



U LISBOA

UNIVERSIDADE
DE LISBOA

M.Sc. Data Analytics

Master's Final Work Internship Report

Optimizing Maintenance Processes With AI & Modern Techniques

Emma Grass Casalini

Supervisors:

Dr. João Afonso Bastos (ISEG, Academic Supervisor)

Pu Sun, PhD Candidate (AstraZeneca, Company Supervisor)

JANUARY - 2025

Abstract

One of today's most prominent issues is the environmental impact of industrial activities, with the manufacturing sector being a big contributor to greenhouse gas emissions, pollution, and waste. As industries face increasing pressure to adopt more sustainable practices, predictive maintenance has been presented as a part of the solution. This study, conducted at the biopharmaceutical company AstraZeneca, develops and evaluates a one-dimensional convolutional neural network (1D-CNN) and a combined one-dimensional convolutional neural network with bidirectional long short-term memory (1D-CNN-BiLSTM), for fault classification in complex industrial packaging machines. As part of predictive maintenance, fault classification can significantly reduce machine downtime by enabling early failure detection, improving productivity, and minimizing costs. Additionally, accurate fault diagnosis helps avoid unnecessary component replacements, reducing waste and supporting sustainable manufacturing.

Fault classification in packaging machines is an overlooked area of research. While existing studies mainly focus on fault classification in larger industrial systems using vibration sensors on isolated motors and in simplified test rigs, little research has been conducted on fault classification within small packaging machinery. Packaging machines present unique challenges due to their intricate mechanical structures, multiple moving parts, and changing operational conditions, such as varying loads and speeds, all of which contribute to complex vibration patterns. This study evaluates model performance across different machine speeds and assesses its ability to generalize fault classification under varying operational conditions. It also examines whether integrating data from multiple vibration sensors and incorporating long-term temporal context improves classification accuracy compared to models processing single-sensor data without long-term dependencies.

Key findings: Fault classification in small and complex packaging machines can be achieved with satisfying results using the proposed approach, which combines vibration sensors and deep learning models. This report also showcases the strengths of the 1D-CNN-BiLSTM model in accurately classifying faults under varying speeds, and in the presence of Gaussian noise. This study also highlights that adding a BiLSTM layer to the 1D-CNN structure is highly beneficial for fault classification on vibration data.

Keywords: *Deep learning, Fault classification, Segmented multivariate time series, 1D-CNN, Hybrid 1D-CNN-BiLSTM, Predictive maintenance*

Contents

1. Introduction	1
1.1 AstraZeneca	1
1.2 Background	1
1.3 Problem impact	2
1.4 Motivation	3
1.5 Research Contributions	4
1.6 Delimitations	4
2. Literature review	5
2.1 Predictive maintenance	5
2.2 Supervised machine learning	6
2.3 Convolutional neural networks	6
2.4 1D Convolutional neural networks	7
2.5 1D Convolutional Neural Network-Bidirectional Long Short-Term Memory	8
2.6 Black-box models	9
2.7 Summary	9
3. Methodology	10
3.1 Vibration sensors	10
3.2 Experimental System	10
3.3 Data acquisition	11
3.3.1 Data acquisition phases	11
3.4 Data description	13
3.5 Model Development	14
3.6 Configuration of 1D-CNN & 1D-CNN-BiLSTM	15
3.7 Training & Validation	18
3.8 Evaluation	19
3.8.1 Initial evaluation of sensor combinations	19
3.8.2 Initial evaluation of separate sensors	20
3.8.3 Baseline performance on separate speeds & baselines	20
3.8.4 Introducing Gaussian noise	20
3.8.5 Cross-speed generalization	21
3.8.6 Combined speeds	21
3.9 Performance Evaluation Metrics	22
4. Results & Discussion	24
4.1 Accuracy for separate sensors with 1D-CNN	24
4.2 Accuracy for different sensor combinations with 1D-CNN-BiLSTM	24

4.3 Separate baselines & speeds	25
4.3.1 1D-CNN-BiLSTM	25
4.3.2 Summary	28
4.4 1D-CNN	28
4.4.1 Summary	31
4.5 Robustness test with Gaussian noise	31
4.6 Cross-Speed Generalization	32
4.6.1 1D-CNN-Bi LSTM	32
4.6.2 1D-CNN	33
4.6.3 Summary	36
4.7 Combined Speeds	37
4.7.1 1D-CNN	37
4.7.2 Summary	37
4.7.3 1D-CNN-BiLSTM	37
4.7.4 Summary	38
5. Conclusions & Future Research	40
Appendix I - Literature Summary	42
Appendix II - Experimental Log	43
Appendix III - Model Configuration	44
Appendix IV - Gaussian Noise	45
References	46

Acknowledgments

I would like to thank Dexter Enrick De La Cruz Edep for your endless encouragement and care. Thank you for always standing by me and helping me through all the difficult moments and challenges faced.

I would also like to thank my family Susanna and Håkan, who have shaped the person I am today. Thank you for all the support and love you have given me throughout my life.

Finally, I would like to express my gratitude to my two supervisors, Pu Sun and João Afonso Bastos for their guidance and expertise which has been an invaluable part of the success of this work.

"If you've taken the devil into the boat, you must row him to shore"

Swedish proverb

List of figures

Figure 1. Data acquisition process _____	11
Figure 2. Example of data: Raw vibration data of ball bearing fault 2 (X, Y & Z-axis included)____	14
Figure 3. Pipeline for the classification model_____	15
Figure 4. Visualization of model (BiLSTM Layer - only applicable for BiLSTM model) _____	15
Figure 5. Training and validation Accuracy/Loss - 1D CNN-BiLSTM (baseline 2)_____	18
Figure 6. Training and validation Accuracy/Loss - 1D CNN (baseline 2)_____	19
Figure 7. Confusion Matrix - 1D CNN-BiLSTM. Baseline 1 (120 pz/min) _____	27
Figure 8. Confusion Matrix - 1D CNN-BiLSTM. Baseline 1 (75 pz/min)_____	27
Figure 9. Confusion Matrix - 1D CNN-BiLSTM. Baseline 2 (120 pz/min) _____	27
Figure 10. Confusion Matrix - 1D CNN-BiLSTM. Baseline 2 (75 pz/min) _____	27
Figure 11. Confusion matrix. 1D CNN - Baseline 1 (120 pz/min) _____	30
Figure 12. Confusion matrix. 1D CNN - Baseline 1 (75 pz/min) _____	30
Figure 13. Confusion Matrix - 1D CNN. Baseline 2 (120 pz/min) _____	31
Figure 14. Confusion Matrix - 1D CNN. Baseline 2 (75 pz/min) _____	31
Figure 15. Barplot of accuracy comparison over noise levels. 1D CNN vs 1D CNN-BiLSTM ____	32
Figure 16. Confusion Matrix - 1D CNN-BiLSTM. Cross-Speed Generalization (Baseline 1) ____	35
Figure 17. Confusion Matrix - 1D CNN-BiLSTM. Cross-Speed Generalization (Baseline 2) ____	35
Figure 18. Confusion Matrix - 1D CNN for Cross-Speed Generalization (Baseline 1)_____	35
Figure 19. Cross-Speed Generalization (Baseline 2) _____	35
Figure 20. Confusion Matrix. 1D CNN-BiLSTM. Combined Speeds & baseline 1. _____	39
Figure 21. Confusion Matrix. 1D CNN. Combined Speeds & baseline 1. _____	39
Figure 22. 2% Gaussian noise added to Baseline 2 (X-axis) _____	45
Figure 23. 5% Gaussian noise added to Baseline 2 (X-axis) _____	45
Figure 24. 10% Gaussian noise added to Baseline 2 (X-axis) _____	45

List of tables

Table 1. Introduced faults - Phase 1 - Position indicates at what sensor fault was induced. _____	12
Table 2. Introduced faults - Phase 2 - Position indicates at what sensor fault was induced. _____	12
Table 3. Classification report - 120 Speed & baseline 2 - 1D CNN-BiLSTM _____	26
Table 4. Classification report - 75 Speed & baseline 2 - 1D CNN-BiLSTM _____	26
Table 5. Classification report for unseen data (75 Speed) - Hybrid 1D CNN _____	26
Table 6. Classification report for unseen data (75 Speed) - Hybrid CNN-BiLSTM _____	26
Table 7. Classification report - 120 pz/min & baseline 2 - 1D CNN _____	29
Table 8. Classification report for unseen data (75 pz/min) - 1D CNN _____	29
Table 9. Classification report - 120 pz/min & baseline 1 - 1D CNN _____	30
Table 10. Classification report - 75 pz/min & baseline 1 - 1D CNN _____	30
Table 11. Classification report - Cross-Speed Generalization- 1D CNN _____	34
Table 12. Classification report - Cross-Speed Generalization - 1D CNN-BiLSTM _____	34
Table 13. Classification report - Cross-Speed Generalization- 1D CNN _____	34
Table 14. Classification report - Cross-Speed Generalization - 1D CNN-BiLSTM _____	34
Table 15. Classification report - Combined Speeds - 1D CNN _____	38
Table 16. Classification report - Combined speeds - 1D CNN-BiLSTM _____	38
Table 17. Literature summary _____	42
Table 18. Experimental Log _____	43
Table 19. Model configuration for 1D CNN & 1D CNN-BiLSTM _____	44

List of equations

Equation 1. Convolutional Operation	16
Equation 2. Batch Normalization	16
Equation 3. L2 Regularization	17
Equation 4. Adam Optimizer	17
Equation 5. Precision	22
Equation 6. Recall	22
Equation 7. F1-Score	23
Equation 8. Accuracy	23

List of abbreviations

AI - Artificial Intelligence

AZ - AstraZeneca

BiLSTM - Bidirectional Long Short-Term Memory

CNN - Convolutional Neural Network

CWRU - Case Western Reserve University

Conv1D - One Dimensional Convolution

CSV - Comma-Separated Values

Hybrid 1D CNN-BiLSTM - Hybrid One Dimensional Convolutional Neural Network -
Bidirectional Long Short-Term Memory

LSTM - Long Short-Term Memory

Leaky ReLU - Leaky Rectified Linear Unit

PdM - Predictive Maintenance

Pz/Min - Pieces Per Minute

ReduceLROnPlateau - Reduce Learning Rate On Plateau

ReLU - Rectified Linear Unit

RMS - Root Mean Square

SEK - Swedish Crowns

SoftMax - Soft Maximum

TPM - Total Productive Maintenance

1D CNN - One-Dimensional Convolutional Neural Network

2D CNN - Two-Dimensional Convolutional Neural Network

RUL - Remaining Useful Life

Chapter 1

1. Introduction

1.1 AstraZeneca

AstraZeneca is a global pharmaceutical company with production sites all over the world. AstraZeneca's campus in Södertälje, Sweden is one of the world's largest pharmaceutical manufacturing units, producing more than 30 different pharmaceuticals to over one hundred markets. Each year, Astras manufacturing site at Snäckviken produces over 12 billion tablets and 40% of AstraZeneca's medicines are manufactured at this campus. AstraZeneca is one of Sweden's largest export companies and manages approximately 8% of the total Swedish merchandise exports. In 2022, the company exported pharmaceuticals from Sweden to a value of over 152 billion Swedish crowns (SEK).

1.2 Background

As a big manufacturing company AstraZeneca is driven by their responsibility and commitment to include sustainability in every aspect of their operations, with the intent of accelerating the transition toward sustainable, net-zero healthcare. In the area of maintenance, broad improvement work is underway through a total productive maintenance (TPM) project. This approach has been introduced to improve AstraZeneca's maintenance procedures and to increase competence about industrial maintenance. TPM has also been introduced with the intent of gathering more knowledge about AstraZeneca's machines and their components to understand what factors affect the machines wear and tear. AstraZeneca also recognizes the opportunity of exploring modern technologies like data analysis and artificial intelligence, to help with predicting AstraZeneca's maintenance needs and replacing the preventive maintenance actions that are currently being employed in the production.

Vibration sensors are currently used at AstraZeneca's production sites Snäckviken and Gärtuna, where they monitor cooling water pumps and bearings of electric motors in the ventilation systems' fans. Although the method of condition monitoring using vibration sensors is new at AstraZeneca, it has already displayed good results. The sensors have been able to detect early signs of bearing damage by identifying abnormal vibration patterns.

This proactive approach has minimized unplanned downtime and allowed for more efficient scheduling of repairs and maintenance. Vibration sensors are also mounted on larger machines in production, for example, on filling machines. However, AstraZeneca needs to gain experience in using vibration sensors and measuring vibrations on smaller and more complex machines, like their packaging machines, leaving room for further research and exploration. Previous experiments concerning machine learning have been performed with promising results. This includes an experiment that used black-box models for predicting specific failure modes by finding different patterns in production data leading up to failures, but since the equipment very rarely runs to failure, the project had a restricted amount of data. A second project which involved developing a model to predict the probability of a successful outcome for the next batch, specifically focusing on a cleaning cycle has also been previously performed with promising results. Projects regarding anomaly detection in machines have also been and are currently being explored with promising results.

1.3 Problem impact

The manufacturing sector is a main contributor to the many challenges of sustainability and has a significant impact on the environment by producing greenhouse gasses, pollution, and waste. As stated by Patalas-Maliszewska and Łosyk [1], despite previous warnings from researchers, manufacturing industries are still faced with difficulties in making their manufacturing processes more sustainable. In response, a study performed by researchers Polese et al [2] emphasizes that strategies like predictive maintenance (PdM) can make it possible for manufacturers to improve both their profitability and sustainability at the same time. One particularly promising approach within PdM is fault classification and fault detection. Research presented by Rao et al [3] shows that fault classification paired with deep learning can make it possible to identify faults faster and more accurately in machinery, leading to reduced downtime and reduced waste. The reliability of the company can also be improved by reducing the downtime of the machinery, which is especially important in critical industries like healthcare, where a timely delivery of medications is crucial for the wellbeing of millions of patients. These developments not only improve the operational efficiency of the manufacturing companies but can also help them to become more sustainable and resilient long term.

1.4 Motivation

Based on the identification of using artificial intelligence and data analysis for predictive maintenance, this study aims to explore the possibility of classifying faults in a packaging machine, by applying AZ's current vibration measurement techniques together with deep learning algorithms. As an initial study, this research focuses on establishing a foundation in fault detection and classification for future research conducted at AZ. This study investigates whether the current vibration measurement methods can be effectively adapted to smaller, more complex machines, building on the success of previous studies involving vibration sensors and machine learning.

This study also aims to compare the performance of the models under varying operating conditions and evaluate how well they can generalize fault classification under two different speeds and two different baselines. This study will also compare models with varying degrees of complexity and evaluate whether a model that processes data from multiple sensors and uses long-term temporal understanding improves fault classification, compared to a model that processes data from a single sensor without long-term temporal understanding. The following objectives have been identified:

1. *Develop two data-driven predictive models to perform fault classification on different machine faults, machine speeds and baselines.*
2. *Explore if the currently used vibration measurement can be applied to smaller machines.*
3. *Explore if a model that processes multiple sensor data and uses long-term temporal understanding improves fault classification, compared to a model that processes single-sensor data without long-term temporal understanding.*

1.5 Research Contributions

This study contributes to fault classification in packaging machines, which is an overlooked area in fault classification. A big part of the existing research focuses on vibration-based fault classification in larger industrial systems with the vibration sensors being directly placed on, for example, single motors and pumps in simplified test-rigs. There is also little research conducted where the vibration sensors are located inside a packaging machine. Much of the data in previous research studies are also being recurrently used, highlighting a lack of availability of original data. Packaging machines have unique operational characteristics which are very separate from simplified test rigs, since they often experience varying loads, speeds, and complex mechanical movements due to the many moving parts that can impact the vibration patterns.

