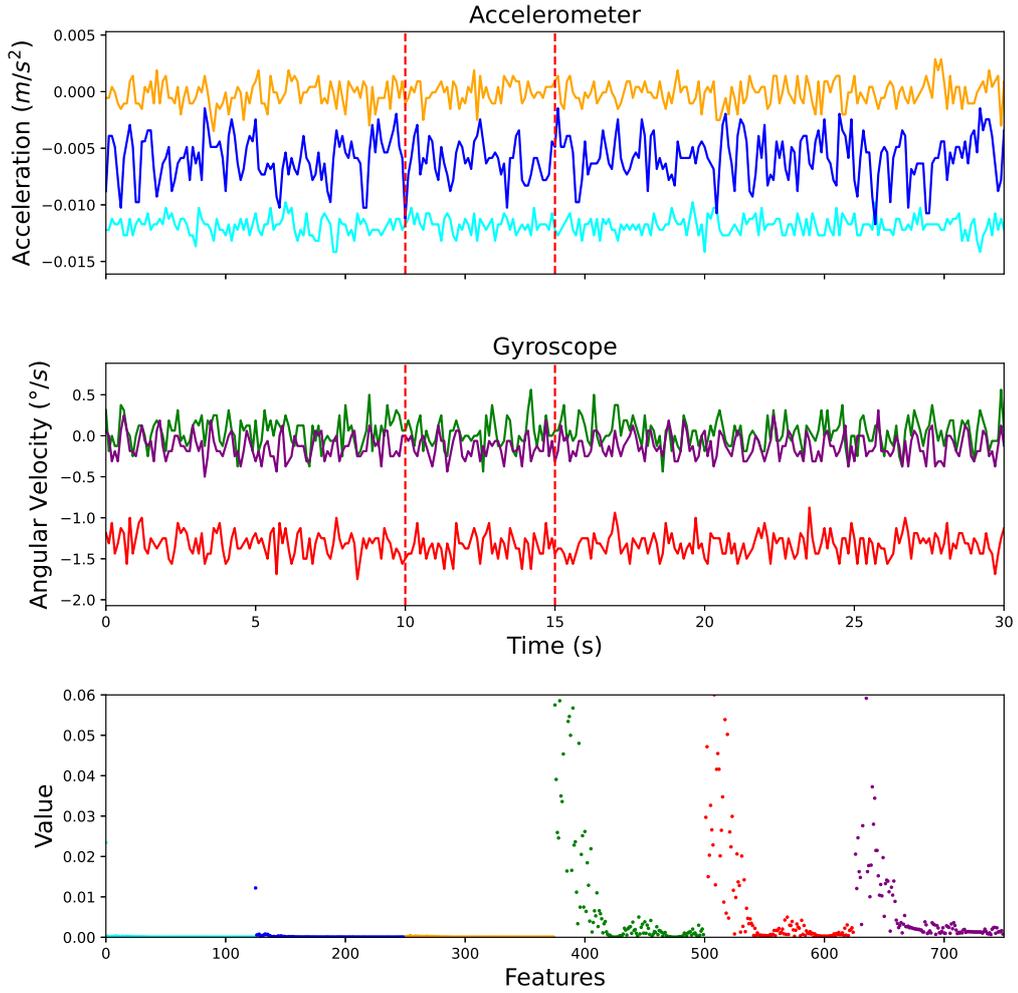


Figure 3.5 Generating DFT representation from 5 seconds of sensor data from the drum rotating at 100 RPM

3.2.3.1 Extrema

The extrema refers to the minimum and maximum of a time series in a given period. Equation 3.3 is used to extract the extrema of the given sample as a 12-dimensional feature vector \mathbf{v} .

$$(3.3) \quad \mathbf{v} = \{ \min(x_i) \mid x_i \in X \} \cup \{ \max(x_i) \mid x_i \in X \}$$

In this equation, X is the sample multivariate time series with 6 components (the 3 axes of the accelerometer and the 3 axes of the gyroscope), and each x_i is a univariate component of X . The models that use this feature are named with the suffix EX .

3.2.3.2 Mean and Variance

The mean and variance of each univariate time series (UTS) are calculated and concatenated as shown in Equation 3.6 to form a 12-dimensional feature vector for model training and testing.

$$(3.4) \quad \mu_i = \frac{1}{n} \sum_{j=1}^n x_i^{(j)}$$

$$(3.5) \quad \sigma_i^2 = \frac{1}{n} \sum_{j=1}^n (x_i^{(j)} - \mu_i)^2$$

$$(3.6) \quad \mathbf{v} = \{ \mu_i \mid 1 \leq i \leq m \} \cup \{ \sigma_i^2 \mid 1 \leq i \leq m \}$$

Here, $x_i^{(j)}$ is the j^{th} element in x_i , m is the dimensionality of the multivariate time series, and n , μ_i , and σ_i^2 are the length, mean, and variance of x_i , respectively. When this feature is used in training a model, the model name is suffixed with *MV*.

3.2.3.3 Kurtosis and Crest Factor

Kurtosis is a measure to describe how tailed a function is. A higher kurtosis value means the signal has sharper peaks (Pearson, 1905). Crest factor is defined as the ratio of the peaks in a signal to its effective value (Andersen, 2001). These values are calculated using Equation 3.7 and Equation 3.8, and concatenated to form a 12-dimensional feature vector. For a better understanding of these features, Figure 3.6 illustrates some functions, and their kurtosis and crest factor values.

$$(3.7) \quad Kurt(x_i) = \frac{\frac{1}{n} \sum_{j=1}^n (x_i^{(j)} - \mu_i)^4}{\sigma_i^4}$$

$$(3.8) \quad Crest(x_i) = \frac{\max(|x_i|)}{RMS(x_i)}$$

$$(3.9) \quad \mathbf{v} = \{ Kurt(x_i) \mid x_i \in X \} \cup \{ Crest(x_i) \mid x_i \in X \}$$

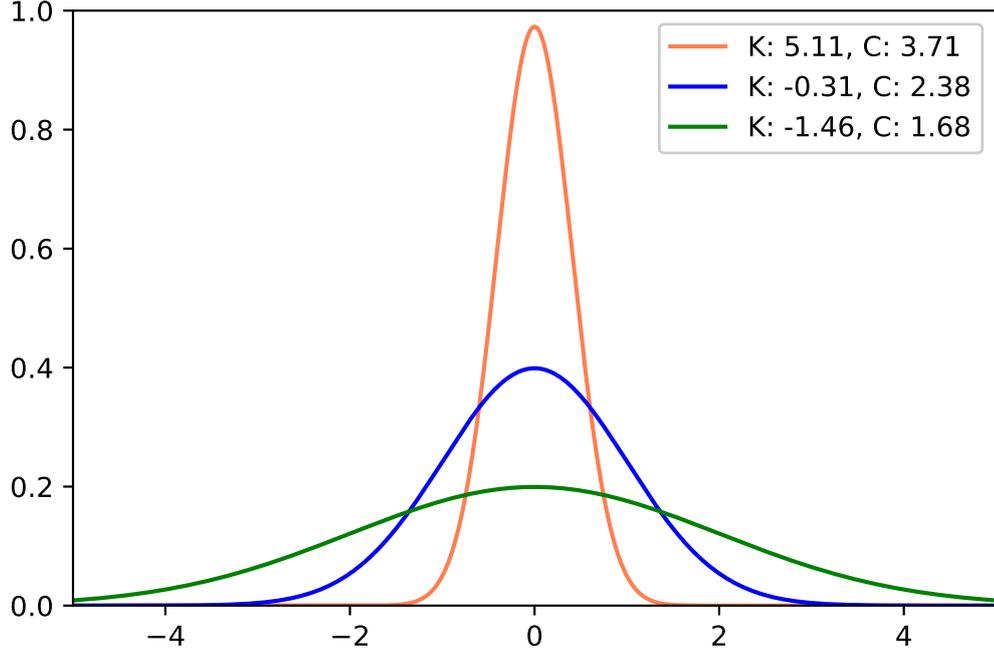


Figure 3.6 Kurtosis (K) and crest factor (C) values of three sample functions

Here, $RMS(x_i)$ is the root mean square value of the univariate time series that can be calculated using Equation 3.10.

$$(3.10) \quad RMS(x_i) = \sqrt{\frac{1}{n} \sum_{j=1}^n (x_i^{(j)})^2}$$

The models that use this feature vector have their name ending in KC .