Another overlooked area in fault classification, especially for packaging machines, is cross-speed generalization and combined-speed research, which ensures that fault classification models perform accurately across different operating speeds. By collecting data from a complex packaging machine, it is ensured that the data used in this study is original and tailored to the specific challenges of fault classification in packaging machines. This study fills a gap and provides a unique foundation for testing and validating deep learning models in this area, improving the reliability of fault classification in packaging machines.

1.6 Delimitations

This study is limited to a labeling machine that is set up in an isolated and controlled environment. The machine used in this research is placed outside of the real production and operates as a standalone unit. It is not part of a complete production process and did not interact with any upstream or downstream equipment during the experiments, hence the performance and functionality of the machine used in this experiment does not reflect the performance of a complete packaging line. The findings and results of this study may hence differ from performance in full-scale production settings. It should also be noted that the sensors used in this study have a frequency range of 10-1000 Hz. This range limits the detection of faults or behaviors that occur outside of this frequency spectrum.

Chapter 2

2. Literature review

2.1 Predictive maintenance

PdM refers to a collection of data-driven strategies aimed at assessing the condition of equipment with the intent of accurately predicting when maintenance should be performed. Using smart scheduling, PdM helps to optimize maintenance activities, reduce the likelihood of unexpected equipment failures, and minimize downtime [4]. In the PdM approach, data is collected over time, to monitor the condition of equipment with the help of predictive analytics [5]. PdM aims to identify equipment failures before they become critical, allowing manufacturers to optimize their maintenance procedures [6]. PdM can also help to improve sustainability in the manufacturing industry by extending the lifetime of components and ensuring that all resources are used efficiently [5]. To identify and classify different machine faults, various machine and deep learning methods are often used in PdM, such as decision trees, support vector machines and neural networks.

The use of classification algorithms can promote proactive decision-making by promoting actionable insights that can help companies prioritize maintenance actions based on the characteristics of the identified fault. Classification algorithms can also help companies to move from reactive maintenance to proactive maintenance. In fault classification within predictive maintenance, there are two primary methodologies, logical classification and algorithmic classification. The logical classification utilizes expert knowledge to organize faults following structured frameworks [7][8][9]. In algorithmic classification, machine learning algorithms are used to classify faults by analyzing patterns and features in the data. Both methods contribute with their own advantages to fault classification and should be utilized based on different needs within the area of maintenance strategies [7] [8].

Chapter 2

2. Literature review

2.1 Predictive maintenance

Predictive maintenance (PdM) refers to a collection of data-driven strategies aimed at assessing the condition of equipment with the intent of accurately predicting when maintenance should be performed. Using smart scheduling, PdM helps to optimize maintenance activities, reduce the likelihood of unexpected equipment failures, and minimize downtime [4]. In the PdM approach, data is collected over time, to monitor the condition of equipment with the help of predictive analytics [5]. PdM aims to identify equipment failures before they become critical, allowing manufacturers to optimize their maintenance procedures [6]. PdM can also help to improve sustainability in the manufacturing industry by extending the lifetime of components and ensuring that all resources are used efficiently [5]. To identify and classify different machine faults, various machine and deep learning methods are often used in PdM, such as decision trees, support vector machines and neural networks.

The use classification algorithms can promote proactive decision-making by promoting actionable insights that can help companies prioritize maintenance actions based on the characteristics of the identified fault. Classification algorithms can also help companies to move from reactive maintenance to proactive maintenance. In fault classification within predictive maintenance, there are two primary methodologies, logical classification, and algorithmic classification. The logical classification uses expert knowledge to organize faults following structured frameworks [7][8][9]. In algorithmic classification, machine learning algorithms are used to classify faults by analyzing patterns and features in the data. Both methods contribute with their own advantages to fault classification and should be used based on different needs within the area of maintenance strategies [7] [8].

2.2 Supervised machine learning

Supervised machine learning algorithms are used in artificial intelligence to predict outcomes by learning from labeled data [10]. The primary objective of supervised learning is for the user or so-called supervisor to train a model using the labeled data, enabling the model to make predictions on unseen or future data [11]. The desired output signals, called labels, are already known in supervised learning. Supervised learning is applied when the supervisor aims to predict a specific outcome based on a given input, using examples of input-output pairs. The supervisor develops a machine-learning model from these pairs, which form the training set [12]. A data set is usually divided into training and testing sets for accuracy evaluation [13].

Although building the data set typically requires human effort, the process of supervised learning automates and speeds up tasks that would otherwise be time-consuming or impractical [12]. Classification is a form of supervised learning technique aimed at predicting a class label selected from a predefined set of options [12]. Classification is an iterative process that identifies and organizes data objects into predefined categories or labels [14]. Classification can be applied to both structured and unstructured data. Its primary objective is to map an unknown pattern to a recognized class [14]. An example of this is classifying emails as "spam" or "not spam" [14] or, for example, classifying images of animals like cats or dogs, based on visual features. In this study, the predefined options correspond to the various machine faults that were created, making this a multiclass classification task.

2.3 Convolutional neural networks

CNNs are deep learning algorithms that are applied in various areas, including fault detection, image classification, and speech recognition [15]. A typical CNN is composed of four different layers. The convolutional layer, the pooling layer, the fully connected layer, and the activation layer [15]. Each layer has its distinct task. The convolutional layer employs kernel filters to compute the convolution of the input data, extracting features from it. The convolutional layers task is to create feature maps that capture patterns in the data. The pooling layer reduces spatial dimensions [15]. The third layer is the fully connected layer which links the extracted features to final output predictions [15]. The fourth layer contains the activation function.

The activation function's role is to determine the output of the neural network, which in the context of fault classification, classifies the data into specific fault categories. A few examples of the more commonly used activation functions are for example ReLU, Sigmoid, and Tanh [15].

2.4 1D Convolutional neural networks

The 1D-CNN is similar to the regular CNN, also known as the 2D-CNN, and differs in the way that its filters move across the data [16]. 1D-CNNs are more commonly used for time series data, where each row represents the raw signals captured at a specific time, whilst each column typically corresponds to data from a different channel, for example, vibration signals in the X, Y, or Z directions [16]. Since the columns represent independent channels, the 1D filter slides only vertically along the time dimension. At any given moment, the filter spans all columns, and the filter's height determines how many time steps are included in each convolution calculation [16]. A 2D-CNN on the other hand, manages data organized in a grid [12]. The 2D-CNN filter moves across the grid both horizontally and vertically, unlike the 1D-CNN, and captures patterns in two dimensions [16]. The 2D-CNN and the 1D-CNN both use filters with shared weights that slide across the data, which allows the network to systematically detect patterns [16].

Based on the approach of Chen et al [17] and Chao et al [16] this research proposes a 1D-CNN architecture designed to process raw vibration data without any prior denoising. Raw data preserves all the inherent patterns and features in the vibration signals, including subtle variations and noise that might indicate faults. Processing or filtering data can remove or alter these features. Raw data also allows CNN to discover features independently. Traditional fault detection typically requires artificial feature extraction, which can compromise accuracy and introduce bias, but the 1D-CNN enables automated feature extraction directly from the raw signal and can overcome this shortcoming [17]. By allowing the model to learn the most relevant features from the data itself, all useful information is preserved, which reduces the risk of bias introduced by human assumptions. This approach enables the 1D-CNN to identify complex relationships in the data that the traditional methods risk to overlook. The proposed approach builds on the work of Chen et al [17] who demonstrated high fault diagnosis accuracy without denoising the vibration data under various load conditions.

Additionally, Chao et al [16] introduced a multi-sensor technique using accelerometer data, showing that a 1D-CNN model can achieve accuracy above 98.87%, with strong robustness in the presence of noise. The validation approach was further enhanced through 10-fold cross-validation. Applying 10-fold cross-validation on different load conditions ensures that the model's performance is robust and consistent across various operational scenarios, enhancing the reliability of the results. The Adam optimizer was chosen, as used by Chen et al [17] and the SoftMax activation function was used in the output layer, as it is effective for multiclass classification tasks. The activation function ReLU was used in the hidden layers, based on Chao's work [16]. ReLU outputs zero for negative values, but since batch normalization, which ensures that the activations have zero mean and unit variance was applied after each Conv1D layer, both negative and positive values were kept intact before they reached the ReLU activation. As a result, the negative values were not entirely discarded but instead transformed into a balanced range, ensuring they indirectly contributed to the gradient calculations during backpropagation. Leaky ReLU was also evaluated, but using ReLU with batch normalization gave the best results it was consequently decided to proceed with the pure ReLU-approach.

2.5 1D Convolutional Neural Network-Bidirectional Long Short-Term Memory

In the BiLSTM model the 1D-CNN part processes sequential data, such as time-series. It works by applying filters to the data to automatically detect patterns, both local, within small windows, and global, across larger windows [18]. The added BiLSTM layer is used to capture long-term temporal dependencies within the data and are particularly important when the order and timing of events matter. The bi-directional part of the BiLSTM allows the model to process data from both the past and the future, helping the model to understand the full context of the data. When combined, the 1D-CNN and BiLSTM layers enable the model to both recognize patterns in the data and understand how they change over time [18]. This study also proposes that a BiLSTM model should be explored based on the work of Choudakkanavar and Mangai [19], who demonstrated the effectiveness of this model in fault diagnosis of bearings using raw vibration data, achieving a classification accuracy of 99.84%. Their study displayed that combining the 1D-CNN for feature extraction, with the BiLSTM layer for capturing temporal dependencies can significantly improve the accuracy of fault classification tasks. This approach uses the 1D-CNN's ability to extract patterns from time-series sensor data, whilst the BiLSTM layer processes the sequence data in both forward and backward directions, allowing for a more comprehensive understanding of temporal dependencies.

2.6 Black-box models

CNNs belong to a group of models referred to as “black-box models”. These models have a high level of complexity and a low level of explainability, meaning that there is a lack of insight into their decision-making process [20]. A black-box model processes data in such a way that it makes it difficult for humans to understand how the input variables are combined to create the predictions [21]. When it comes to CNNs, they have thousands of weighted connections that are tuned during training and that rely on combinations of features that are unknown to humans. Considering the nature of black-box models is crucial, especially in high-stakes industries like healthcare, where understanding the decision-making process is essential for validation.

2.7 Summary

In summary (see Appendix I, Table 17), the proposed 1D-CNN and 1D-CNN-BiLSTM models have demonstrated strong performance in fault classification, achieving accuracy rates between 98.4% and 99.8% in previous studies. The prior research involving multiple sensors indicates that combining a diverse set of data sources can significantly enhance model performance. While these models have proven suitable for fault classification using complex vibration data, it is important to note some limitations of the previous research. None of the reviewed studies used sensor data collected from machinery that contains many moving parts. Additionally, two studies relied on the same dataset (CWRU) and used accelerometers, which differ from the vibration sensors used in this project.

Chapter 3

3. Methodology

3.1 Vibration sensors

The sensors used in this research capture features such as RMS, peak, peak-to-peak, crest factor, kurtosis, and skewness. The sensors are triaxial and measure vibration along three axes, X, Y, and Z, to capture a 3D picture of the machine's movement. Capturing this 3D movement is crucial because machines vibrate in multiple directions due to factors like imbalances, misalignment, and wear. By monitoring data from the three axes, the sensors can enable accurate detection of various machine faults. The sensors measure within a range of 10-1000 Hz, making them especially effective for detecting low-frequency faults, but less effective for high-frequency faults.

3.2 Experimental System

Three wireless vibration sensors were mounted inside the packaging machine and placed near its critical components. The sensors were installed with expert help from technicians and engineers, making sure that the sensors were strategically positioned to monitor vibrations where detection of anomalies is crucial. The first sensor, referred to as position 1, was installed on a front belt component. The second sensor, referred to as position 2, was installed next to the first roller component of the machine. These two sensors are located close to many moving parts, for example, the spacer component, the main belt, and horizontal belts. The third sensor, referred to as position 3, was installed on the other end of the machine, further away from sensor 1 and sensor 2. Sensor 3 was also positioned next to the second and last roller component and under a horizontal belt. The two roller components are connected to one conveyor each.

Placing the vibration sensors on each end of the machine covers a broader range of the machine and can help track how the vibration propagates. The placements of the two sensors, positioned near the roller components (sensor 2 and sensor 3), were selected due to a recent breakdown involving ball bearings on a similar roller component from a similar packaging machine in the real production. The sensor placed on the front belt was chosen because this area is a complex part of the machine, interconnected with several other components. Additionally, there have also been prior faults near this position in the real production environment.

3.3 Data acquisition

The vibration sensors were manually activated during data collection, which led to a few seconds delay in time between the sensors. The sampling frequency of the vibration sensors were 2560 samples/second, and the sampling intervals were ranging between two and five minutes. During the data acquisition, the vibration data is transferred wirelessly via Wi-Fi to an entity server, which acts as an intermediary between the vibration sensors and AstraZeneca's SQL database. The entity server facilitates tasks such as noise reduction, performing fast Fourier transformation, normalization, and converting acceleration measurements into vibration velocity. After processing, the data is transferred to the SQL database, which serves as the central repository for managing the collected information. Users have the option to access both the raw and processed data through a remote desktop (see Figure 1). However, this study only uses the raw data. The data is stored as text files in the remote desktop environment. Each text file corresponds to a single measurement segment and does not provide any exact timestamps. Each text-file contains approximately 20,000 individual data points. The text files were cleaned from information like the date and time of extraction. Once cleaned, the data points from each text file were systematically organized and transferred into Excel files.

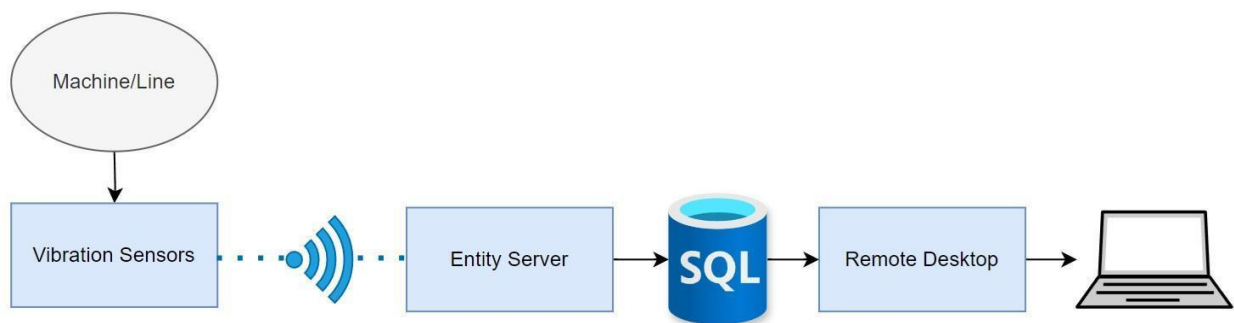


Figure 1. Data acquisition process

3.3.1 Data acquisition phases

The data acquisition process was conducted in two phases, to ensure complete data gathering. For the complete experimental log, see Appendix II, Table 18. In phase one old machine components were replaced with new components, but the machine was left with the original spacer component. Data was collected at two different speeds, 75 and 120 pz/min. Five different faults were introduced in addition to the baseline (see Table 1).