3.2.3.4 Histogram

We obtain a histogram from the time series by dividing the range of possible values of the signal into a predefined number of bins, and the repetition frequency of the values inside that bin is taken to be the height of that bin. We calculate histograms for each component of the multivariate time series. We calculate the ranges and the bins to obtain the histogram on the train data, and use the same values to obtain the histogram of the test data. The features represent the content of each bin, and since there are 6 components for the multivariate time series, the number of features is $6 \times$ the number of bins. To see the effect of the number of bins in the performance of the models, two histogram representations with 10 and 20 bins were used to train and test the models which produce 60- and 120-dimensional feature

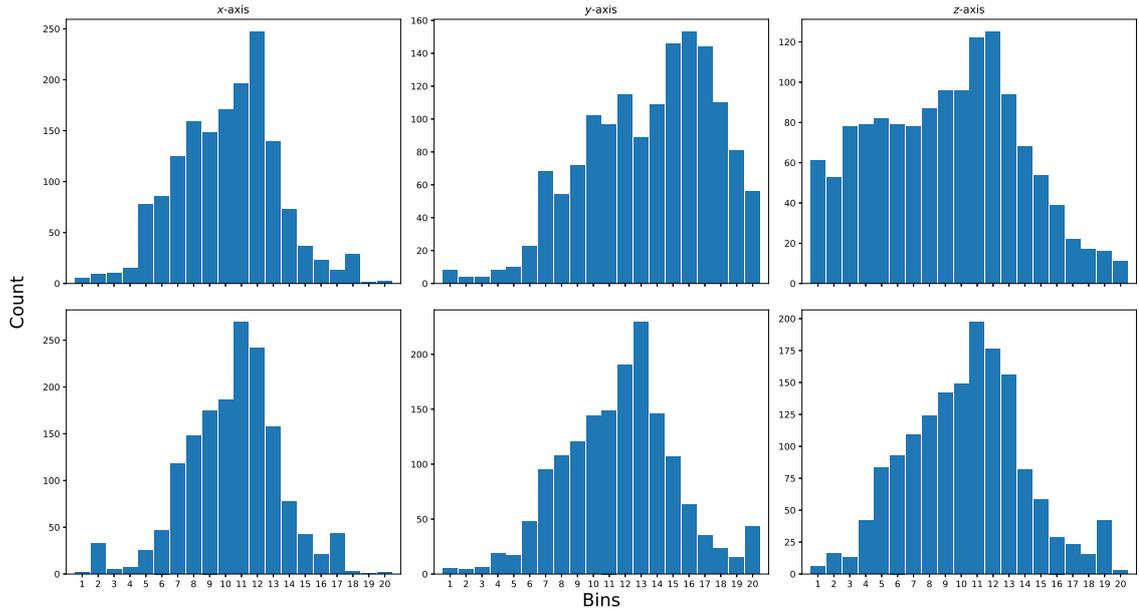


Figure 3.7 Histogram comparison of x -, y -, and z -axis of accelerometer data from normal wash with balanced machine (top) and unbalanced machine (bottom)

vectors, respectively.

If this feature representation is used in training a model, its name ends in $H10$ or $H20$, where 10 and 20 illustrate the number of bins used. Figure 3.7 shows the 20-bin histogram of two 10-second samples from a balanced machine and an unbalanced machine during a normal wash cycle.

3.2.3.5 Uniform Manifold Approximation and Projection

Uniform manifold approximation and projection (UMAP) is a dimension reduction method that is based on Riemannian geometry and algebraic topology (McInnes, Healy, Saul & Grossberger, 2018). This method can be described in terms of weighted graphs, and particularly k -neighbor based graph learning. As such, the reduction is achieved by constructing a weighted graph from the given data. The low-dimensional layout of this graph is then calculated and used to represent the reduced form of the input data. We use this algorithm to reduce each sample to a 3-dimensional feature vector. These vectors are then fed into different machine learning models to compare their performances. The models using this data representation method are named with the suffix $UMAP$.

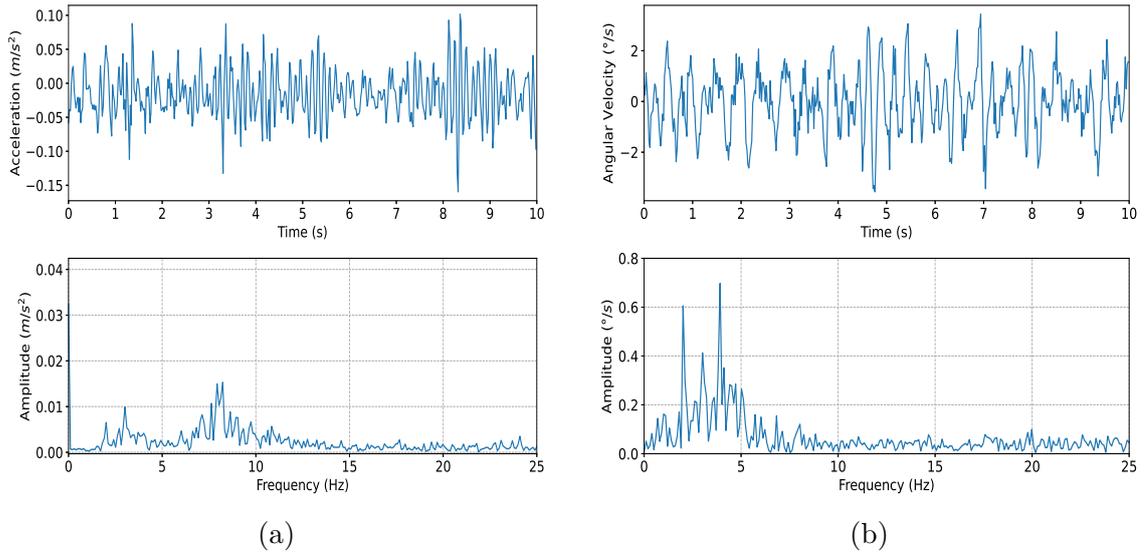


Figure 3.8 Raw data and DFT of x -axis of (a) accelerometer and (b) gyroscope data from a balanced machine during normal wash

3.2.3.6 Frequency Spectrum

The frequency spectrum of a discrete signal is obtained using discrete Fourier transform (DFT), which represents the signal in the frequency domain. Figure 3.8 demonstrates 10 seconds of raw data from the accelerometer and gyroscope during a normal washing cycle and their corresponding DFT representations. Since most of the anomalies in the machine show themselves as a form of vibration, this information can play a significant role in detecting the unbalanced working of the washing machine. This can be observed in Figure 3.9. In this figure, the amplitude of the signal at frequencies around the spin speed of the drum (1.6 Hz) is increased in the presence of an unbalanced load. Fourier transform of a discrete signal can be calculated using Equation 3.11.

$$(3.11) \quad F(j\omega) = \sum_{k=0}^{N-1} f[k]e^{-j\omega kT}$$

In this equation, N is the number of samples and T is the time delay between two consecutive samples. The discrete Fourier transform for each component of the multivariate time series was calculated and concatenated to form the feature vector. Since the length of the Fourier transform is proportional to the signal length, the dimension of the feature vector changes based on the length of the processed sample. This feature representation is shown with the suffix *DFT* in the model names.

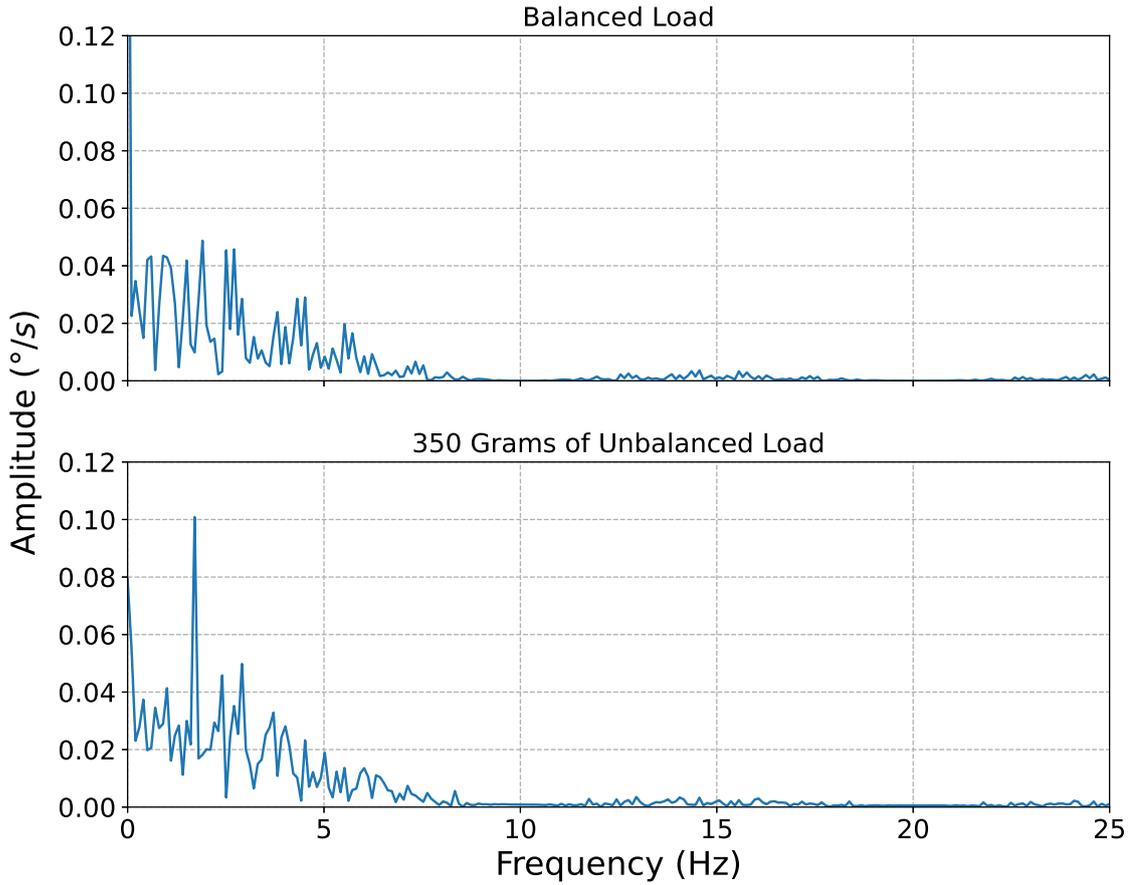


Figure 3.9 Comparing DFT of x -axis of gyroscope data from drum rotating at 100 RPM with balanced load and 350 grams of unbalanced load

3.2.3.7 Wasserstein Time Series Kernel

Wasserstein time series kernel introduced by Bock et al. (2019) measures the similarity between two distribution functions. It implements a sliding window to extract subsequences of the two given series. It then calculates a pairwise distance matrix between the subsequences extracted in the previous step. An optimal transport plan is then calculated to make the similarities more visible. This transport plan is used to match the subsequences of the input series. The kernel is given using Equation 3.12, in which W_1 is the 1st Wasserstein distance between the two time series T_i and T_j .

$$(3.12) \quad WTK(T_i, T_j) := \exp(-\lambda W_1(T_i, T_j)), \quad \lambda \in \mathbb{R}_{>0}$$

The 1st Wasserstein distance is defined as

$$(3.13) \quad W_1(T_i, T_j) := \min_{P \in \Gamma(T_i, T_j)} \langle D, P \rangle_F$$

where P is the transport matrix, $\Gamma(T_i, T_j)$ is the set of all transportation plans, D is the pairwise distance of all subsequences, and $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product.