The two different speeds were chosen because packaging lines run at different speeds in the real production environment, and training a model on multiple speeds can help it learn to identify fault patterns that are speed-invariant and consistent across different operational conditions, resulting in the model being better at generalizing to other, untested speeds in future.

Fault	Description of faults (collected at 75 Pz/min & 120 pz/min speed)
BL	Baseline with new components & unstable spacer component
SGPI	Small gear, old and worn component with no extra damage, position 1
BBI	Ball bearing fault 1, impact on balls with induced clicking sound
BBII	Ball bearing fault 2, graveled bearing with less resistance in rolling
LBTPIII	Looseness in the belt, induced looseness in the belt, position 3
TB	Overtightened horizontal belt, position 1

Table 1. Introduced faults - Phase 1 - Position indicates at what sensor fault was induced.

In the second phase, a new and less noisy baseline was created by replacing the old spacer component in the machine. Three different faults were introduced in addition to the new baseline (see Table 2). The baseline data represents vibrations when the machine is operating under normal conditions without any introduced faults. In both phases faults of different severity levels were introduced. For example, looseness faults represent errors that may arise from incorrect handling during maintenance. More severe faults like component damage were also introduced in both phases, for example, various levels of damage to ball bearings.

Fault	Description of faults (collected at 75 pz/min & 120 pz/min speed)
BL	Baseline with exchange of spacer component from back to front
SPACER	Unstable spacer component
SGPI	Bearing fault on small gear, position 1
LBTPIII	Looseness in the belt, position 3

Table 2. Introduced faults - Phase 2 - Position indicates at what sensor fault was induced.

3.4 Data description

The extracted data were compiled into 12 CSV format data sets. These datasets include two sets each for the 75 and 120 pz/min speeds, with one set representing the baseline for each speed. Additional datasets were created for each individual sensor, as well as for three different sensor combinations. The datasets used with the 1D-CNN model consist of the X-axis, Y-axis, Z-axis. A fourth column named Class was also introduced, which is the abbreviated name of the machine fault. For the BiLSTM model, the datasets also include additional columns tailored to its ability to capture long-term temporal relationships. These datasets contain columns indicating the sensor, the axis and a segment-index column that provides time indexing. The time indexing was introduced as a solution to address the gaps caused by the time-series data being segmented. The segment-index ensures that the temporal order is preserved, allowing the BiLSTM layer to process and understand dependencies across segments. This index helps the BiLSTM layer to understand how patterns change over time, even when the data is presented as chunks, due to the segmentation.

When it comes to introducing multiple sensors into the dataset, the BiLSTM model benefits more because it can learn not only the temporal development in the data within individual sensors, but also the relationships between sensors. This is not something that the 1D-CNN model is able to do, as it lacks the capability to model temporal dependencies between data from multiple sensors. The 1D-CNN can detect local patterns within each sensor's data but cannot process how signals from different sensors are related or how they change over time, in relation to one another. To improve the training of the models, two combined datasets were also created, one for each model. These combined datasets include data from both the 75 and 120 pz/min speed, allowing the models to learn from a broader range of examples. The separate datasets allow the models to focus on speed-specific behaviors, while the combined datasets enable them to identify patterns that are consistent across speeds, as well as those unique to each speed. All the data are raw time-series signals, with the Y-axis representing acceleration (m/s^2) and the X-axis representing time. For visualization of data, see Figure 2.

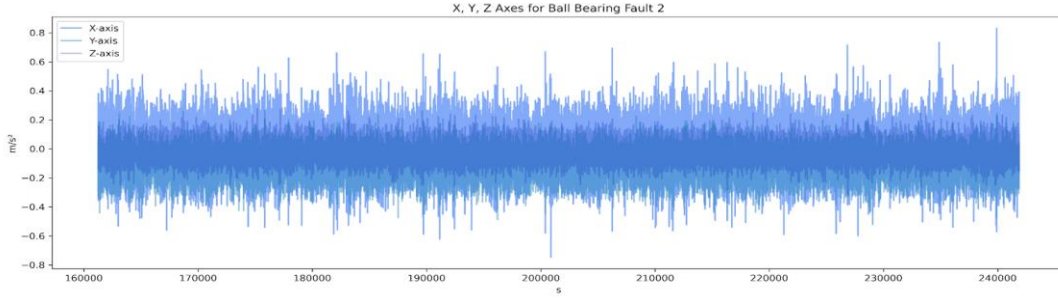


Figure 2. Example of data: Raw vibration data of ball bearing fault 2 (X, Y & Z-axis included)

3.5 Model Development

Initially, the data was prepared by being divided into sliding windows that create sequential samples for training. Each window has a fixed length of 400 time-steps, and the windows overlap with a stride of 100. This sliding-window approach generates multiple sequences of features from the data, where each sequence is associated with a corresponding class label. The class labels are first encoded into numerical values using Label Encoder and then transformed into one-hot encoded vectors, which represent the labels as binary vectors. The one-hot encoding enables the model to output probabilities for each fault class. Finally, the data set is split into training, validation, and test sets. The test set holds 20% of the data, while the remaining 80% is used for training and validation.

For cross-speed generalization a whole data set (75 pz/min) was used as the test set, after training and validating at the 120 pz/min speed. To address any class imbalance in the data set, under sampling was applied to ensure that the test set contains an equal number of instances from each fault class. Stratified 10-fold cross-validation was used for evaluation of the model. This technique divides the data into multiple folds, maintaining the same class distribution across each fold. The model is trained and validated on different folds, ensuring that the evaluation is not biased by any data split. This process helps with obtaining a reliable estimate of the model's performance across different subsets of the data. For visualization of classification pipeline, see Figure 3.

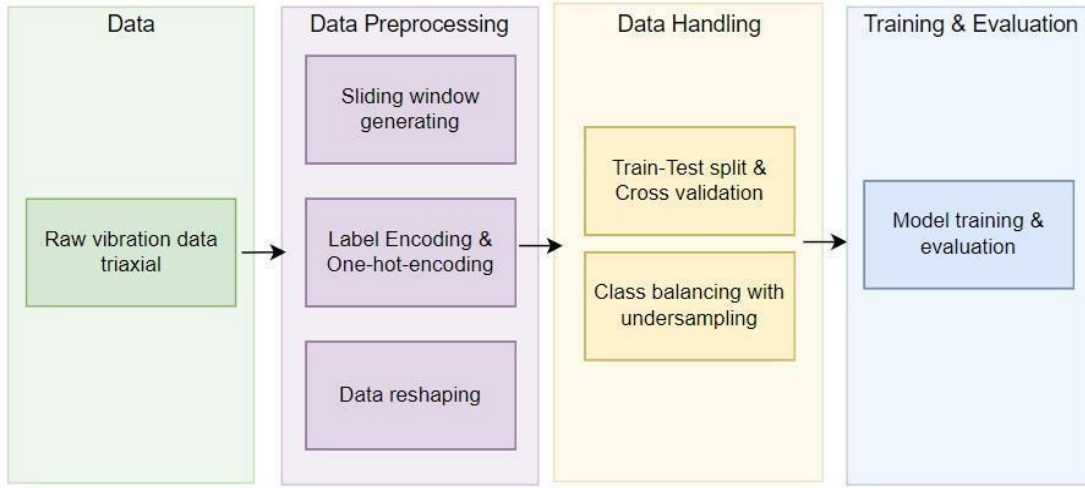


Figure 3. Pipeline for the classification model

3.6 Configuration of 1D-CNN & 1D-CNN-BiLSTM

For the configuration of the models see Appendix III, Table 4, and for visualization see Figure 4. The tuning of hyperparameters in this study was performed through a trial-and-error approach which involved initially selecting baseline values for each hyperparameter, and then systematically adjusting the values by increasing or decreasing them. If an increase or decrease in the parameter resulted in no improvement, the parameter was set to its previous setting. All model development and configuration were carried out in the Python environment. For comparison of the two models, each model was optimized separately.

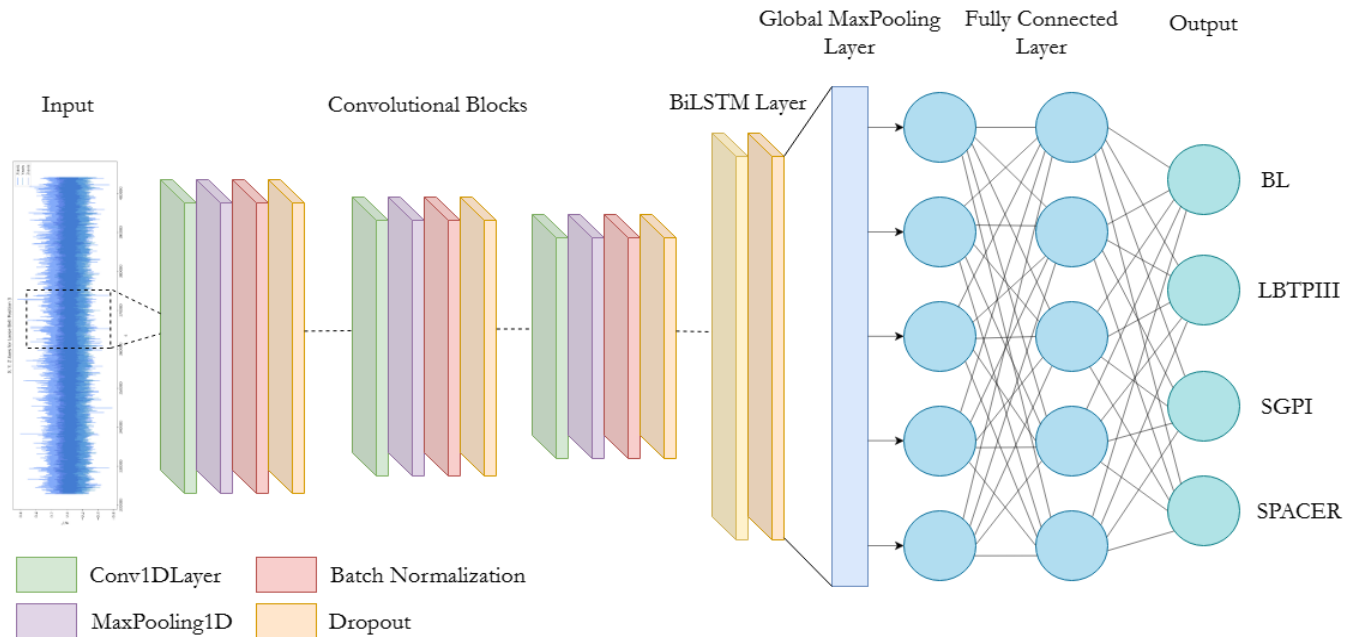


Figure 4. Visualization of model (BiLSTM Layer - only applicable for BiLSTM model)

Three Conv1D layers were used to extract features from the data. In the first layer, the 1D-CNN model uses 128 filters with a kernel size of 10, while the BiLSTM model uses a kernel size of 15. Larger kernel sizes like these help the model capture broader patterns by considering a larger portion of the input data at once, which allows the model to identify broader features. Following layers with 64 filters (kernel size 5 and 10) and 32 filters (kernel size 3 and 8) were then added to progressively capture the more refined features. As seen in Equation 1, the convolution operation in the Conv1D layer calculates the output y_t at position t , by taking a weighted sum of a segment of the input data x , which is determined by the kernel w , its size k , and the stride s . The bias term b is added to increase the flexibility which allows the model to capture the patterns.

$$y_t = \sum_{i=1}^k x_{t \cdot s + i - 1} \cdot w_i + b \quad (1)$$

After each convolutional layer, a pooling layer was added to help reduce the dimensionality of the data. The first pooling layer uses a pool size of 4, and the second layer uses a pool size of 2. The pooling is applied through max pooling, which identifies the highest value in each pooling region [22]. Batch normalization was added after every convolutional layer to standardize the outputs, this was done to allow faster convergence. This addition also enables the use of higher learning rates and reduces sensitivity to weight initialization, as noted by Ioffe and Szegedy [23]. As seen in Equation 2, the batch normalization standardizes the output of a layer to have zero mean and unit variance, making the learning process more stable. Epsilon is a small constant which is added to the denominator to prevent division by zero. The scaling parameter gamma and the shifting parameter beta are learnable variables that allow the network to adjust the normalized values as needed.

$$\hat{y}_t = \gamma \cdot \frac{y_t - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (2)$$

L2 regularization with a weight decay of 0.001 was applied to retain all information from each sensor axis. The regularization term adds a penalty to the loss function that is proportional to the sum of the squares of the model weights. This ensures that the weights remain small but are never driven all the way to zero, which results in that the model [24] is performing no feature selection. Since this approach allows the model to retain all information from each vibration sensor axis, it was applied to keep all the unique patterns and interdependencies from each axis that could be important for making accurate predictions.

The L2 regularization also preserves the contribution of all features, avoiding the risk of oversimplification and ensures that the model captures the full complexity of the data without overfitting. As seen in Equation 3, $L(w)$ is the original loss function, λ is the regularization parameter, which controls the trade-off between the original loss and the penalty term.

$$L2 = L(w) + \lambda \sum w_i^2 \quad (3)$$

For optimization of the two models the Adam optimizer with a learning rate of 0.001 was applied to update the weights. As stated by Kingma and Ba [25], the Adam optimizer is computationally efficient and suited for the noisy gradients that vibration data displays. The Adam optimizer combines the benefits of the momentum technique that helps speed up training, and the adaptive learning rates technique that adjusts the learning rate based on the model's performance. This allows Adam to update the weights and help the model learn faster, while also avoiding issues like overshooting the optimal values. As seen in Equation 4, θ_t represents the model's weights at time step t , and θ_{t+1} represents the updated weights. α is the learning rate that controls the size of the step taken during optimization. \hat{m}_t and \hat{v}_t are bias corrected first and second moment estimates, which are derived from the gradient of the loss function with respect to the weights. Epsilon is a small constant added to the denominator to avoid division by zero.

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (4)$$

Lastly, the ReduceLROnPlateau callback with a patience setting of 5 was applied to decrease the learning rate if there is no improvement in validation loss for five epochs. This allows the model to use high learning rates in the early stages of training while at the same time ensuring more gradual updates as it approaches optimal values and hence prevents overshooting. When creating the BiLSTM model a bidirectional LSTM layer with 64 units and `return_sequences = True` was added. This configuration allows the model to retain the full temporal sequence output from the BiLSTM layer and to learn temporal dependencies in both directions, left to right and right to left.

3.7 Training & Validation

The training and validation accuracy and loss were initially monitored to evaluate potential overfitting or underfitting, and to monitor the model's learning behavior. The BiLSTM model is displaying strong performance during training and validation, as it achieves perfect accuracy and zero loss on both the training and validation sets. These behaviors indicate that the model is effectively capturing the underlying patterns in the data, allowing it to generalize well to unseen data. The decrease in validation loss over time, paired with the increase in validation accuracy, suggests that the model is gradually improving its learning. The stability of training accuracy and loss at 1 and 0 highlights that the model has successfully converged and achieved a balance in its ability to handle both the training and validation datasets. For visualization, see Figure 5.

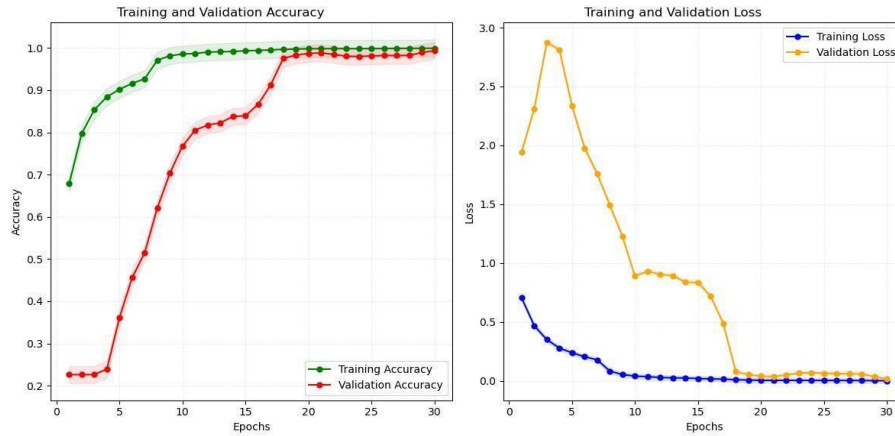


Figure 5. Training and validation Accuracy/Loss - 1D CNN-BiLSTM (baseline 2)

Similarly, the 1D-CNN model achieves near-perfect accuracy and demonstrates strong performance in capturing the most significant features from the data. However, the 1D-CNN model does not stabilize as consistently as the BiLSTM model. As seen in Figure 6, the training loss steadily decreases during the training, but it never manages to reach zero. This behavior suggests that the model is continuing to adjust its parameters, and is potentially struggling, compared to the BiLSTM model. The training and validation accuracy of the 1D-CNN model is displaying strong performance, although it notably never manages to reach 1, which suggests that the model has not fully learned certain patterns in the data.

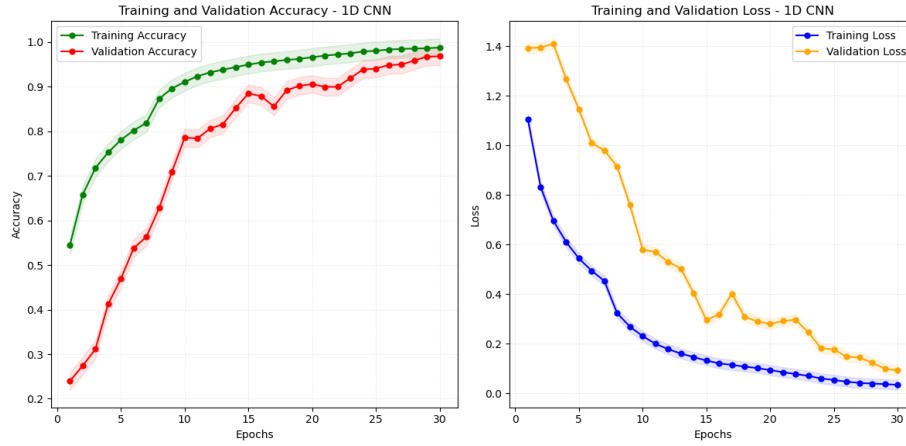


Figure 6. Training and validation Accuracy/Loss - 1D CNN (baseline 2)

3.8 Evaluation

3.8.1 Initial evaluation of sensor combinations

During the initial evaluation of the BiLSTM model, three different sensor combinations were evaluated: sensor 1, sensor 1 & 3 and sensor 1, 2 & 3. This evaluation was performed to investigate whether adding more sensors leads to significant improvement in performance, or if the additional sensors contribute with noise and redundant information, leading to a decrease in performance. Sensor 1 was chosen as the single-sensor option since most faults were introduced near this sensor. Faults occurring near a sensor tend to produce stronger signals because the sensor is closer to the source of the fault, allowing it to detect the vibrations more clearly. As the distance between the sensor and the fault increases, the signal tends to weaken, as the sensor picks up less of the fault's impact.

The location of sensor 1 also has the highest chance of minimizing potential interference and noise that could come from environmental factors or signals from other components in the machine. A sensor located farther away, like sensor 3, could potentially pick up more of this unwanted noise, which in return could lead to degradation of the data quality and impact the result of the fault classification negatively. The evaluation of different sensor combinations was performed to optimize the sensor setup for the following tests which included combined speeds, separate speeds & baselines, and the cross-speed-generalization. The evaluation was performed with baseline 2 on the 120 Pz/min speed.

3.8.2 Initial evaluation of separate sensors

The 1D-CNN was evaluated on the separate sensors 1, 2, and 3. This initial test was conducted to identify which sensor provides the most valuable information in fault classification and should be prioritized for further testing. Additionally, this evaluation investigated whether the placement of the sensors in the machine influenced their ability to detect relevant vibration patterns. Understanding how the physical positioning of each sensor impacts its data quality is crucial for optimizing sensor placement in future experiments or in a production environment. This evaluation also helped with identifying whether certain placements resulted in more qualitative and reliable data, or if they were more susceptible to noise and interference from other components in the machine, especially since there are many moving parts near the sensors. Furthermore, this initial evaluation is used as the foundation for determining the most optimal configuration of sensors, providing insights into whether a single, strategically placed sensor could be sufficient for accurate fault classification or if a combination of sensors would be required to achieve high accuracy. The results from this evaluation will help guide the sensor selection and placement strategy for further experiments.

3.8.3 Baseline performance on separate speeds & baselines

The 1D-CNN model and the BiLSTM model were first evaluated on separate speeds and separate baselines. The test was performed to get a clear understanding of how each factor influences the model's performance, to get an initial feeling of the model's performance, and to identify how each model behaves under different speeds and baselines. By evaluating the models at two different speeds and baselines, it can be identified how each model behaves separately, and initial performance benchmarks can be established. For example, baseline 1 has more noise introduced due to the unstable spacer component, and this extra noise can provide an early indication of which model is more robust and performs better under a higher level of noise. It can also be established if any model performs better at a certain speed.

3.8.4 Introducing Gaussian noise

The second step was to evaluate the robustness of the models with Gaussian noise, following the work of Sarkar et al [26] who proposed a method to evaluate model robustness by introducing unseen noise, such as Gaussian noise, with different levels of noise severity.

Gaussian noise is commonly used because it simulates real-world data issues, for example, sensor errors. Models that perform well under Gaussian noise can be considered more robust, as they can focus on underlying patterns rather than being distracted by random noise. Testing models on Gaussian noise is also a good approach to benchmark and compare different models' robustness against each other. In this study three different noise levels were introduced during testing; 2%, 5% and 10% noise. 2% is a low level of noise, 5% represents a moderate amount of noise and 10% represents a high level of noise in the data (see Appendix IV figures 22, 23 & 24). The models were also evaluated without any added noise and only on the noise that is inherent to the data set.

3.8.5 Cross-speed generalization

In this part of the training and evaluation phase, the models were trained on data from the 120 Pz/min speed and evaluated on unseen data from the 75 Pz/min speed. This test was performed to evaluate the two model's ability to generalize to different operational speeds. By training the model on 120 Pz/min data separately, a controlled environment was created to isolate the impact of speed variations on model performance. Testing on 75 Pz/min data challenges the model to adapt to an unseen speed condition, providing insights into its robustness and flexibility. A model that generalizes well across different speeds shows its ability to extract fundamental patterns and features that are invariant to the operational speed, for example, fault signatures or vibration patterns that persist despite changes in speed. This evaluation is beneficial for situations where machine operating speeds vary due to, for example, production demands or process requirements. The results help decide whether the model can classify faults effectively at speeds it has not been explicitly trained on, ensuring reliability in real-production scenarios.

3.8.6 Combined speeds

Based on the work of Ding et al [27], which emphasizes the importance of evaluating a model's ability to handle variability under fluctuating speed conditions, this study's final step of the testing and evaluation phase involved training and testing the two models on a data set with the combined speeds of 75 and 120 Pz/min speed. Combining data from multiple speed conditions during training can enable the models to become more robust and adaptable to different speeds and fluctuating speeds. This is beneficial in the real production environment, where machines very rarely operate at a fixed speed, but instead can vary based on operational demands.

Training and assessing the models on combined speeds demonstrate how well they can generalize across varying speed conditions, which simulates the real-production environment where the systems are exposed to fluctuating speeds. Testing on a combination of speeds is also important because fault patterns can display different characteristics across speeds due to changes in vibration, load, or even thermal variations. Including the two speeds in training enables the models to capture speed-dependent fault characteristics, which improves its ability to accurately classify faults regardless of the operating state and ensures consistent performance in varying environments.

3.9 Performance Evaluation Metrics

A confusion matrix was used to evaluate the classification and model performance, together with F1-score, precision, recall & accuracy following Vakili et al [28].

- 1) **Precision:** Answers the question “*Out of all predicted positives, how many are positive?*”. This means that out of all the observations the model has predicted as positive, how many of them were truly positive? Precision tells us how accurate the model’s positive predictions are. As seen in Equation 5, precision is calculated by dividing the number of true positives by the sum of true positives and false positives.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (5)$$

- 2) **Recall:** Answers the question “*Out of all actual positives, how many did the model correctly predict?*” Recall measures the percentage of actual positive cases that the model successfully identifies. As seen in Equation 3.6, recall is calculated by dividing the number of true positives by the sum of true positives and false negatives.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (6)$$

- 3) **F1-score:** Combines precision and recall into a single value. The F1-score provides a balanced measure of the model's performance, especially if there is an imbalance between precision and recall. The F1-score is extra useful when both false positives and false negatives carry significant cost. As seen in Equation 7, the F1 score is calculated as the harmonic mean of precision and recall, balancing the trade-off between precision and recall.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

- 4) **Accuracy:** Used to measure the overall performance of a model. It shows the percentage of correct predictions made by the model out of all predictions on the test set. Accuracy should be used carefully when dealing with unbalanced data sets and used together with precision and recall. As seen in Equation 8, accuracy is the proportion of correctly classified instances out of the total instances.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Number\ Of\ Instances} \quad (8)$$

- 5) **Confusion Matrix:** Provides a detailed breakdown of how the model's predictions compare to the actual labels, helping the user understand the model's errors.

Chapter 4

4. Results & Discussion

4.1 Accuracy for separate sensors with 1D-CNN

When evaluating the separate sensors, sensor 1 and 3 both achieved an accuracy of 97%, and is providing data that is highly informative, with patterns or features that the 1D-CNN easily can learn and use effectively. In contrast, sensor 2 with a lower accuracy of 83% highlights that it might have noisier data, or less distinct patterns which makes the classification task more difficult and making it harder for the model to achieve strong accuracy. The varying accuracy across the sensors suggests that there are some differences in the quality and relevance of the data provided by each sensor. This indicates that the placement of sensor 2 was not as optimal as the other sensor placements, highlighting the importance of how the location of the sensor can affect the results and data quality. The weaker performance of sensor 2 could potentially be attributed to environmental factors unique to its location, such as electromagnetic interference or vibrations caused by nearby moving components. Factors like these could introduce greater variability in its readings compared to the other sensors. Based on the results it was decided to continue with sensor 1 only, for further classification tasks with the 1D-CNN model.

4.2 Accuracy for different sensor combinations with 1D-CNN-BiLSTM

The results from the sensor combination comparison highlight that the second highest accuracy of 97% was achieved with sensor 1 alone, indicating that sensor 1 is highly effective and capable of single-handedly providing the information necessary to perform fault classification alone. This suggests that sensor 1 is informative. When sensor 1 was combined with sensor 3, the accuracy slightly increased to 98%, indicating that sensor 3 adds some value to the model, but the improvement is marginal. This suggests that sensor 3 provides complementary information that enhances the model's performance when used with sensor 1. However, it is also possible that sensor 3, on its own, could be a strong candidate for fault classification. When sensors 1, 2, and 3 were combined the performance dropped significantly to 84%. This indicates that the inclusion of sensor 2 may be detrimental to the model's performance. It is possible that sensor 2 introduces redundancy or noise to the data, which negatively impacts the model's ability to make accurate predictions.

Rather than enhancing the model, sensor 2 seems to damage its performance, suggesting that sensor 1 and sensor 3 are sufficient for achieving the best results, and that the inclusion of sensor 2 makes the prediction more difficult for the model. Due to these results, it was decided to proceed with the combination of sensor 1 and 3, and to not use the data from sensor 2.

4.3 Separate baselines & speeds

4.3.1 1D-CNN-BiLSTM

Looking at the results from baseline 2 at the 120 Pz/min speed, the BiLSTM model displays strong performance, with perfect precision, recall, and F1-scores of 100% across every fault class as seen in Table 3. The overall accuracy of 100% demonstrates that the model is highly effective at this baseline and speed, with no apparent issues in classification. This is also presented in Figure 9, which highlights only a few misclassifications in total. With baseline 2 at the 75 Pz/min speed, the model's performance decreases, with the overall accuracy dropping to 95% as seen in Table 4. While some fault classes still maintain perfect recall, precision and F1-scores, other fault types like the loose belt indicate challenges with adapting to the lower speed. This can also be seen in Figure 10, where it is highlighted that the loose belt often is being confused with the baseline.

As seen in Table 5, for baseline 1 at the 120 Pz/min speed, the model continues to show strong performance, with an accuracy of 98%. While the results for baseline 1 are slightly lower than those seen at baseline 2, the model still achieves high precision, recall, and F1-scores. Baseline 1 has both a higher noise level and more faults to classify, indicating that the model is robust. This is also displayed in Figure 7, where there are very few misclassifications. At baseline 1 with the 75 Pz/min speed, the model's performance drops again, with accuracy decreasing to 96% as seen in Table 6. There is noticeable decrease in recall and F1-scores for certain classes, for example, the baseline and the loose belt fault, which highlights the difficulties with generalizing to lower speeds and different baseline conditions. Figure 8 highlights that the small gear fault and is getting confused with the ball bearing 1 fault. Figure 8 also shows that the loose belt fault and the baseline is getting confused with the ball bearing 2 fault.