Since histograms are representatives of signal distribution among bins, we used this method with the 10-bin histogram that was constructed earlier. The resulting kernel was then given to a support vector machine classifier model as a pre-computed kernel. The model names are suffixed with *WTK* to show that they use this feature extraction method.

4. MODEL SELECTION AND TRAINING

In this chapter, the models used to classify the data are described in detail. Section 4.1 describes the different machine learning and neural network algorithms and models used for the imbalance detection task. The hyperparameters used to tune these models, as well as the tuning method used to find the best parameters are reported in Section 4.2. Section 4.3 discusses the different performance metrics used to assess different models and compare them together. Finally, in Section 4.4 we discuss the earliness criteria that we implemented and used in this thesis.

4.1 Models

We classify the imbalances using traditional machine learning algorithms, as well as deep learning models. When training the traditional machine learning models, we use the various feature representations described in Section 3.2.3, while the deep learning models are directly given the data acquired after resampling the raw sensor output. We describe these models in the following sections.

4.1.1 Support Vector Machines

Support vector machines (SVM) are supervised learning models that find the linear hyperplane that maximizes the margin between two classes (Cortes & Vapnik, 1995). Although the model is linear in the original feature space, using the kernel trick, SVMs can efficiently perform non-linear classification. This algorithm was originally introduced as a solution to binary classification, i.e. when there are only two classes for prediction. Although this solution meets our need for machine im-

balance detection, we cannot use it for load imbalance classification without proper modification. To this end, the one-vs-one method was used to expand SVM to a multiclass classifier. In this strategy, each sample is classified for each two possible label combinations, and the sample label is decided via a voting mechanism (Bishop, 2006). Assuming we have k distinct class labels, this method requires $k(k-1)/2$ binary classifiers to classify the k labels.

As for the kernel, radial basis function (RBF) was used in all the implemented SVM models. This kernel can be calculated using Equation 4.1. In this equation, \mathbf{x} and \mathbf{x}' are two feature vectors, and σ is known as the kernel parameter.

$$(4.1) \quad K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

4.1.2 Gradient Boosted Decision Trees

Extreme gradient boosting (XGBoost) is a scalable and resource-efficient tree-based gradient boosting algorithm (Chen & Guestrin, 2016). This algorithm uses an ensemble of trees and gradient boosting to produce a model for classification or regression. It gradually adds new trees to the ensemble to compensate for the residuals of the current models, and hence improve the overall performance of the ensemble. In this work, we used this algorithm to perform classification for both machine imbalance and load imbalance detection problems.

4.1.3 Neural Networks

Artificial neural networks, generally called neural networks, are learning models based on neuron units. These units, which are inspired by brain neurons, compute the weighted sum of their inputs, and use an activation function to produce the output. Equation 4.2 illustrate the mathematical model of an artificial neuron.

$$(4.2) \quad y = \Phi\left(\sum_{i=0}^n \omega_i x_i + b\right)$$

In this equation, x_i is the i^{th} input, ω_i in the weight assigned to it, and b is the bias term for the neuron. The function $\Phi(\cdot)$ is known as activation function, and

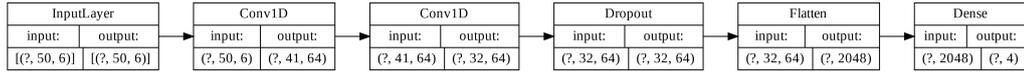


Figure 4.1 CNN model for load imbalance detection and classification with 1 second of input data

activates the output of the neuron based on a threshold function.

Neural networks use these units in different arrangements and layers to perform the learning and prediction task. What makes these networks powerful is that they can operate with the raw data and extract the required features on their own, without the need for manual feature extraction steps. In this work, we used the following models for imbalance detection.

4.1.3.1 Convolutional Neural Networks

Convolutional neural network (CNN) is one of the most widely used neural networks and has made significant achievements in different areas such as natural language processing and computer vision (Li, Yang, Peng & Liu, 2020). Figure 4.1 shows one of the CNN models used for detecting load imbalance issue and classify the weight of the load. This sample model operates on the 6-dimensional inputs of size 1 second and consists of two 1-dimensional convolution layers, each with 64 filters and a kernel size of 10, followed by a dropout layer with rate 0.5, and a fully connected layer with four outputs, each representing one of the classes to be predicted. We use categorical cross-entropy for model training, and the training stops early if the validation loss does not improve by more than 0.001 during five epochs.

4.1.3.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to more accurately model the long-range dependencies among temporal sequences (Hochreiter & Schmidhuber, 1997). In this work, we built and trained three LSTM models with 25, 50, and 100 nodes. The training loss function is selected to be categorical cross-entropy, and the same early stopping criteria as explained in Section 4.1.3.1 is implemented to avoid overtraining the networks.

4.2 Hyperparameter Tuning

The hyperparameters of the different used models are tuned using the grid search method to produce the best outcome. We apply 3-fold cross-validation on training data to assess the performance of the models trained with different combinations of the hyperparameters. The values yielding the best mean performance are then used to train the final models and we report the performance on the test data. The hyperparameters used in the best-performing models can be found in Appendix C.

The SVM models were tuned for regularization parameter (C) and kernel coefficient (γ). The values chosen for these parameters are as follows.

$$C = \{10^i \mid i \in \mathbb{Z}, -2 \leq i \leq 4\}$$

$$\gamma = \{10^i \mid i \in \mathbb{Z}, -6 \leq i \leq 0\}$$

As an example, Figure 4.2 shows the grid search results for the SVM model trained with 10-bin histogram representation of samples of length 2 seconds. In this example, the optimal parameter values are $\gamma = 1.0$ and $C = 10$.

XGBoost has more parameters to be tuned. In this work, these models were tuned for maximum tree depth of base learners (depth_{max}), minimum sum of instance weight needed for a child (w_{min}), and minimum loss reduction (γ). These parameters are tuned using the following values.

$$\text{depth}_{max} = \{i \in \mathbb{Z} \mid 3 \leq i \leq 9\}$$

$$w_{min} = \{i \in \mathbb{Z} \mid 1 \leq i \leq 5\}$$

$$\gamma = \left\{ \frac{i}{10} \mid i \in \mathbb{Z}, 0 \leq i \leq 4 \right\}$$

4.3 Performance Metrics

We evaluate the implemented models using different classification performance metrics. These metrics were used during model selection, as well as assessing the final

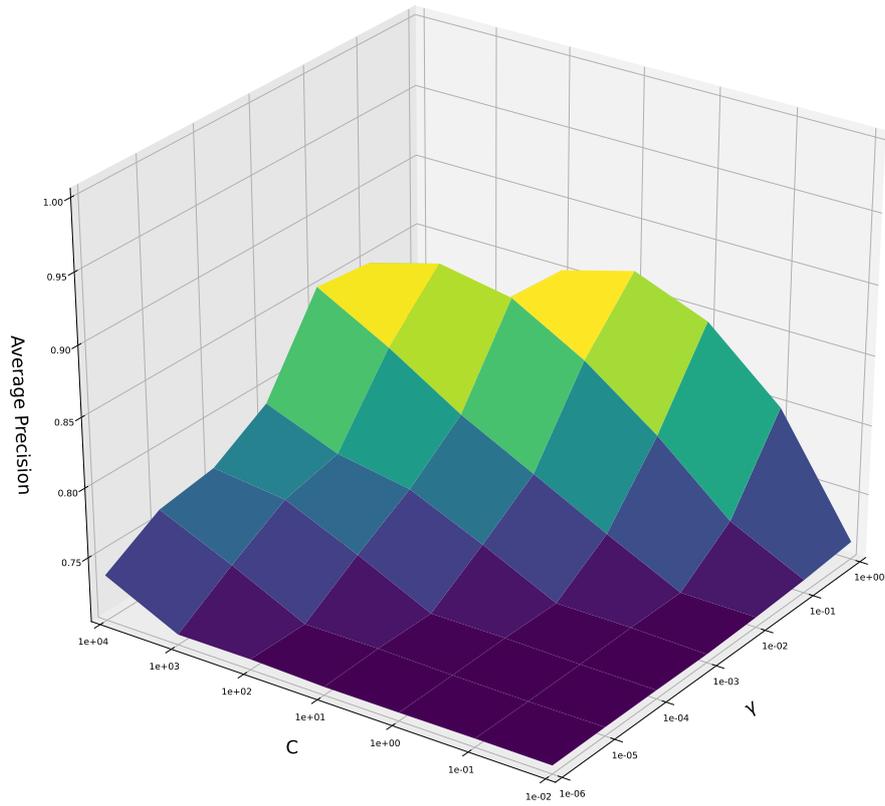


Figure 4.2 Grid search results with SVM model for γ and C parameters with 10-bin histogram as feature vector and samples of length 2 seconds

classifiers. These metrics are as follows:

4.3.1 Precision

Precision is the ratio of truly predicted positive classes to all the samples predicted as being in positive. It can be calculated using Equation 4.3. In this equation, tp and fp stand for truly classified positive samples and mistakenly classified negative samples, respectively.

$$(4.3) \quad \text{Precision} = \frac{tp}{tp + fp}$$

4.3.2 Recall

Recall is the ratio of truly predicted positive classes to the number of all positive samples in the data set. It is calculate as shown in Equation 4.4, with fn being the number of positive samples that are mistakenly classified as negative.

$$(4.4) \quad \text{Recall} = \frac{tp}{tp + fn}$$

4.3.3 F₁ Score

This metric is the harmonic mean of precision and recall metrics, and is calculated as shown in Equation 4.5.

$$(4.5) \quad F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This metric is used during the grid search, as well as performance assessment of the models used for load imbalance detection and classification. Since this problem is a multiclass classification problem, macro averaged F₁ score was used. Macro averaging includes calculating the average of the scores for each individual class as shown in Equation 4.6, in which n is the number of classes.

$$(4.6) \quad F_{1macro} = \frac{\sum_{i=1}^n F_{1i}}{n}$$

4.3.4 Average Precision

Average precision is the precision averaged over all recall values. This metric can be calculated using the Equation 4.7.

$$(4.7) \quad \text{AP} = \sum_{k=1}^N P(k) \Delta r(k)$$

In this equation, N is the total number of samples, $P(k)$ is the precision achieved with the first k samples, and $\Delta r(k)$ is the change in recall from $k - 1$ to k samples. Average precision was used to tune hyperparameters and measure the performance of the models used for machine imbalance detection task.

4.4 Earliness

In this work, we first establish a definition for earliness to be able to achieve early detection of balance issues in the washing machine. We define minimum imbalance detection length (MIDL) as the shortest length of the multivariate time series sensor data for which the model performance is at maximum, and no data series longer than that can produce a better result. Equation 4.8 illustrates this definition.

$$(4.8) \quad \text{MIDL} = L \ni \text{Perf}(L) \geq \text{Perf}(L') \forall L < L' \leq N$$

Here, $\text{Perf}(L)$ is the performance of the model with a data of length L , and N is the full length of the sample data. To select the best model, we find the model with the best performance among all, while having the smallest MIDL. To do so, the models discussed in Section 4.1 were trained and tested with samples of the following lengths: 0.5, 1, 2, 5, 10, 20, and 30 seconds. As a result of combining the different models, feature extraction methods, and sample lengths, a total of 259 models were trained and tested for the two imbalance detection problems at hand.

5. RESULTS

As discussed earlier in Section 4.3, F_1 performance score was used for the binary classification problem of detecting machine imbalance issues, and average precision was used when detecting the unbalanced load and classifying its weight. The models are trained and tested with samples of different lengths as explained in Section 4.4, and the resulting performance scores are recorded and illustrated in Appendix A.

Table A.1 reports the test average precision of the models trained for machine imbalance detection. The data used for this task includes the drum spinning clockwise and counterclockwise, with pauses between the two rotations. Due to this non-homogeneous nature of the data, neural network models performed very poorly and were not included in the results. Table A.2 demonstrates the F_1 test scores of the models used to classify load imbalance with respect to the length of the input sequence. The best performance for each sequence length is highlighted, and the corresponding confusion matrix is illustrated in Appendix B.

It can be seen from these tables that the frequency spectrum feature extraction method results in the best performance among the models with the shortest data length in both imbalance detection tasks. However, as longer sequences are used, other methods outperform DFT and produce better results. In the machine imbalance detection task, the SVM model provides the best overall prediction performance when used with mean and variance representation of the signal. Load imbalance is better detected and classified with a 10-bin histogram representation of the data and an SVM classifier.

The selected best-performing models for the two tasks are then tested with two different strategies as follows:

- In the first method, the test cases were divided into *easy* and *difficult* sets. The *easy* samples are the ones that the simplest model with the most basic feature extraction method can correctly classify, and the *difficult* ones are the remaining samples. Among the models used in this work, SVM with extrema as the feature vector is chosen as the baseline model. The test samples that

Table 5.1 Average precision test score of best-performing models with respect to sample lengths for machine imbalance detection with difficult cases

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
SVM_MV	0.865	0.962	0.988	0.995	0.998	0.998	0.999
XGB_DFT	0.971	0.980	0.996	0.998	0.998	0.997	0.991
XGB_MV	0.885	0.972	0.995	0.998	0.999	0.998	0.993

Table 5.2 F₁ test score of best-performing models with respect to sample lengths for load imbalance detection with difficult cases

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
SVM_H10	0.579	0.796	0.923	1.000	1.000	1.000	1.000
SVM_H20	0.608	0.836	0.937	0.974	0.544	0.357	1.000
SVM_UMAP	0.746	0.922	0.971	1.000	0.297	1.000	1.000
XGB_DFT	0.638	0.833	0.960	0.987	1.000	1.000	1.000

this model could successfully classify were labeled as *simple*, and the rest as *difficult*. The models with the best performance were used to classify the *difficult* samples separately, and the corresponding scores are illustrated in Table 5.1 and Table 5.2. As manifested by these tables, the performance of the models shows a general decline with the *hard* samples; however, as the length of the samples increases, they start to produce better results and even achieve the same performance as the ones tested on all the data.

- The second strategy involves adding noise to the training data and re-training the best performing models with noisy data. The new models are then tested with the non-noisy test data. In order to produce the noise, a multivariate normal distribution is used. A k -dimensional normal distribution is shown using Equation 5.1.

$$(5.1) \quad \mathbf{X} \sim \mathcal{N}_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

In this equation, $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix. The mean and covariance of the original signals were used to produce the noise. The noise produced using this distribution is added to the training data, and the best-performing models are re-trained using this data. The test scores are

Table 5.3 Average precision test score of best-performing models with respect to sample lengths for machine imbalance detection with noisy data

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
SVM_MV	0.752	0.893	0.967	0.987	0.996	0.998	0.999
XGB_DFT	0.833	0.956	0.968	0.993	0.979	0.986	0.976
XGB_MV	0.724	0.896	0.964	0.985	0.990	0.987	0.984

Table 5.4 F_1 test score of best-performing models with respect to sample lengths for load imbalance detection with noisy data

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
SVM_H10	0.419	0.369	0.502	0.621	0.657	0.755	0.973
SVM_H20	0.338	0.463	0.291	0.470	0.830	0.881	0.786
SVM_UMAP	0.266	0.098	0.094	0.114	0.109	0.233	0.103
XGB_DFT	0.746	0.831	0.865	0.961	1.000	1.000	0.997

illustrated in Table 5.3 and Table 5.4. As seen, the general performance of these models is lower than the ones with the non-noisy data. In addition, the models used to predict load imbalance show a sharp decrease in performance with the introduction of the noise. The reason for this is that the signal amplitudes in the data collected during this imbalance condition are much lower than the ones in machine imbalance data. As a result, noise greatly affects and overwhelms the original signal. Yet again, the performances start to recover as longer sequences of data are used in training the models.

Different strategies are implemented in order to choose a single best-performing model for each of these imbalance detection problems. However, we need to keep in mind that these models are supposed to work on microcontrollers with limited available resources. As a result, we might need to add some constraints, such as limiting input signal length, choosing simpler models, and preferring less resource-consuming feature extraction strategies, and possibly sacrifice performance to be able to deploy the models on the available hardware. The strategies for selecting the best-performing models are discussed and summarized as follows.

- Minimum Performance Threshold

In this strategy, a minimum required performance score is defined, and the

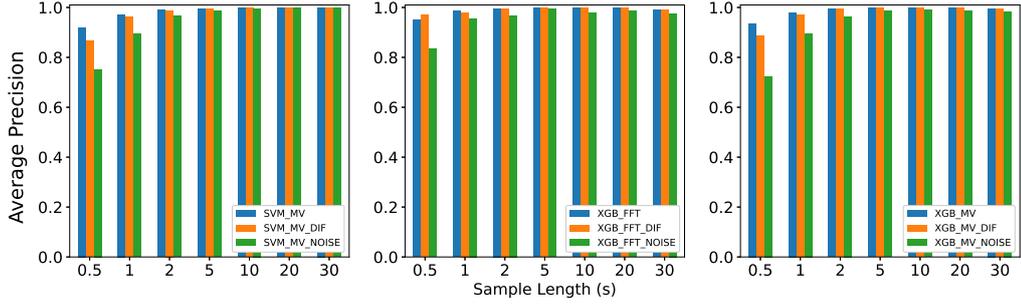


Figure 5.1 Comparison of performance of the models with difficult cases and noisy training data for machine imbalance detection task

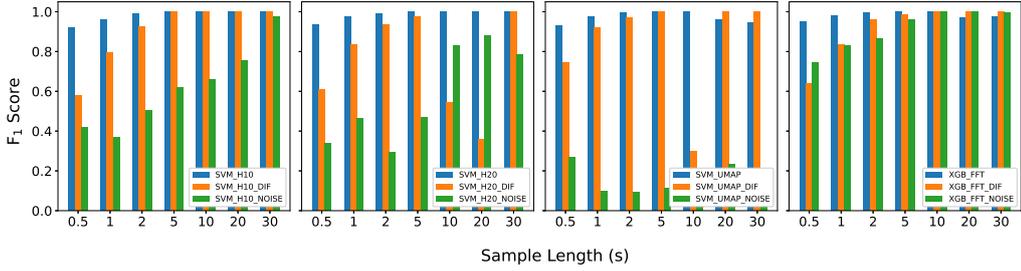


Figure 5.2 Comparison of performance of the models with difficult cases and noisy training data for load imbalance detection task

model that can achieve that performance with the shortest input sequence is selected to be the best-performing model. For the purpose of this work, we define this performance threshold to be 0.900. As a result, there are six models for machine imbalance detection, and 16 models for load imbalance classification that satisfy this criterion. Among the machine imbalance detection models, XGBoost with DFT representation of the input data has an average precision score of 0.951 with 500 ms of input data, which is the highest among all other six and is chosen as the best model. Likewise, XGBoost with DFT feature extraction performs the best with the F_1 score of 0.948 for the load imbalance classification problem.

- **Best Performance with Different Data Lengths**
In order to choose the best models with this strategy, we consider the best performing models highlighted in Table A.1 and Table A.2, and choose the ones that perform the best with multiple data lengths. It can be observed that using this criterion, the best-performing model for machine imbalance detection is XGBoost with mean and variance representation of the data, and SVM with 10-bin histogram feature extraction for load imbalance classification problem.
- **Lowest Minimum Imbalance Detection Length**
The third strategy makes use of the MIDL definition introduced in Section

4.4. The desired model is chosen as the one which has a performance above a predefined threshold while maintaining the smallest MIDL. Using this criterion, it can be found that the best performing model for machine imbalance detection is XGBoost with mean and variance features with $MIDL = 5$. LSTM with 50 nodes and $MIDL = 1$ is observed to be the best performing model for load imbalance detection using the discussed criterion.

- Lowest Performance Drop with Noise and Difficult Samples

The final method of choosing the best model uses the results of the tests with *difficult* samples and noisy train data. As illustrated by Figure 5.1 and Figure 5.2, XGBoost with DFT data representation shows the least amount of drop in performance for both machine imbalance and load imbalance detection tasks. Consequently, this model is chosen as the best performing model with the mentioned criterion.

6. CONCLUSION

Among the durable goods found in most households nowadays, washing machines are the ones most susceptible to anomalous behavior due to the heavy loads inside them and the high spin speeds of their drums. These anomalies generally manifest themselves as forms of vibration and can be detrimental to the performance and lifespan of the machine if not detected and controlled in a timely manner. However, if they are detected at low spin speeds and below the resonant frequency of the machine, proper measures can be taken to alleviate their consequences, or warnings can be displayed to the user to take necessary actions. Among these anomalies, imbalance issues such as improper leveling of the machine and uneven distribution of the load inside the drum at high spin speeds can cause vigorous vibrations and permanently damage the machine if stayed unnoticed.

The machine imbalance problem, which is the improper adjustment of the leveling legs, can be avoided by the end-user. However, the failure to do so needs to be compensated by the machine to avoid the phenomena known as "Walking Washer" and the harm resulted from it. On the other hand, since the arrangement of the laundry inside the drum at high spin speeds cannot be foreseen, it is up to the washing machine to detect the uneven distribution of the load inside the drum, and take necessary precautions to avoid extreme vibrations and possible damage to the machine.

In this thesis, we propose a method for detecting both types of imbalance issues by using an inertial measurement unit attached to the drum to monitor its vibrations. We investigate whether these imbalance issues can be detected early in the normal washing cycles, without the need for any special condition provided by the machine. To this end, we built machine learning models to predict machine imbalance and load imbalance using data collected from machines.

We trained 259 models with different machine learning algorithms, using different lengths of the input data, and inputting different feature extraction methods. Among these, only the ones that provide the best performance for each detection

task were selected. In the machine imbalance detection problem, XGBoost with discrete Fourier transform as feature extraction strategy offers the best performance with the shortest length of input data, while the highest performance is achieved using SVM with mean and variance of the input time series. Similarly, in the load imbalance classification task XGBoost with discrete Fourier transform features has the best performance score with the shortest input. However, the highest performance is achieved using a 10-bin histogram of the input data and SVM classifier. In both imbalance detection tasks, the selected models can detect the problem with just 500 ms of sensor data, and a performance score of 95%. We also observe that the performance can be enhanced even more by providing the models with longer sequences of input data. The developed imbalance detection methods do not need any mathematical modeling of the machine, nor any expertise about the internal workings of it.

All the data used for this study were collected from a single washing machine. To further generalize the results of this study, more machines with different models need to be used for data collection. This work can also be further expanded by expanding it to other less evident faults in the machine such as improper belt tension and bearing lubrication. In addition, the data can be collected in a uniform manner, and the two imbalance detection models can be merged into a single model that can detect both types of imbalance issues simultaneously, and classify the amount of unbalanced load in the machine if such a problem is detected. Furthermore, estimating the horizontal position of the unbalanced load in the drum can be investigated as a future work.

In these models, we experiment with simple feature extraction and classification algorithms to make the solution implementable on the limited resources available to the control systems inside the washing machines. However, to deploy the models in the machines other hardware constraints could be taken into account and the models could be further simplified.

BIBLIOGRAPHY

- Achenchabe, Y., Bondu, A., Cornuéjols, A., & Dachraoui, A. (2020). Early classification of time series. cost-based optimization criterion and algorithms. *ArXiv, abs/2005.09945*.
- Alonso González, C. J. & Diez, J. J. R. (2004). Boosting interval-based literals: Variable length and early classification. In *Data mining in time series databases* (pp. 149–171). World Scientific.
- Andersen, B. (2001). Crest factor analysis for complex signal processing. *RF DESIGN*, 24(10), 40–57.
- Anderson, H. S., Parrish, N., & Gupta, M. R. (2012). Early time series classification with reliability guarantee. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).
- Baydogan, M. G., Runger, G., & Tuv, E. (2013). A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence*, 35(11), 2796–2802.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*, (pp. 128). Berlin, Heidelberg: Springer-Verlag.
- Bock, C., Togninalli, M., Ghisu, E., Gumbsch, T., Rieck, B., & Borgwardt, K. (2019). A wasserstein subsequence kernel for time series. In *Proceedings of the 19th IEEE International Conference on Data Mining (ICDM)*.
- Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, (pp. 785–794).
- Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.
- Dachraoui, A., Bondu, A., & Cornuejols, A. (2013). Early classification of individual electricity consumptions. *RealStream2013 (ECML)*, 18–21.
- Ghalwash, M. F. & Obradovic, Z. (2012). Early classification of multivariate temporal observations by extraction of interpretable shapelets. *BMC bioinformatics*, 13(1), 195.
- Ghalwash, M. F., Radosavljevic, V., & Obradovic, Z. (2014). Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 402–411).
- Hatami, N. & Chira, C. (2013). Classifiers with a reject option for early time-series classification. In *2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL)*, (pp. 9–16). IEEE.
- He, G., Duan, Y., Peng, R., Jing, X., Qian, T., & Wang, L. (2015). Early classification on multivariate time series. *Neurocomputing*, 149, 777–787.
- Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Janke, B. P., Richmond, P. J., & Zasowski, P. E. (2015). Method for detecting satellization speed of clothes load in a horizontal axis laundry treating appliance. U.S. Patent No. 9,115,456. Google Patents.
- Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufer, M., Verstockt,