Classification Report CNN Bi-LSTM (120 Speed & baseline 2)			
Class	Precision	Recall	F1-Score
BL	1.00	1.00	1.00
LBTPIII	1.00	1.00	1.00
SGPI	1.00	1.00	1.00
SPACER	1.00	1.00	1.00
Accuracy	1.00		
Macro avg	1.00	1.00	1.00
Weighted avg	1.00	1.00	1.00

Table 3. Classification report - 120 Speed & baseline 2 - 1D CNN-BiLSTM

Classification Report CNN Bi-LSTM (75 Speed & Baseline 2)			
Class	Precision	Recall	F1-Score
BL	0.84	1.00	0.91
LBTPIII	1.00	0.88	0.93
SGPI	1.00	0.93	0.97
SPACER	1.00	1.00	1.00
Accuracy	0.95		
Macro avg	0.96	0.95	0.95
Weighted avg	0.96	0.95	0.95

Table 4. Classification report - 75 Speed & baseline 2 - 1D CNN-BiLSTM

Classification Report CNN Bi-LSTM (120 Speed & Baseline 1)			
Class	Precision	Recall	F1-Score
BBI	0.97	0.95	0.96
BBII	0.97	0.97	0.97
BL	0.99	1.00	0.99
LBTPIII	1.00	1.00	1.00
SGPI	0.98	1.00	0.99
TB	1.00	1.00	1.00
Accuracy	0.98		
Macro avg	0.98	0.98	0.98
Weighted avg	0.98	0.98	0.98

Table 5. Classification report for unseen data (75 Speed) - Hybrid 1D CNN

Classification Report CNN Bi-LSTM (75 Speed & Baseline 1)			
Class	Precision	Recall	F1-Score
BBI	0.91	1.00	0.95
BBII	0.89	1.00	0.94
BL	1.00	0.92	0.96
LBTPIII	1.00	0.94	0.97
SGPI	1.00	0.91	0.95
Accuracy	0.96		
Macro avg	0.96	0.96	0.96
Weighted avg	0.96	0.96	0.96

Table 6. Classification report for unseen data (75 Speed) - Hybrid CNN-BiLSTM

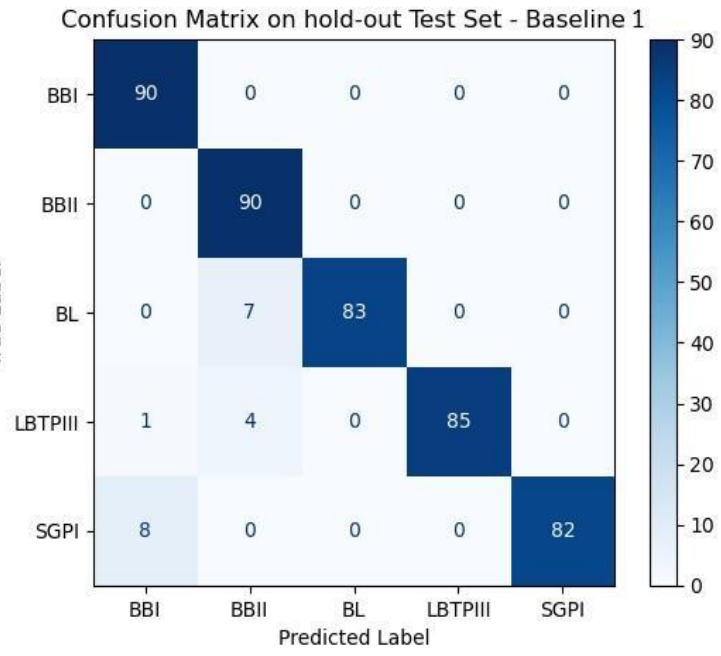
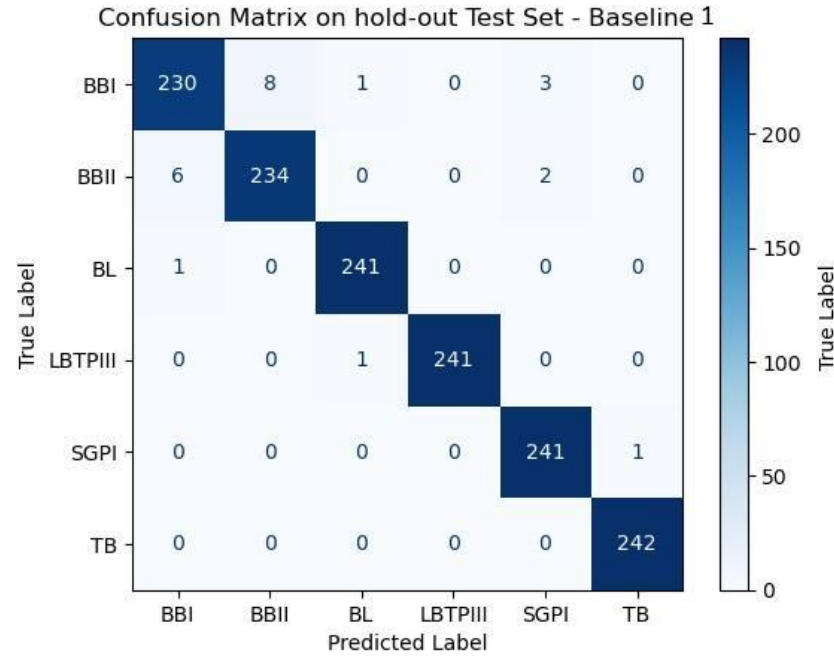


Figure 7. Confusion Matrix. 1D CNN-BiLSTM - Baseline 1 (120 Pz/min) & Figure 8. Confusion Matrix. 1D CNN-BiLSTM - Baseline 1 (75 Pz/min)

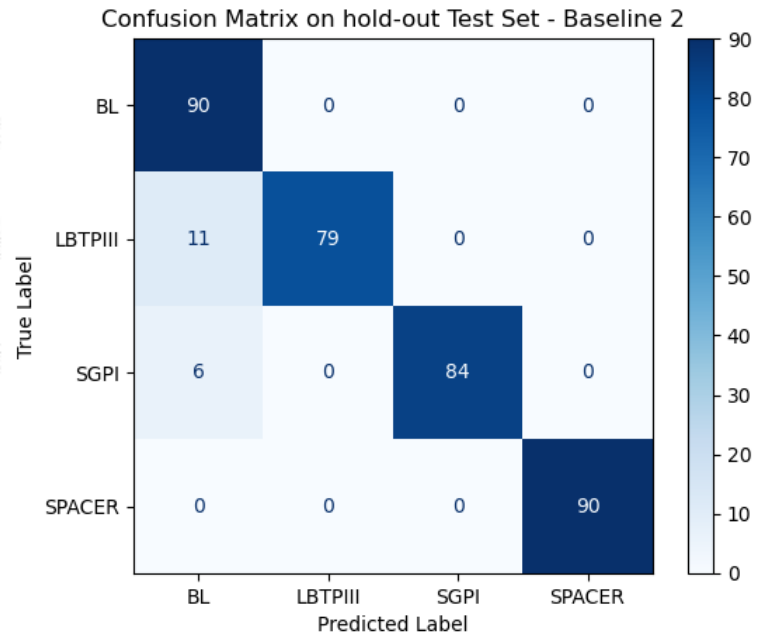
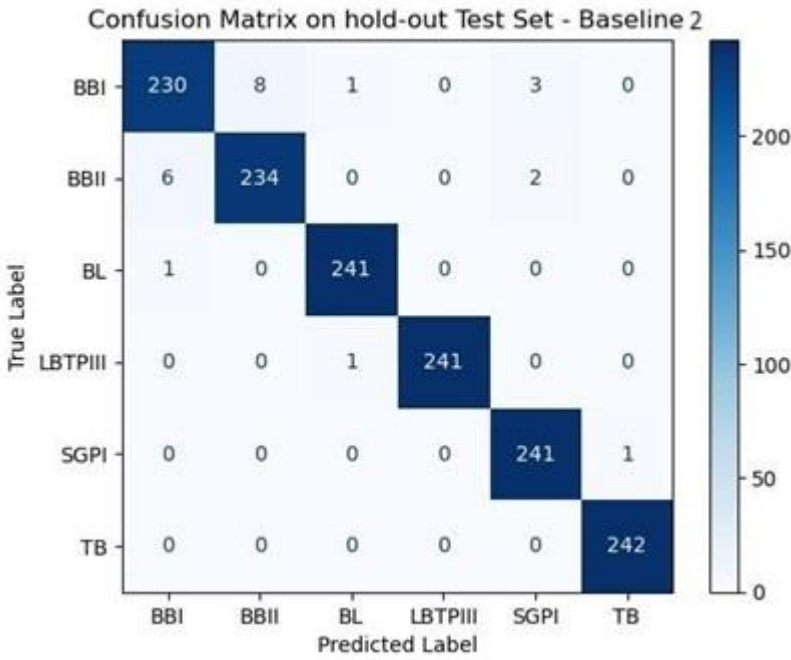


Figure 9. Confusion Matrix - 1D CNN-BiLSTM. Baseline 2 (120 Pz/min) & Figure 10. Confusion Matrix - 1D CNN-BiLSTM. Baseline 2 (75 Pz/min)

4.3.2 Summary

The model's performance varies depending on the baseline and the speed conditions. At baseline 2 with the 120 Pz/min speed, the model performs exceptionally well. At baseline 2 and baseline 1 with the 75 Pz/min speed, the accuracy drops to 95% and 96%, respectively.

This indicates that the model has an easier time classifying faults at a higher speed, and that the nature of these faults is more prominent at higher speeds. At baseline 1 with the 120 Pz/min speed, the model still shows strong performance with a 98% accuracy, despite the added challenge of having more faults to classify, and more noise caused by the unstable spacer component. This demonstrates the model's robustness under more complex conditions.

4.4 1D-CNN

The 1D-CNN model shows strong performance at the 120 speed with baseline 2, achieving an accuracy of 91% as seen in Table 7. The spacer fault has a perfect precision, recall, and F1-score of 100%, indicating strong performance. In contrast, the baseline has a lower precision of 78%, though the recall remains high at 98%. This suggests that the model struggles to achieve high precision but can detect most instances of the baseline. The loose belt fault has a high precision of 99%, but its recall drops to 75%, indicating that while the model is good at avoiding false positives, it misses a significant number of the total instances. As seen in Figure 13, the performance varies across the different faults at the 120 speed and baseline 2.

As seen in Table 8, at the 75 Pz/min speed with baseline 2, the 1D-CNN model's accuracy drops to 89%. The baseline shows a slight improvement in precision at 79%, but the recall decreases to 94%. The loose belt fault, on the other hand, shows an improvement in recall at 98%, but the precision drops to 89%, indicating that the model detects more instances of the loose belt fault, but with a higher number of false positives. The small gear fault experiences a notable decrease in recall at 72%, while the precision remains high at 93%, showing a drop in its detection ability at the slower speed. As seen in Figure 14, the most misclassified fault is the small gear fault, which is getting confused with the baseline. As seen in Table 9, at baseline 1 with the 120 Pz/min speed the model has an overall accuracy of 86%. The baseline highlights a precision of 100% and a recall at 95%, which is better than baseline 2's result at the same speed. Overall, the model shows decent performance in most faults, but the recall for some faults, like the small gear, is notably lower compared to baseline 2, at the same speed.

Figure 11 highlights that it is the two ball bearing faults that have the biggest difficulty with being confused and misclassified for each other. At the 75 Pz/min speed with baseline 1, the 1D-CNN model's performance is displaying a more significant decline, achieving an accuracy of 86% as seen in Table 10. The ball bearing 1 fault experiences a decrease in both precision at 86% and recall at 93%, compared to the 120 speed. The ball bearing 2 fault shows a notable drop in precision at 68%, although recall remains perfect at 100%, indicating a high rate of false positives. The baseline continues to show a decline in precision at 78% and a perfect recall of 100%. The loose belt fault highlights a significant decrease in recall at 72%, though its precision remains high at 97%. Similarly, the small gear fault shows a decrease in recall at 89%, while precision remains at 98%. Overall, these changes contribute to the model's reduced accuracy at the 75 Pz/min speed with baseline 1. As seen in Figure 12 it is the baseline and loose belt fault that is getting the most misclassified and confused with the ball bearing 2 fault.

Classification Report 1D-CNN (120 Pz/min & baseline 2)			
Class	Precision	Recall	F1-Score
BL	0.78	0.98	0.87
LBTPIII	0.99	0.75	0.85
SGPI	0.92	0.92	0.92
SPACER	1.00	1.00	1.00
Accuracy	0.91		
Macro avg	0.92	0.91	0.91
Weighted avg	0.92	0.91	0.91

Table 7. Classification report - 120 Pz/min & baseline 2 - 1D CNN

Classification Report 1D-CNN (75 Pz/min & Baseline 2)			
Class	Precision	Recall	F1-Score
BL	0.79	0.94	0.86
LBTPIII	0.89	0.98	0.93
SGPI	0.93	0.72	0.81
SPACER	1.00	0.93	0.97
Accuracy	0.89		
Macro avg	0.90	0.89	0.89
Weighted avg	0.90	0.89	0.89

Table 8. Classification report for unseen data (75 Pz/min) - 1D CNN

Classification Report 1D-CNN (120 Pz/min & Baseline 1)			
Class	Precision	Recall	F1-Score
BBI	0.68	0.80	0.74
BBII	0.81	0.61	0.69
BL	1.00	0.95	0.97
LBTPIII	0.91	0.94	0.92
SGPI	0.86	0.90	0.88
TB	0.94	0.98	0.96
Accuracy	0.86		
Macro avg	0.87	0.86	0.86
Weighted avg	0.87	0.86	0.86

Table 9. Classification report - 120 Pz/min & baseline 1 - 1D CNN

Classification Report 1D-CNN (75 Pz/min & Baseline 1)			
Class	Precision	Real	F1-Score
BBI	0.86	0.93	0.89
BBII	0.68	1.00	0.81
BL	1.00	0.78	0.88
LBTPIII	0.97	0.72	0.83
SGPI	0.98	0.89	0.93
Acura	0.86		
Marco avg	0.90	0.86	0.87
Weighted avg	0.90	0.86	0.87

Table 10. Classification report - 75 Pz/min & baseline 1 - 1D CNN

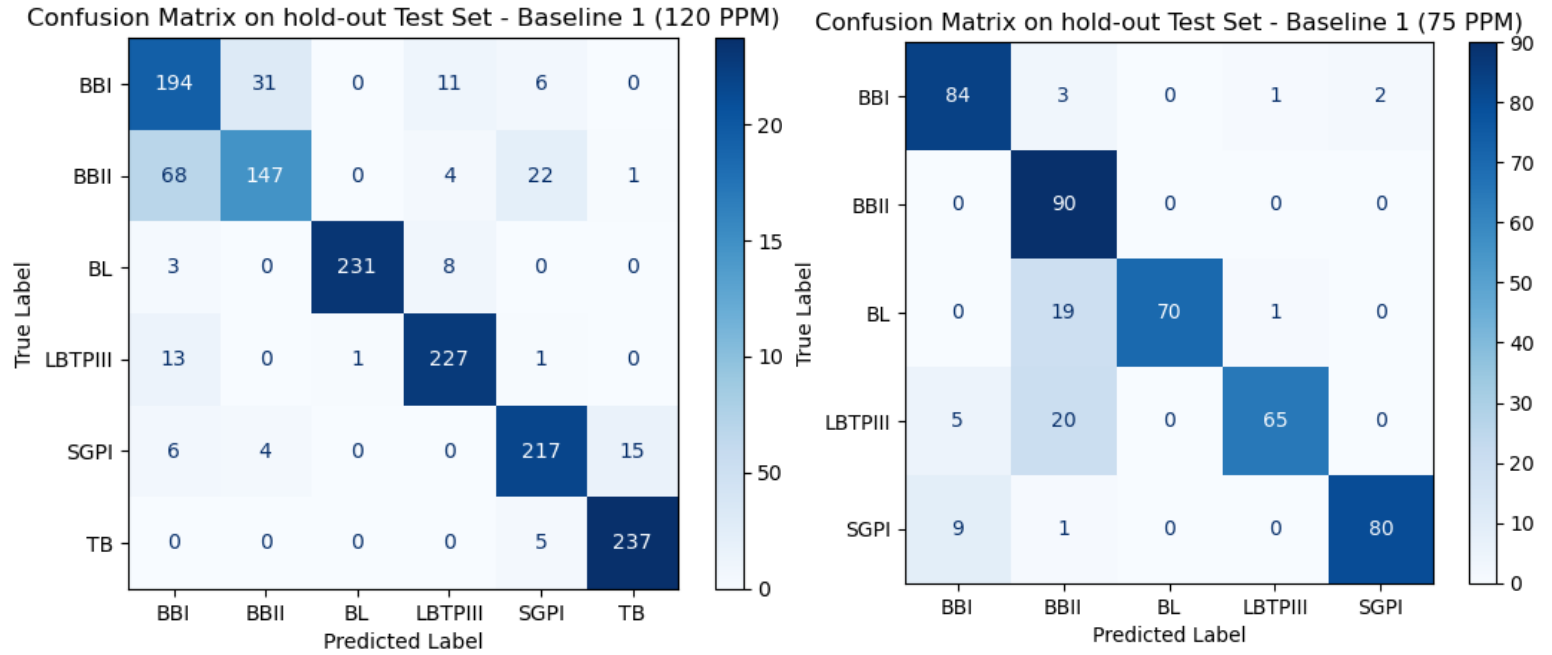


Figure 11. Confusion matrix. 1D CNN - Baseline 1 (120 Pz/min) & Figure 12. Confusion matrix. 1D CNN - Baseline 1 (75 Pz/min)