- S., Van de Walle, R., & Van Hoecke, S. (2016). Convolutional neural network based fault detection for rotating machinery. *Journal of Sound and Vibration*, 377, 331–345.
- Kadous, M. W. (1999). Learning comprehensible descriptions of multivariate time series. In *ICML*, volume 454, (pp. 463).
- Lee, D.-S. & Kim, D.-w. (2010). Apparatus for sensing type of unbalance of washing machine and method thereof. U.S. Patent No. 7,788,755. Google Patents.
- Li, Z., Yang, W., Peng, S., & Liu, F. (2020). A survey of convolutional neural networks: Analysis, applications, and prospects.
- McInnes, L., Healy, J., Saul, N., & Grossberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3, 861.
- Murray, P., Henderson, M. I., Marcetic, D., Marcinkiewicz, J. G., Sadasivam, V., & Rajarathnam, A. V. (2011). Method and system for determining a washing machine load unbalance. U.S. Patent No. 7,905,122. Google Patents.
- Nelwamondo, F. V., Marwala, T., & Mahola, U. (2006). Early classifications of bearing faults using hidden markov models, gaussian mixture models, mel-frequency cepstral coefficients and fractals. *International Journal of Innovative Computing, Information and Control*, 2(6), 1281–1299.
- Pearson, K. (1905). "Das Fehlergesetz und Seine Verallgemeinerungen Durch Fechner und Pearson." A Rejoinder. *Biometrika*, 4(1-2), 169–212.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Ramasubramanian, M. K. & Tiruthani, K. (2009). A capacitive displacement sensing technique for early detection of unbalanced loads in a washing machine. *Sensors*, 9(12), 9559–9571.
- Shannon, C. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1), 10–21.
- Xing, Z., Pei, J., & Philip, S. Y. (2009). Early prediction on time series: a nearest neighbor approach. In *Twenty-First International Joint Conference on Artificial Intelligence*. Citeseer.
- Xing, Z., Pei, J., Yu, P. S., & Wang, K. (2011). Extracting interpretable features for early classification on time series. In *Proceedings of the 2011 SIAM international conference on data mining*, (pp. 247–258). SIAM.
- Yörükoğlu, A. & Altuğ, E. (2012). Estimation of unbalanced loads in washing machines using fuzzy neural networks. *IEEE/ASME Transactions on Mechatronics*, 18(3), 1182–1190.
- Yuan, Y. (2008). Sensor fusion based testing station for unbalanced load estimation in horizontal washing machines. In *2008 IEEE Instrumentation and Measurement Technology Conference*, (pp. 1424–1428).
- Zhang, Z., Xie, T., Garstecki, G. M., Xie, M. M., Slabbekoorn, S. D., & Buendia, A. R. (2011). Method and apparatus for monitoring load size and load imbalance in a washing machine. U.S. Patent No. 7,958,585. Google Patents.

APPENDIX A

Performance of the Models

Table A.1 Average precision test score of different models with respect to sample lengths for machine imbalance detection

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
SVM_EX	0.860	0.853	0.845	0.828	0.797	0.782	0.743
SVM_DFT	0.938	0.978	0.991	0.994	0.995	0.994	0.996
SVM_H10	0.890	0.933	0.961	0.982	0.988	0.992	0.996
SVM_H20	0.909	0.951	0.972	0.983	0.992	0.996	0.996
SVM_KC	0.706	0.715	0.706	0.741	0.738	0.731	0.789
SVM_MV	0.920	0.972	0.990	0.995	0.998	0.999	0.999
SVM_UMAP	0.679	0.671	0.686	0.671	0.686	0.635	0.676
SVM_WTK	0.841	0.894	0.928	0.965	0.976	0.985	0.994
XGB_EX	0.871	0.871	0.857	0.835	0.820	0.806	0.804
XGB_DFT	0.951	0.986	0.995	0.998	0.998	0.997	0.989
XGB_H10	0.886	0.928	0.956	0.982	0.991	0.996	0.999
XGB_H20	0.908	0.945	0.969	0.989	0.993	0.997	0.999
XGB_KC	0.727	0.738	0.727	0.726	0.728	0.718	0.759
XGB_MV	0.936	0.979	0.995	0.998	0.998	0.998	0.995
XGB_UMAP	0.674	0.670	0.669	0.669	0.651	0.695	0.676

Table A.2 F_1 test score of different models with respect to sample lengths for load imbalance detection

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
CNN	0.920	0.938	0.977	0.976	0.997	0.998	1.000
LSTM25	0.917	0.947	0.977	0.468	0.933	0.096	0.000
LSTM50	0.933	0.968	0.967	0.764	0.492	0.706	0.000
LSTM100	0.931	0.967	0.976	0.626	0.000	0.221	0.124
CNN_LSTM25	0.908	0.880	0.706	0.936	0.936	0.653	0.636
CNN_LSTM50	0.881	0.939	0.949	0.990	0.000	0.003	0.800
CNN_LSTM100	0.897	0.900	0.961	0.667	0.701	0.000	0.498
SVM_EX	0.922	0.944	0.960	0.975	0.975	0.982	0.976
SVM_DFT	0.921	0.966	0.994	0.999	0.999	0.920	0.907
SVM_H10	0.919	0.962	0.988	1.000	1.000	0.998	1.000
SVM_H20	0.933	0.973	0.990	0.999	0.999	1.000	1.000
SVM_KC	0.684	0.731	0.799	0.868	0.898	0.934	0.943
SVM_MV	0.943	0.971	0.993	0.998	1.000	1.000	1.000
SVM_UMAP	0.929	0.976	0.996	0.999	1.000	0.959	0.946
SVM_WTK	0.849	0.906	0.946	0.993	1.000	1.000	0.997
XGB_EX	0.925	0.945	0.962	0.976	0.984	0.988	0.989
XGB_DFT	0.948	0.979	0.994	0.999	1.000	0.972	0.976
XGB_H10	0.921	0.964	0.987	0.999	0.999	1.000	0.997
XGB_H20	0.937	0.973	0.991	1.000	0.998	0.998	0.997
XGB_KC	0.680	0.722	0.792	0.860	0.905	0.938	0.965
XGB_MV	0.942	0.975	0.994	0.998	0.998	1.000	0.997
XGB_UMAP	0.927	0.974	0.994	1.000	1.000	0.956	0.928

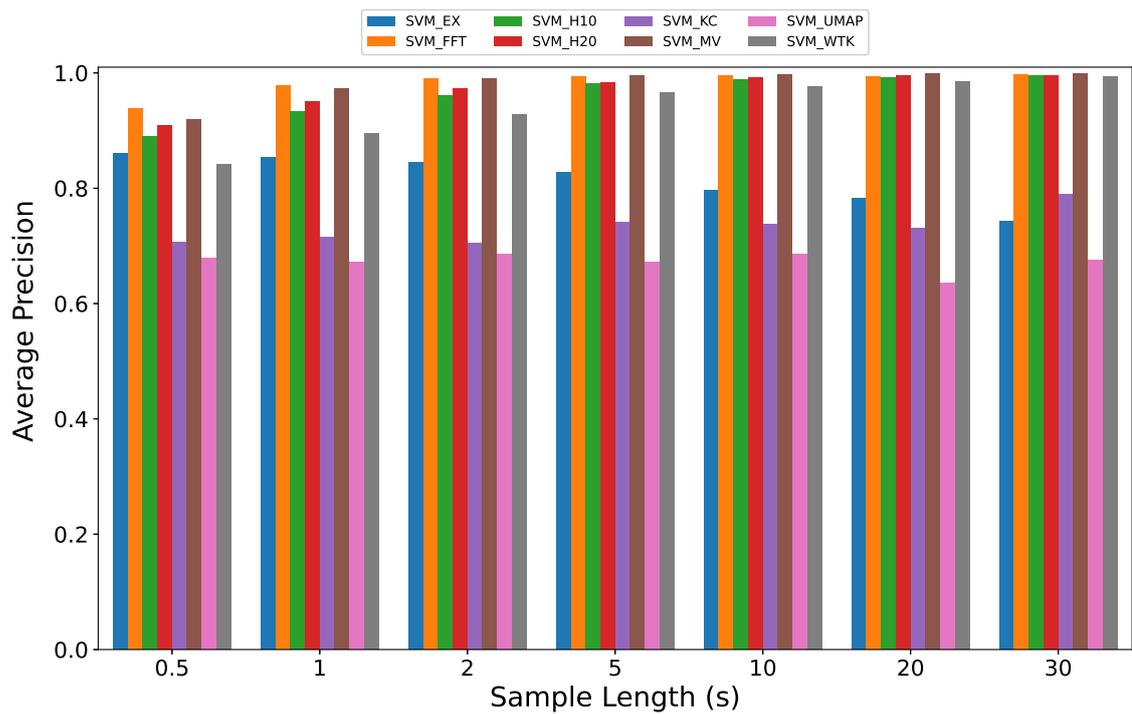


Figure A.1 Average precision of SVM models on test data to detect machine imbalance with different input lengths

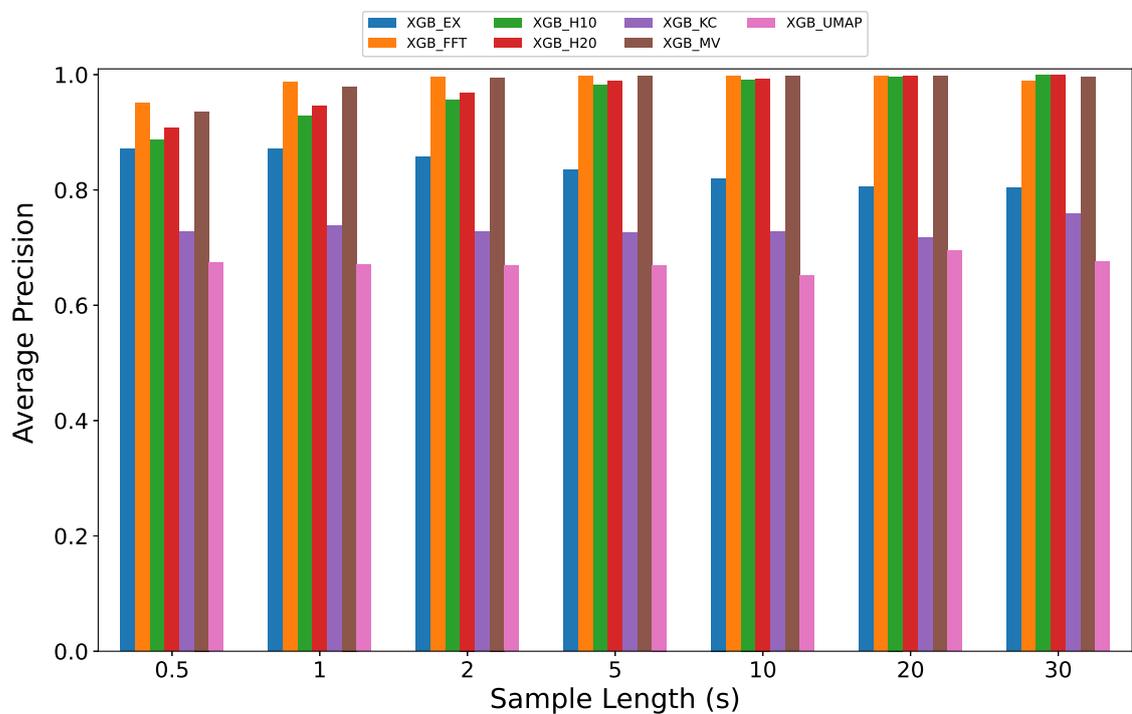


Figure A.2 Average precision of XGBoost models on test data to detect machine imbalance with different input lengths

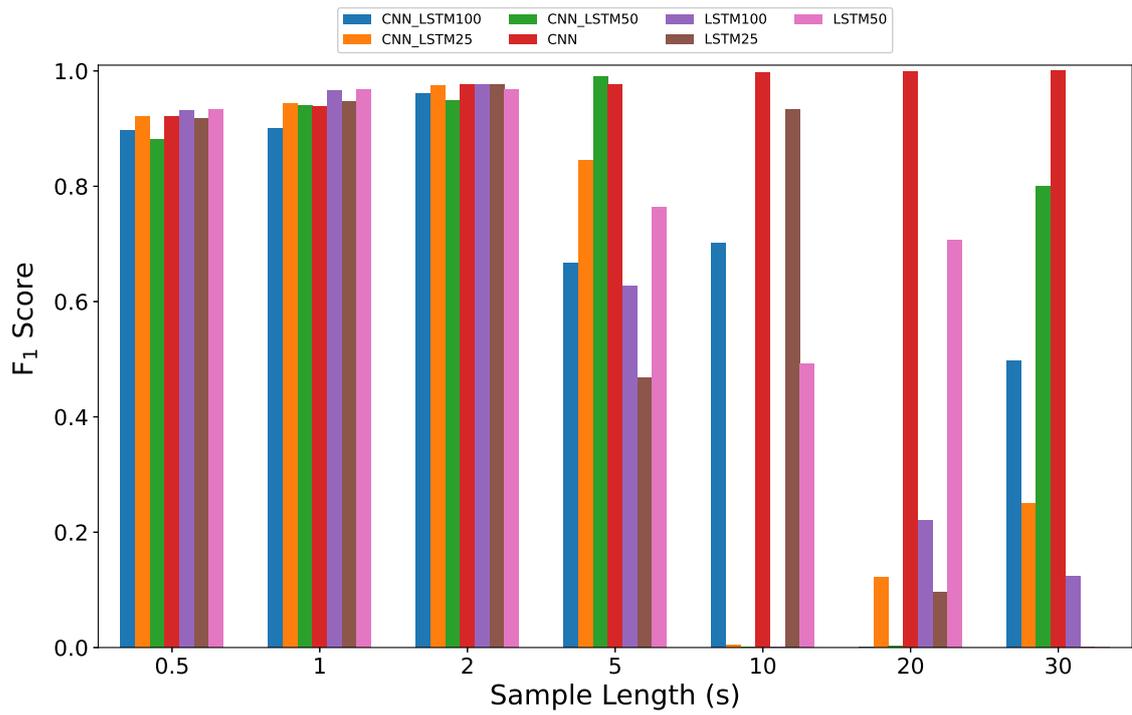


Figure A.3 F₁ score of neural network models on test data to detect load imbalance with different input lengths

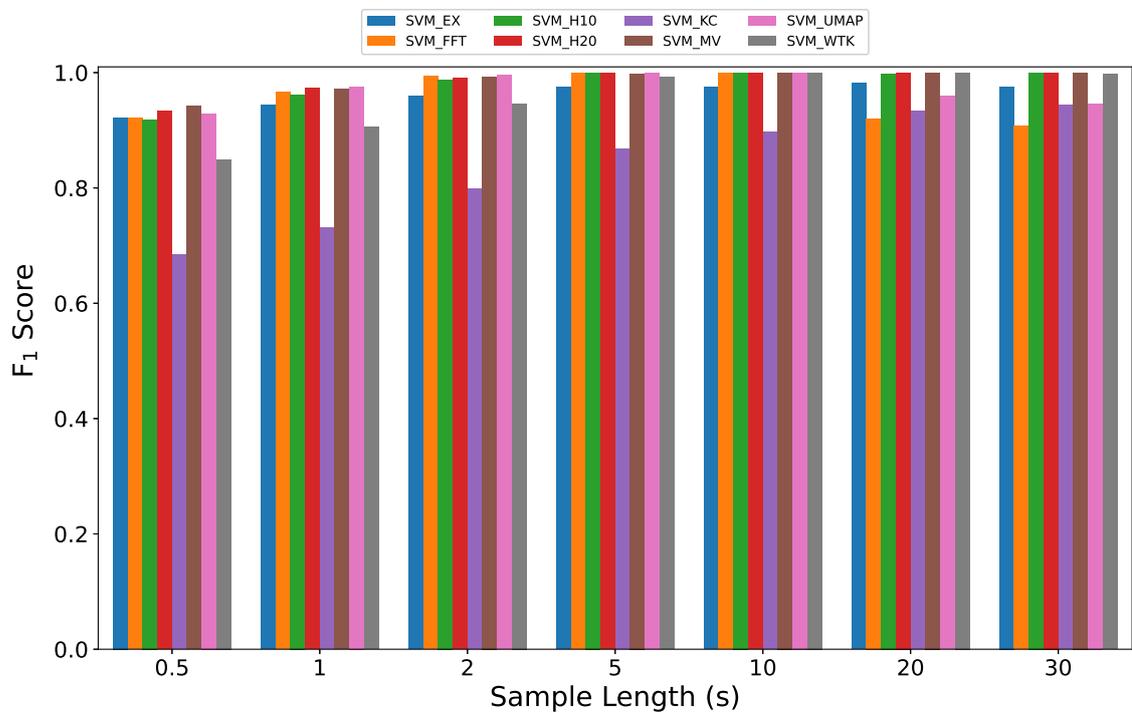


Figure A.4 F₁ score of SVM models on test data to detect load imbalance with different input lengths

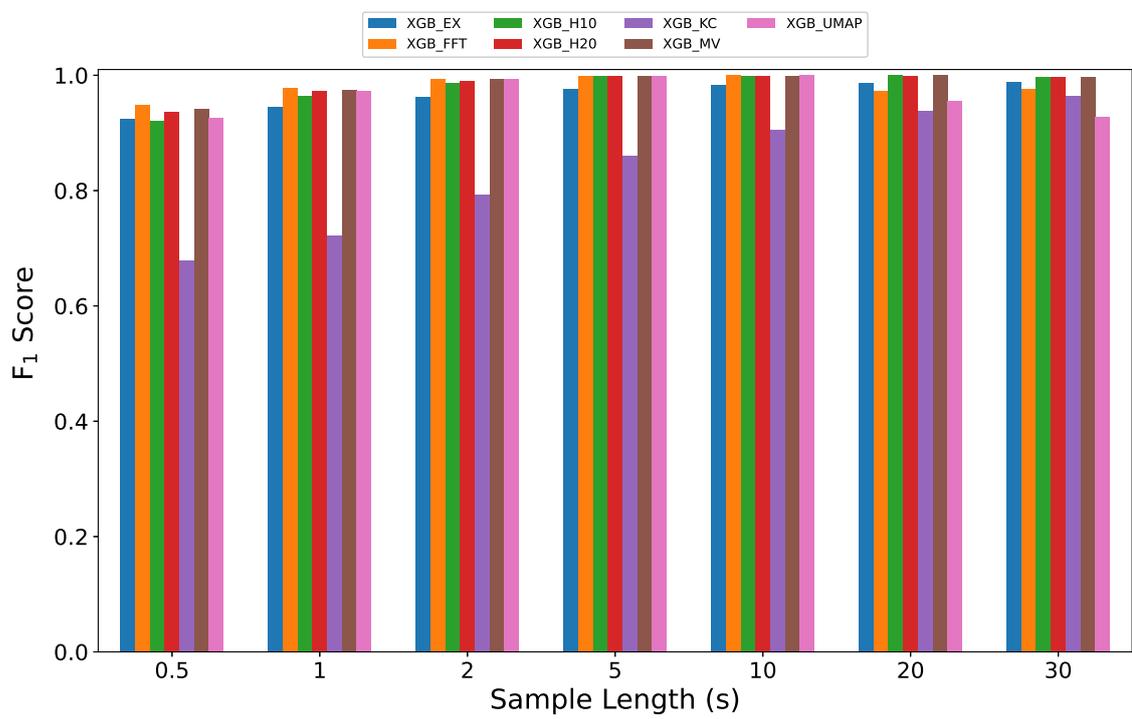


Figure A.5 F₁ score of XGBoost models on test data to detect load imbalance with different input lengths