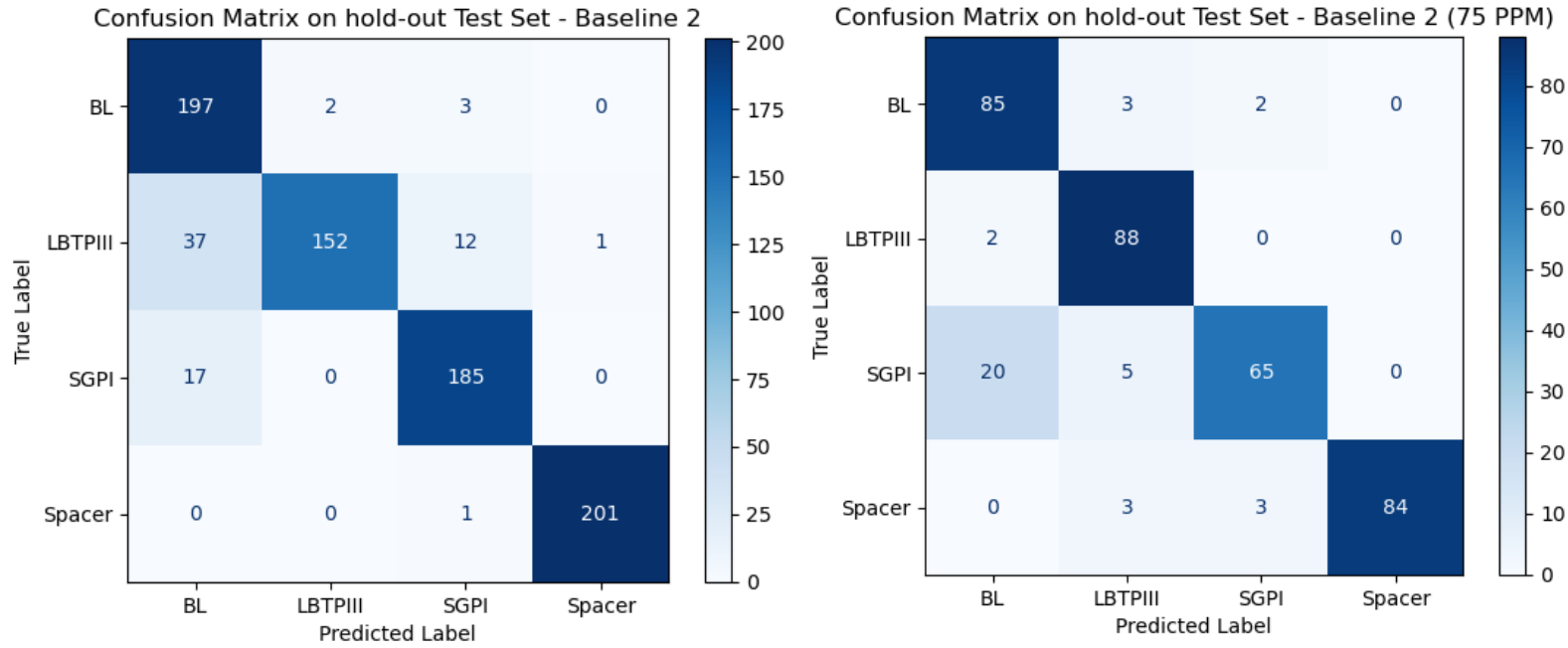


Figure 13. Confusion Matrix - 1D CNN. Baseline 2 (120 Pz/min) & Figure 14. Confusion Matrix - 1D CNN. Baseline 2 (75 Pz/min)

4.4.1 Summary

The 1D-CNN model performs best with baseline 2 at the 120 Pz/min speed, achieving the highest accuracy of 91%. Its performance deteriorates at the 75 speed, with an accuracy decreasing to 89% for baseline 2, and 86% for baseline 1. This indicates that it is easier for the model to classify faults at a higher speed, most probably due to the nature of the fault itself and on a less noisy baseline.

4.5 Robustness test with Gaussian noise

Figure 15 displays how the Gaussian noise impacts the robustness of the 1D-CNN and the BiLSTM model differently. At 0% noise, both models are highlighting a strong performance, with the 1D-CNN achieving 97% accuracy, and the BiLSTM model slightly outperforming it at 100%. This indicates that both models are highly capable of extracting relevant features from the raw vibration data, though the BiLSTM model demonstrates a slightly better performance. As noise increases to 2%, the accuracy of the 1D-CNN drops significantly to 78%, while the BiLSTM model maintains a high accuracy of 99%. This suggests that the BiLSTM model is more resilient to moderate noise levels.

At 5% noise, the performance of both models decreases significantly, with the 1D-CNN model achieving a 25% accuracy and the BiLSTM model performing slightly better at 28%. At this noise-level both models struggle, and the noise is severely disrupting the data's underlying patterns. At 10% noise, the 1D-CNN model achieves 24% accuracy, while the BiLSTM model performs similarly at 23%. Both models perform close to a random guessing at the 10% noise level.

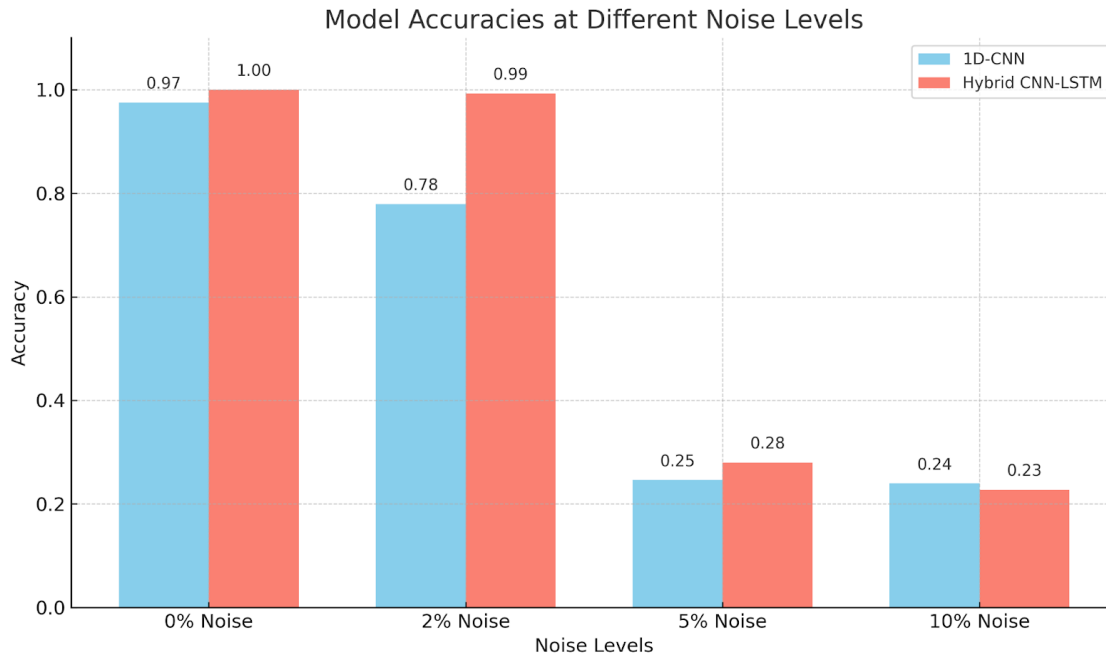


Figure 15. Grouped barplot of accuracy comparison over noise levels. 1D CNN vs 1D CNN-BiLSTM

4.6 Cross-Speed Generalization

4.6.1 1D-CNN-Bi LSTM

As seen in Table 12, the model achieves a moderate accuracy of 71% with baseline 2. For baseline 1 (see Table 14) which has slightly more noise in it, the model achieves a lower accuracy of 68%. When it comes to the different fault types at baseline 2, the spacer fault is detected and classified correctly at both baselines, showing a high precision of 94% and recall of 97% together with a F1-Score of 96% for baseline. One fault type that the model struggles to classify at baseline 2, but not at baseline 1 is the small gear fault that was induced close to sensor 1. Looking at Table 12, it is highlighted that the small gear fault has a precision of 83%, recall of 30% and a F1-score of 44%. As shown in Table 14, the small gear fault at baseline 1 has a precision of 93%, a recall of 97% and a F1-score of 95%.

The baseline has a strong recall for both baseline 1 and baseline 2. However, precision is lower, indicating some misclassifications. As shown in Figure 17, the small gear fault is often getting mistaken for the baseline. When looking at the ball bearing fault 1 at baseline (Figure 16) it is highlighted that it has a relatively strong performance. It is also displayed that the model struggles with the ball bearing 2 fault, with a precision, recall, and F1-score of 0%, which suggests that the fault is difficult for the model to classify from one speed to another. Figure 17 showcases that the loose belt fault is being mistaken for the baseline and that the ball bearing 2 fault is the most misclassified fault, which is getting confused with the baseline and the loose belt fault.

4.6.2 1D-CNN

As seen in Tables 11 & 13, the 1D-CNN model achieved a low accuracy of 66% at baseline 1, and 63% with baseline 2, which is both lower than the BiLSTM model's performances. Figures 18 and 19 highlight a similar pattern to the BiLSTM model where the spacer fault is highlighting a strong precision at 98%, strong recall at 93% and an F1-score of 95% for baseline 2 (see Table 11). However, the model struggles to classify the small gear fault on baseline 2, a challenge that is not observed with baseline 1, highlighting a significant difference in performance between the two baselines. The metrics for the small gear fault at baseline 1 are 88% precision, 100% recall and an F1-score of 93%. At baseline 2 the small gear fault has a performance of 91% precision, 6% recall and an F1-score of 12%. As seen in Figure 19, the most misclassified class is the small gear fault and the loose belt fault, which is getting confused with the baseline. As seen in Figure 18, the most confused fault is the ball bearing 2 fault that is getting confused with the baseline and the loose belt fault.

Classification Report - Cross-Speed-Generalization - 1D CNN (75 Pz/min & baseline 2)			
Class	Precision	Recall	F1-Score
BL	0.43	0.93	0.59
LBTPIII	0.74	0.60	0.66
SGPI	0.91	0.06	0.12
SPACER	0.98	0.93	0.95
Accuracy	0.63		
Macro avg	0.76	0.63	0.58
Weighted avg	0.76	0.63	0.58

Table 11. Classification report - Cross-Speed Generalization- 1D CNN

Classification Report - Cross-Speed-Generalization - Bi-LSTM (75 Pz/min & Baseline 2)			
Class	Precision	Recall	F1-Score
BL	0.51	0.92	0.66
LBTPIII	0.79	0.65	0.71
SGPI	0.83	0.30	0.44
SPACER	0.94	0.97	0.96
Accuracy	0.71		
Macro avg	0.77	0.71	0.69
Weighted avg	0.77	0.71	0.69

Table 12. Classification report - Cross-Speed Generalization - 1D CNN-BiLSTM

Classification Report - Cross-Speed-Generalization - 1D CNN (75 Pz/min & Baseline 1)			
Class	Precision	Recall	F1-Score
BBI	0.94	0.69	0.80
BBII	0.00	0.00	0.00
BL	0.59	0.85	0.70
LBTPIII	0.46	0.75	0.57
SGPI	0.88	1.00	0.93
Accuracy	0.66		
Macro avg	0.57	0.66	0.60
Weighted avg	0.57	0.66	0.60

Table 13. Classification report - Cross-Speed Generalization- 1D CNN

Classification Report - Cross-Speed-Generalization - Bi-LSTM (75 Pz/min & Baseline 1)			
Class	Precision	Recall	F1-Score
BBI	0.95	0.72	0.82
BBII	0.00	0.00	0.00
BL	0.51	0.92	0.65
LBTPIII	0.59	0.81	0.69
SGPI	0.93	0.97	0.95
Accuracy	0.68		
Macro avg	0.60	0.68	0.52
Weighted avg	0.60	0.68	0.62

Table 14. Classification report - Cross-Speed Generalization - 1D CNN-BiLSTM

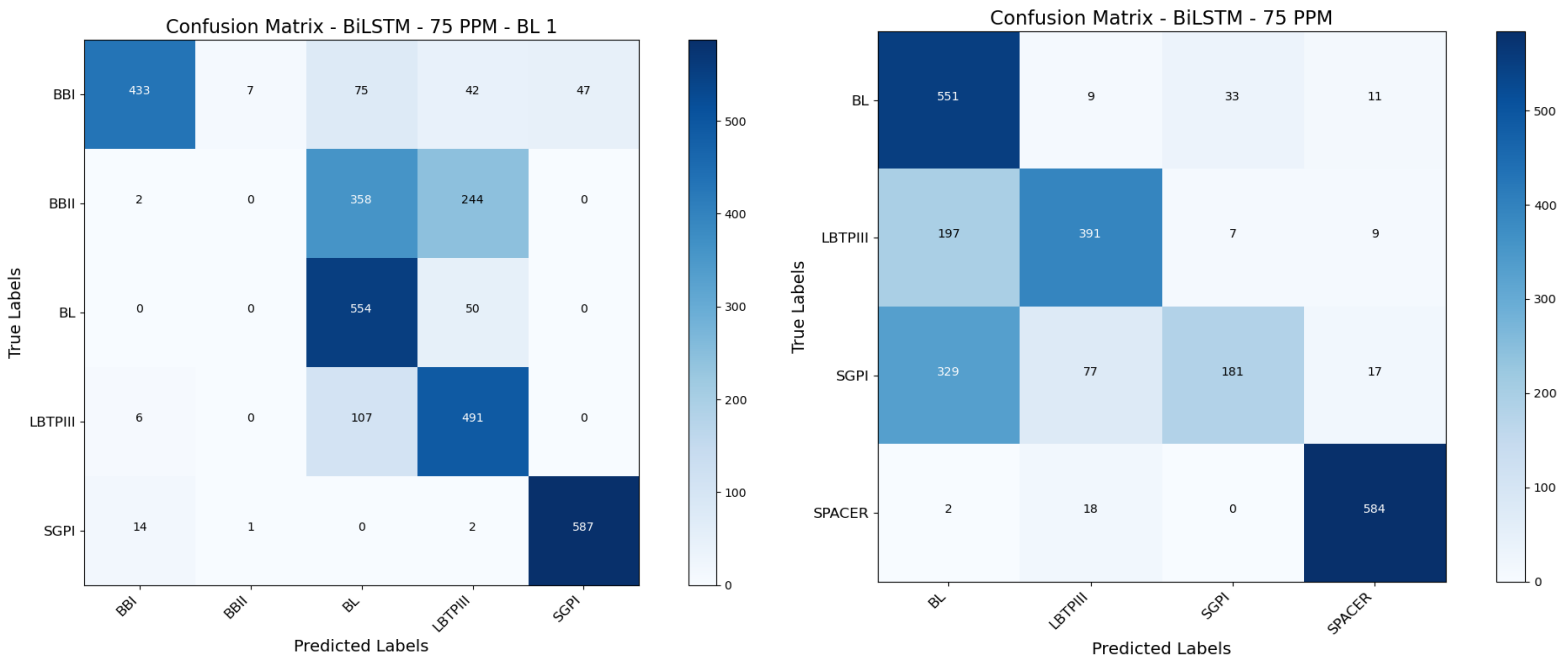


Figure 16. Confusion Matrix - 1D CNN-BiLSTM. Cross-Speed Generalization (Baseline 1) & Figure 17. Confusion Matrix - 1D CNN-BiLSTM. Cross-Speed Generalization (Baseline 2)