APPENDIX B

Confusion Matrices of Best-performing Models

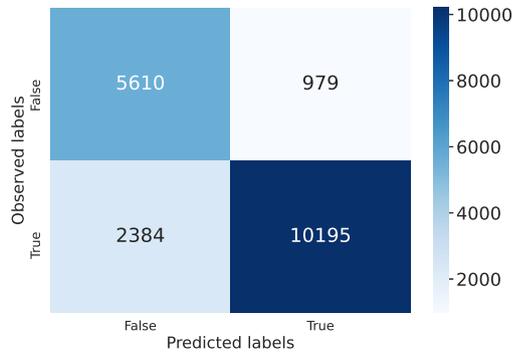


Figure B.1 Confusion matrix of XGBoost model with DFT features to detect machine imbalance with input length of 0.5 second

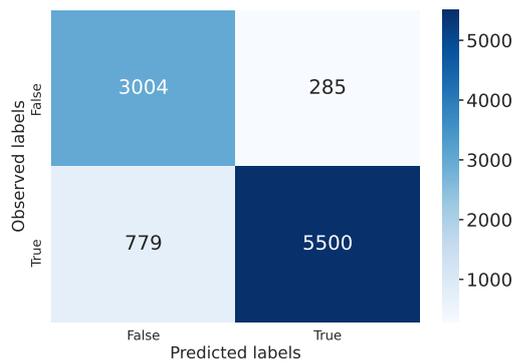


Figure B.2 Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 1 second

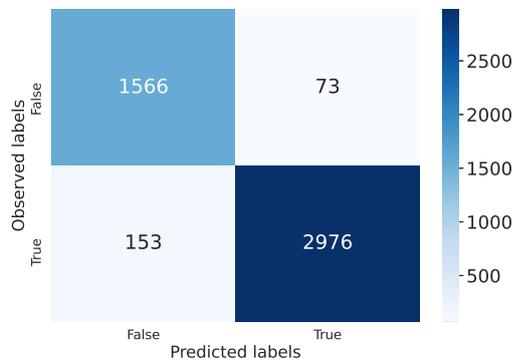


Figure B.3 Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 2 seconds



Figure B.4 Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 5 seconds



Figure B.5 Confusion matrix of XGBoost model with mean and variance features to detect machine imbalance with input length of 10 seconds



Figure B.6 Confusion matrix of SVM model with mean and variance features to detect machine imbalance with input length of 20 seconds

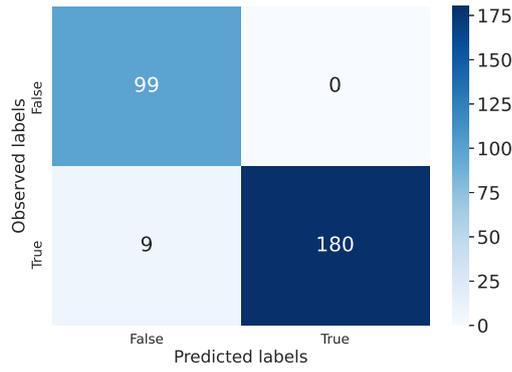


Figure B.7 Confusion matrix of SVM model with mean and variance features to detect machine imbalance with input length of 30 seconds

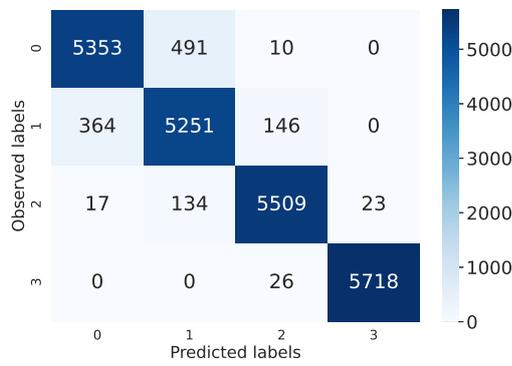


Figure B.8 Confusion matrix of XGBoost model with DFT features to detect load imbalance with input length of 0.5 second



Figure B.9 Confusion matrix of XGBoost model with DFT features to detect load imbalance with input length of 1 second

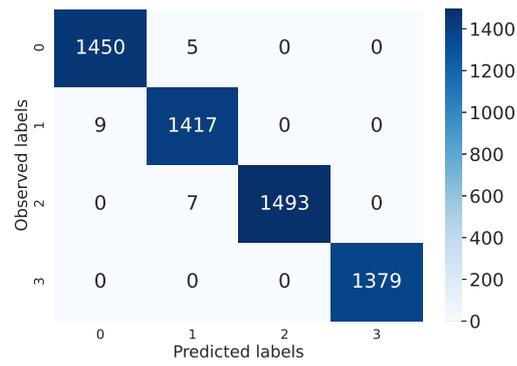


Figure B.10 Confusion matrix of SVM model with UMAP features to detect load imbalance with input length of 2 seconds

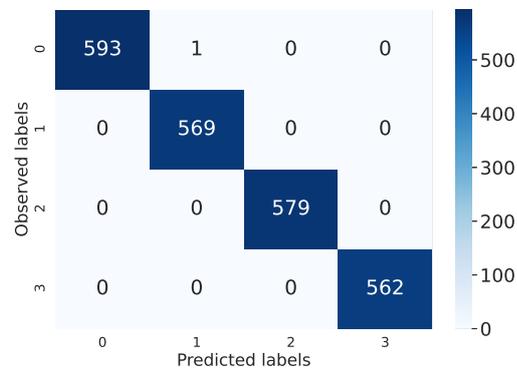


Figure B.11 Confusion matrix of SVM model with 10-bin histogram features to detect load imbalance with input length of 5 seconds

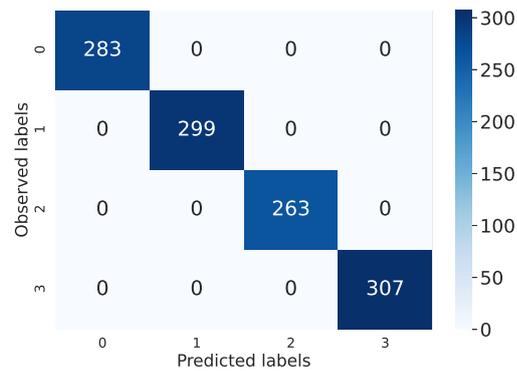


Figure B.12 Confusion matrix of SVM model with 10-bin histogram features to detect load imbalance with input length of 10 seconds

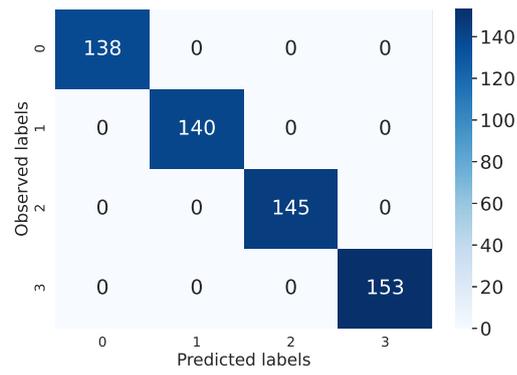


Figure B.13 Confusion matrix of SVM model with 20-bin histogram features to detect load imbalance with input length of 20 seconds

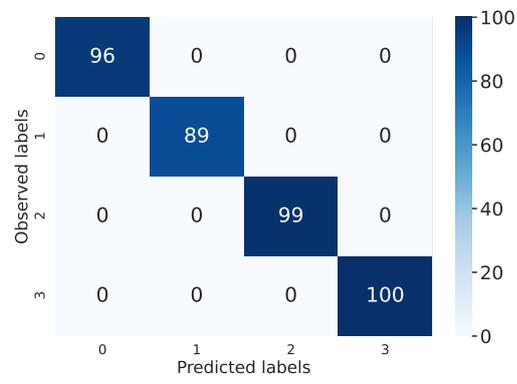


Figure B.14 Confusion matrix of SVM model with 10-bin histogram features to detect load imbalance with input length of 30 seconds

APPENDIX C

Hyperparameters for Best-performing Models

Table C.1 Hyperparameters for best-performing machine imbalance detection models - SVM models as (C, γ) , and XGBoost models as $(\text{depth}_{max}, w_{min}, \gamma)$

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
SVM_MV	1000, 1	10000, 1	10000, 1	1000, 1	100, 1	10, 1	10, 1
XGB_DFT	9, 4, 0.1	9, 5, 0.2	8, 3, 0.4	6, 2, 0.3	5, 2, 0.2	3, 2, 0.1	8, 1, 0.3
XGB_MV	7, 4, 0.3	8, 2, 0.2	9, 3, 0.4	9, 2, 0.3	5, 1, 0.1	7, 2, 0.1	6, 1, 0.1

Table C.2 Hyperparameters for best-performing load imbalance classification models - SVM models as (C, γ) , and XGBoost models as $(\text{depth}_{max}, w_{min}, \gamma)$

Model	Sequence Length (s)						
	0.5	1	2	5	10	20	30
SVM_H10	100, 0.1	1, 1	1, 1	1, 1	10, 0.1	1, 1	1, 1
SVM_H20	1, 1	1, 1	100, 0.1	10, 0.1	0.1, 0.1	0.1, 0.1	0.1, 0.1
SVM_UMAP	10, 1	0.1, 1	10, 0.1	0.01, 0.1	0.01, 0.1	1, 0.1	10, 1
XGB_DFT	8, 5, 0	7, 4, 0.3	4, 2, 0.3	3, 2, 0	3, 3, 0.2	5, 2, 0.3	6, 5, 0.2