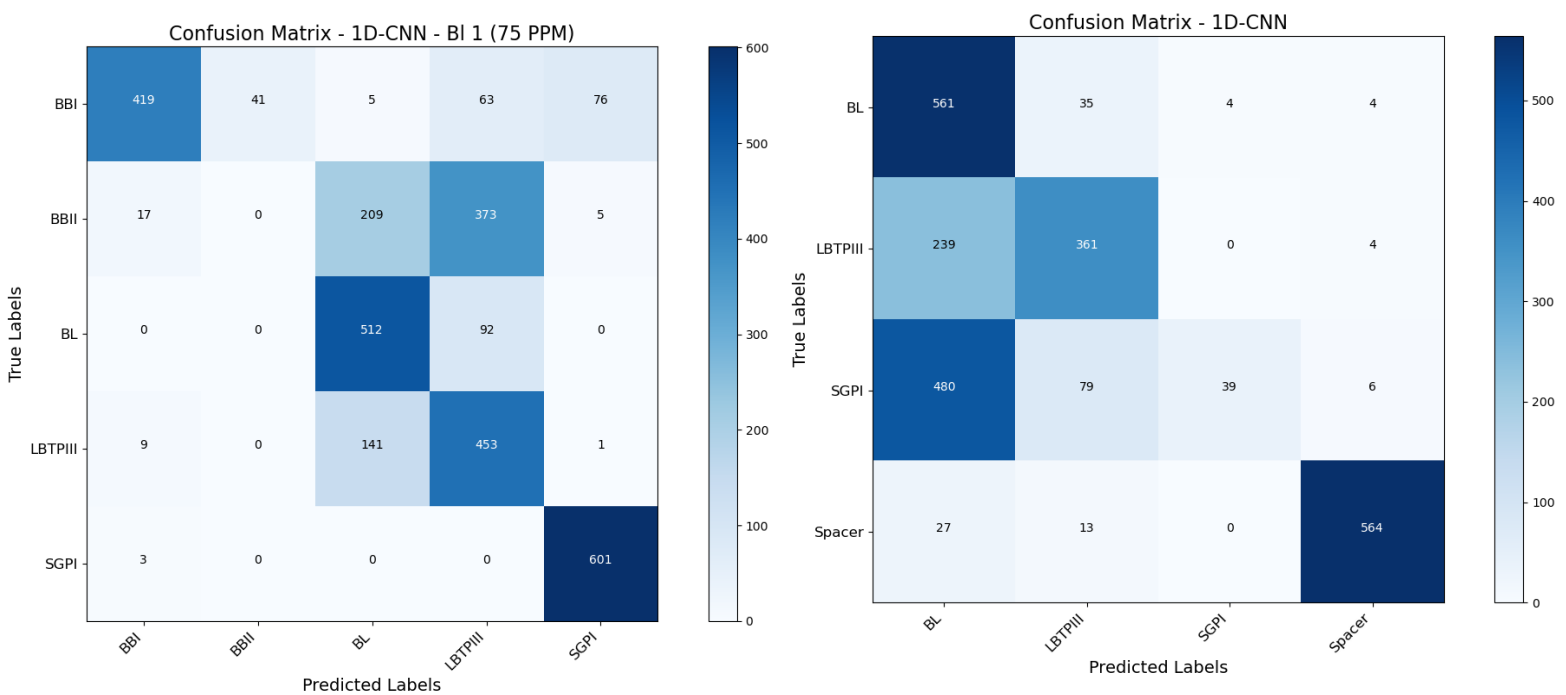


Figure 18. Confusion Matrix - 1D CNN for Cross-Speed Generalization (Baseline 1) & Figure 19. Cross-Speed Generalization (Baseline 2)

4.6.3 Summary

The BiLSTM model outperforms the 1D-CNN in cross-speed generalization. It achieves accuracies of 71% on baseline 1 and, 66% on baseline 2, compared to the 1D-CNN's 63%, and 66%, respectively. The BiLSTM model consistently demonstrates better fault detection and classification metrics compared to the 1D-CNN and it also shows a stronger performance on the noisier baseline 1. These results display the strength of the bidirectional layer which allows the model to process the input data both forwards and backwards in time, enabling it to capture long-term information. This has shown to be particularly useful for cross-speed generalization, where changes in patterns due to varying speeds can affect how faults manifest in the data. The BiLSTM model's recurrent structure is allowing it to smooth out inconsistencies over time and detect patterns that persist across sequence, which is effective in scenarios like cross-speed generalization where noise and variability is introduced. In contrast, the 1D-CNN model is primarily focusing on extracting local features and lacks the ability to consider sequential dependencies as effectively as the BiLSTM model. This could be the reason for the weaker generalization across the different speed conditions, where the temporal development of the patterns is critical. The ability of some faults to generalize across speeds while others struggle could potentially depend on how distinct and speed-invariant the fault signals are at different speeds.

Faults that generate clear, repetitive patterns, such as the spacer fault, tend to be easier for the model to classify at both the 120 and 75 speed. However, faults like the small gear fault could potentially display more complex or speed-dependent behaviors, which is making it harder for the model to recognize and classify when evaluated at a lower speed. The model could also potentially have overfitted to the features specific to the 120 speed, leading to a weak generalization ability at the 75 speed, for certain fault types. Finally, it can be stated that speed generalization remains a significant challenge, as none of the models demonstrate optimal performance across the two speeds, highlighting their limitations in adapting to changing operating conditions.

4.7 Combined Speeds

4.7.1 1D-CNN

The 1D-CNN model is highlighting weak performance on unseen data with combined speeds and has an accuracy of 60% as seen in Table 15. The precision, recall and F1-scores of the model indicates that there is variability between the faults. The fault that the models fail to classify the most is the ball bearing 1 fault, with an F1-score of 35%, and the loose belt fault with an F1-score of 54%. The small gear fault demonstrates a high recall of 100% and a lower precision of 49%, leading to an F1-score of 66%. The model classifies the baseline the most accurately, highlighting an F1-score of 85%. Overall, the macro and weighted averages of precision, recall, and F1-score display an accuracy of 60%, which reflects a lack of robustness across all faults. Figure 21 further displays that the ball bearing 2 fault is getting confused with the ball bearing 1 fault and the small gear fault.

4.7.2 Summary

The 1D-CNN model's performance decreases on combined speeds compared to the cross-speed-generalization performance. This could be since when the model is trained on combined speeds, it is exposed to a wider range of data variability, making it harder for the model to capture consistent patterns. The increased complexity in the data could potentially be the reason for the reduced performance because the model must generalize across different speed conditions. Another reason for the decreased performance could be the model's difficulty in capturing speed-specific features, when each speed has unique fault characteristics or patterns, and training on combined speeds forces the model to learn features that work across all speeds. If these fault patterns do not generalize well between speeds, the model might struggle to capture the relevant features. Additionally, when training on a single speed, the temporal variations of the fault signals are more uniform, but combining speeds could introduce shifts in the timing of fault signatures.

4.7.3 1D-CNN-BiLSTM

The BiLSTM model is highlighting a robust performance with an accuracy of 96% on the combined speeds. As seen in Table 16, the precision, recall and F1-scores are consistently high across all faults with the lowest F1-score being 91% for the ball bearing 2 fault.

The faults that the model can classify the best are the baseline, loose belt fault and the small gear fault, which all achieve almost perfect metrics. The model displays a strong ability to manage more complex patterns and variability in the data. The macro and weighted averages for this model are 96% over all classes, which is a significant improvement in overall performance and reliability compared to the 1D-CNN model. Figure 20 highlights the robust performance of the model with very few misclassifications.

4.7.4 Summary

The results indicate that incorporating a BiLSTM layer into the 1D-CNN architecture, especially when processing data with varying speeds, results in a more robust model that outperforms the regular 1D-CNN significantly. The inclusion of the BiLSTM layer is allowing the model to capture sequential dependencies and temporal patterns.

Classification Report - Combined Speeds - 1D CNN (Baseline 1)			
Class	Precision	Recall	F1-Score
BBI	0.35	0.35	0.35
BBII	0.69	0.51	0.59
BL	0.97	0.76	0.85
LBTPIII	0.90	0.38	0.54
SGPI	0.49	1.00	0.66
Accuracy	0.60		
Macro avg	0.68	0.60	0.60
Weighted avg	0.68	0.60	0.60

Table 15. Classification report - Combined Speeds - 1D CNN

Classification Report - Combined Speeds - 1D CNN-BiLSTM (Baseline 1)			
Class	Precision	Recall	F1-Score
BBI	0.98	0.90	0.94
BBII	0.86	0.98	0.91
BL	0.98	0.96	0.97
LBTPIII	0.99	0.94	0.96
SGPI	0.99	1.00	0.99
Accuracy	0.96		
Macro avg	0.96	0.96	0.96
Weighted avg	0.96	0.96	0.96

Table 16. Classification report - Combined speeds - 1D CNN-BiLSTM

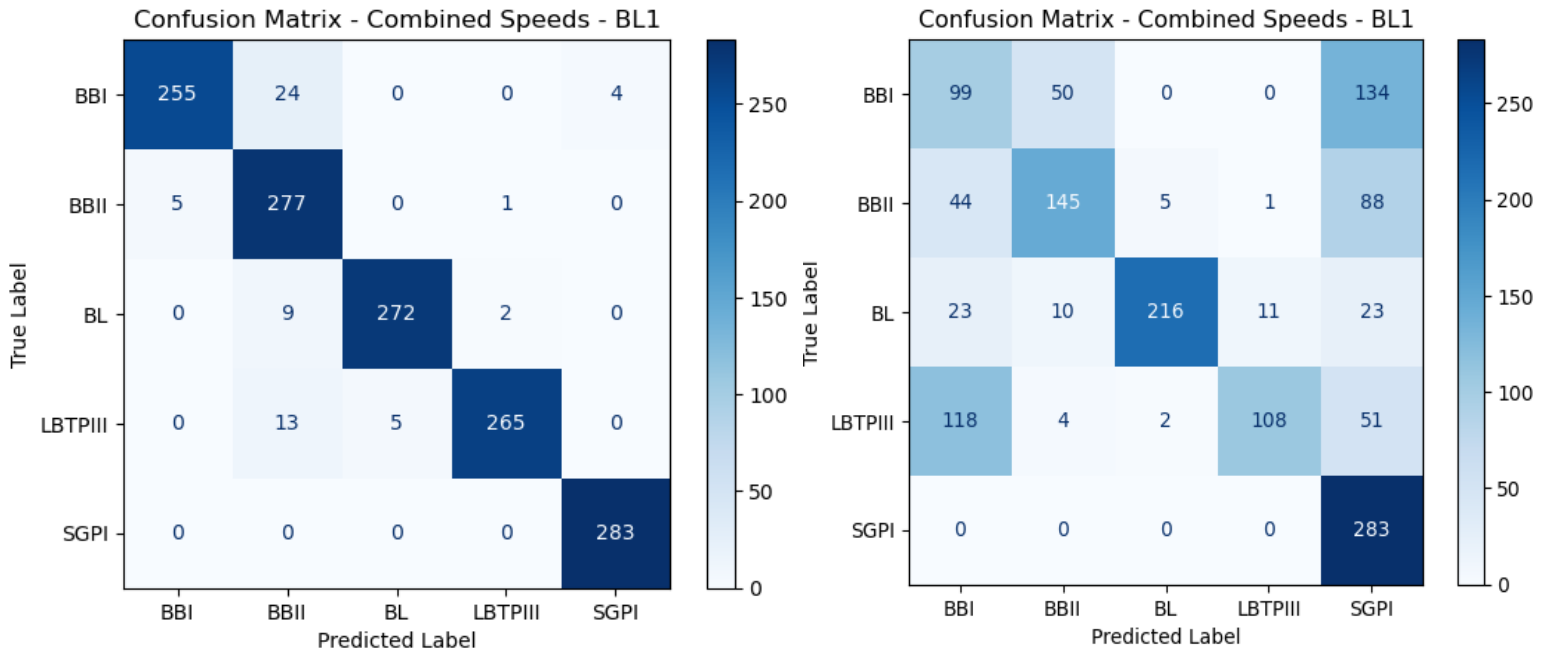


Figure 20. Confusion Matrix. 1D CNN-BiLSTM. Combined Speeds & baseline 1 & Figure 21. Confusion Matrix. 1D CNN. Combined Speeds & baseline 1.

Chapter 5

5. Conclusions & Future Research

This study demonstrates promising results for fault classification in small and complex packaging machines. The three sensors were positioned inside the machine, close to numerous moving parts, yet the approach proved to be highly effective. Furthermore, the findings suggest that the vibration measurements currently used by AstraZeneca are sufficient for this task. However, it is important to note that this study was conducted in an isolated environment and not within a full production setting. Therefore, it cannot be stated that the method used for fault classification will produce similar results in a production environment where upstream and downstream equipment are present. Additionally, the study did not account for disturbances typically found in real production settings, such as robots and operators performing tasks like refilling material. Despite these limitations, the study revealed that the machine faults could be classified using vibration sensors with a measurement range of 10-1000 Hz, which is an encouraging outcome.

When it comes to cross-speed generalization it has proven challenging to achieve good accuracy and performance with both the 1D-CNN model and the BiLSTM model. However, the BiLSTM model achieves strong performance on combined speeds, and is showing promising results in this area, something that the 1D-CNN model failed to accomplish. It has been displayed that the BiLSTM model, which has a long-term temporal understanding, is the superior model in all categories explored, and has consistently shown the strongest performance and highest accuracy throughout this study. This suggests that the inclusion of the BiLSTM layer can effectively capture temporal patterns that are crucial for accurate predictions, which the 1D-CNN model fails to do. The stronger performance of the BiLSTM model indicates that the bidirectional LSTM layer provides the model with a better ability to identify subtle variations and trends in the input data. The model also showed strong performance, even if small time-delays were introduced due to the manual activation of the vibration sensors.

This suggests that the segment index and the BiLSTM layer together enable the model to maintain high accuracy despite challenges like minor time lags, highlighting the model's capacity for effective temporal reasoning. In this study, incorporating multiple sensors provided a minor increase in performance. However, it is important to note that sensor placement plays a crucial role in model performance, as evidenced by the significantly weaker results and performance observed from sensor 2. It should also be mentioned that even though the models had different kernel sizes which potentially have contributed to the different performances, the BiLSTM models temporal capabilities most likely played a more significant role in improving accuracy, since the kernel sizes are contributing to the feature extraction, but the BiLSTM's temporal memory is what really differentiates the two models, as well as the inclusion of multiple sensors. For future research, when it comes to advancing fault classification in packaging machines and using these techniques in a production setting, several things should be considered. First, every machine is like its own individual and they often have unique baselines, meaning that models trained on one baseline might not generalize to others without retraining.

In future research it is proposed that cross-baseline-generalization is explored. Models should also account for variations in machine speeds and noise by training on a diverse set of data, and applying noise-robust techniques, hence it is suggested to explore even more noise-robust models that can manage variation in speeds better. It should also be noted that this study did not have data on precursor signals leading up to faults, but only on baselines with directly induced faults. For future experiments, collecting data on precursor signals would allow the model to learn patterns and trends that occur before a fault is introduced, enabling the model to predict failures before they happen, which offers a more proactive approach to maintenance. Being able to forecast a machine breakdown could potentially be the next step for AstraZeneca and exploring the prediction of the machine's RUL is also suggested as a future approach. Lastly, introducing high frequency faults and using sensors with a broader frequency range could also be beneficial for enhanced fault detection in future research. For future research it is also proposed that the isolated effect of multiple sensors and the BiLSTM layer is explored separately, as this research only compares overall performance of the two models.

Appendix I - Literature Summary

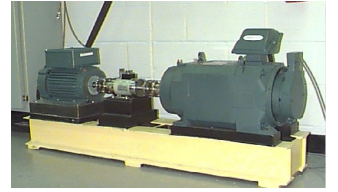
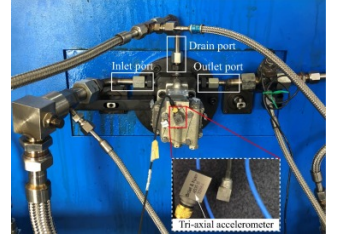
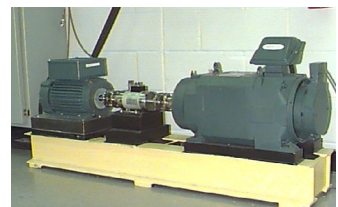
Authors	Model	Hyper-parameters	Result	Data	Multisensor	Sensor placement
Choudakkanavar & Mangai (2022)	1D-CNN-BiLSTM	<ul style="list-style-type: none"> - Kernel Size: 3x1 - Stride: 1 - Pooling: Max Pooling, 2x2 (Stride 2) - Fully connected layer: 256 units - Softmax activation - Loss Function: Cross-entropy - Weight update: Gradient descent 	99.8 %	Raw Vibration data No feature extraction Multi-channel (CWRU data set)	Yes	Rolling Bearing Test Platform 
Chao et al. (2019)	1D-CNN	<ul style="list-style-type: none"> - Kernel Size: 3 - Stride: 1 - Pooling: Max Pooling, 2x2 - Fully connected layer: 128 units - Softmax activation in Output -ReLU activation in hidden layers 	98.6 %	Raw vibration data Feature extraction Single-channel (Own data collection)	No	Accelerometer mounted on pump housing 
Chen et al (2024)	1D-CNN	<ul style="list-style-type: none"> - Kernel Size: 16x1 (Conv1) - Tanh activation - Max-pooling: 2x2 - Dropout: 0.3, 0.25 	98.9 %	Raw vibration data No Feature extraction Multi-channel (CWRU dataset)	Yes	Rolling Bearing Test Platform 

Table 17. Summary of reviewed literature's proposed models & results

Appendix II - Experimental log

	Date of Experiment	Start Time	End Time	Name	Type	#	Speed (Pz/min)	Comment
	10/14/2024	10:53	11:35	Trials	Trial		75	Trials with baseline after changing components
	10/14/2024	12:13	12:30	BL 1	Baseline		75	Baseline with new components
	10/14/2024	12:40	13:30	BL 1	Baseline		120	Baseline with new components
	10/14/2024	14:52	15:11	SGP1	Old component		120	Old component, no extra damage
	10/14/2024	15:11	15:20	SGP1	Old component		75	Old component, no extra damage
	10/14/2024	15:47	15:56	BB1	Damage		75	Impact on balls that is creating "clicking" when rotating
	10/14/2024	15:57	16:16	BB1	Damage		120	Impact on balls that is creating "clicking" when rotating
	10/14/2024	16:24	16:43	BB2	Damage		120	Graveled bearing with less resistance in rolling & bumps
	10/15/2024	08:46	09:06	BB2	Damage		75	Graveled bearing with less resistance in rolling & bumps
	10/15/2024	10:45	11:08	BTP3	Looseness		120	Looseness introduced to belt
	10/15/2024	11:35	11:44	BTP3	Looseness		75	Looseness introduced to belt
	10/18/2024	14:00	14:29	HBTP1	Tight belt		120	Overtightened with screwdriver
	10/22/2024	14:45	15:01	BL 2	Baseline		120	New baseline: Exchange of spacer from back to front
	10/22/2024	15:02	15:21	BL 2	Baseline		75	New baseline: Exchange of spacer from back to front
	11/6/2024	10:05	10:20	SGP1	Damage		75	Both bearings have been exposed to external force
	11/6/2024	10:22	10:46	SGP1	Damage		120	Both bearings have been exposed to external force
	11/6/2024	10:50	11:10	BTP3	Looseness		120	Damaged bearings still left in system, sensor 1
	11/6/2024	11:10	11:24	BTP3	Looseness		75	Damaged bearings still left in system, sensor 1

Table 18. Experimental Log

Appendix III - Model Configuration

Model Configuration	Hyper Parameters
Window parameters	Win_len = 400 & Stride = 100
Convolutional Layers (1D-CNN)	Conv1D Layer 1: 128 filters, kernel size = 10 Conv1D Layer 2: 64 filters, kernel size = 5 Conv1D Layer 3: 32 filters, kernel size = 3
Convolutional Layers (1D CNN-BiLSTM)	Conv1D Layer 1: 128 filters, kernel size = 15 Conv1D Layer 2: 64 filters, kernel size = 10 Conv1D Layer 3: 32 filters, kernel size = 8
Pooling Layers	MaxPooling1D with pool sizes of 4 and 2
Batch Normalization	Applied after each Conv1D layer
Dropout Layers	Dropout rate of 0.2 after Conv1D layers Dropout rate of 0.3 before the final dense layer
Bidirectional LSTM Layer (1D CNN-BiLSTM)	64 units, return_sequences=True
Dense & Hidden Layers	Dense layer with 128 units and ReLU activation in hidden layers. Output layer with number of units equal to the number of classes, Softmax activation
Regularization	L2 regularization, weight decay = 0,001
Optimizer	Adam optimizer, learning rate = 0,001
Callbacks	ReduceLROnPlateau with patience = 5
Cross-Validation	Stratified K-Fold with 10 folds
Epochs & Batch Size	30 epochs, batch size = 32

Table 19. Model configuration for 1D CNN & 1D CNN-BiLSTM

Appendix IV - Gaussian Noise

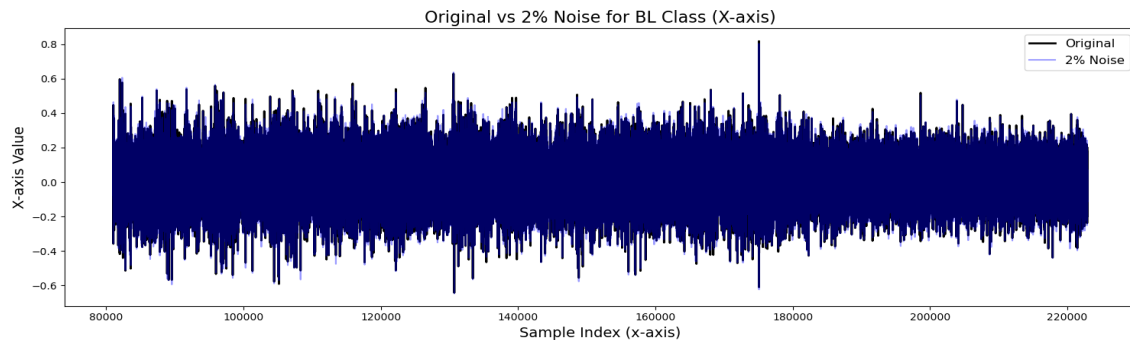


Figure 22. 2% Gaussian noise added to Baseline 2 (X-axis)

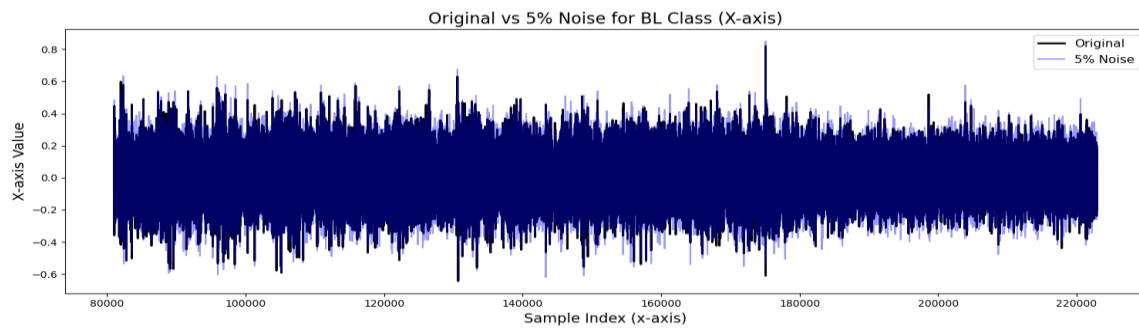


Figure 23. 5% Gaussian noise added to Baseline 2 (X-axis)

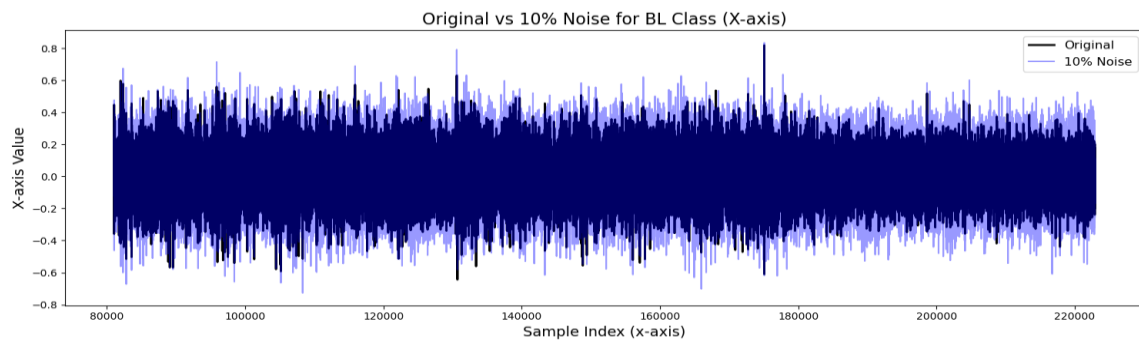


Figure 24. 10% Gaussian noise added to Baseline 2 (X-axis)

References

- [1] Patalas-Maliszewska, J. & Łosyk, H. (2024). *Changes in Sustainable Development in Manufacturing in Cases of Unexpected Occurrences—A Systematic Review*. Sustainability, 16(2), p.717. DOI: <https://doi.org/10.3390/su16020717>
- [2] Polese, F. Gallucci, C. Carrubbo, L. & Santulli, R. (2021). *Predictive Maintenance as a Driver for Corporate Sustainability: Evidence from a Public-Private Co-Financed R&D Project*. Sustainability, 13(11), 5884. DOI: <https://doi.org/10.3390/su13115884>
- [3] Rao, S. N. V. Kumar, Y. V. P. Amir, M. & Muyeen, S. M. (2024) *Fault detection and classification in hybrid energy-based multi-area grid-connected microgrid clusters using discrete wavelet transform with deep neural network*. Electrical Engineering. DOI: <https://doi.org/10.1007/s00202-024-02329-4>
- [4] Ahmad R & Kamaruddin S. (2012). *An overview of time-based and condition-based maintenance in industrial application*. Computers & industrial engineering. 2012;63(1):135-49
- [5] Kane, A. Kore, A. Khandale, A. Nigade, S. & Joshi, P.P. (2022). *Predictive Maintenance using Machine Learning*, Computer Engineering Department, Pune Institute of Computer Technology. DOI: 10.48550/arXiv.2205.09402.
- [6] Satwaliya, D.S. Thethi, H.P. Dhyani, A. Kiran, G.R. Al-Tae, M. & Alazzam, M.B. (2023). *Predictive maintenance using machine learning: a case study in manufacturing management*. 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering. DOI: 10.1109/ICACITE57410.2023.10183012.
- [7] Bukhsh, Z.A. Saeed, A. Stipanovic, I. & Doree, A.G. (2019). *Predictive maintenance using tree-based classification techniques: A case of railway switches*. Transportation Research Part C: Emerging Technologies, 101, 35-54.
- [8] Trojan, F. & Marcal, R.F.M. (2017). *Proposal of Maintenance Types Classification to Clarify Maintenance*. Journal of Business and Economics, 8(7), 560-572.
- [9] Goodlin, B.E. Boning, D.S. Sawin, H.H. & Wise, B.M. (2003). *Simultaneous Fault Detection and Classification for Semiconductor Manufacturing Tools*. Journal of The Electrochemical Society, 150, G778.

- [10] Yuval, M. Yaman, B & Tosun, Ö. (2022). *Classification Comparison of Machine Learning Algorithms Using Two Independent CAD Datasets*. Mathematics, vol. 10, no. 3, p. 311, 2022. DOI: <https://doi.org/10.3390/math10030311>
- [11] Raschka, S. & Mirjalili, V. (2017). *Python Machine Learning*. 2nd ed. Packt Publishing.
- [12] Müller, A. & Guido, S. (2017). *Introduction To Machine Learning With Python*. 2nd ed. O'Reilly Media.
- [13] Reddy, R.V.K. & Babu, U.R. (2018). *A review on classification techniques in machine learning*. International Journal of Advance Research in Science and Engineering, 7(3), pp.40-47.
- [14] Alnuaimi, A. F. A. H. & Albaldawi, T. H. K. (2024). An overview of machine learning classification techniques. BIO Web of Conferences, 97(4), 00133. DOI: 10.1051/bioconf/20249700133
- [15] Purwono, P. Ma'arif, A. Rahmiani, W. Fathurrahman, H.I.K. Frisky, A.Z.K. & ul Haq, Q.M. (2023). *Understanding of Convolutional Neural Network (CNN): A Review*. International Journal of Robotics and Control Systems, January 2023. DOI: 10.31763/ijrcs.v2i4.888.
- [16] Chao, Q. Tao, J. Wei, X. Wang, Y. Meng, L. & Liu, C. (2020). *Cavitation intensity recognition for high-speed axial piston pumps using 1-D convolutional neural networks with multi-channel inputs of vibration signals*. Alexandria Engineering Journal, Volume 59, Issue 6, December 2020, Pages 4463-4473. DOI: <https://doi.org/10.1016/j.aej.2020.07.052>.
- [17] Chen, C. Liu, Z. Yang, G. Wu, C. & Ye, Q. (2021). *An Improved Fault Diagnosis Using 1D-Convolutional Neural Network Model*. Electronics, 10(1), p.59. DOI: <https://doi.org/10.3390/electronics10010059>.
- [18] Obeidat, Y. & Alqudah, A.M. (2021). *A hybrid lightweight 1D CNN-LSTM architecture for automated ECG beat-wise classification*. Traitement du Signal, 38(5), pp.1281-1291. DOI:10.18280/ts.380503
- [19] Choudakkanavar, G. & Mangai, J. A. (2022). *A Hybrid 1D-CNN-Bi-LSTM based model with spatial dropout for multiple fault diagnosis of roller bearing*. International Journal of Advanced Computer Science and Applications (IJACSA), 13(8), pp.637-644. DOI:10.14569/IJACSA.2022.0130873

- [20] Hasija, V. Chamola, V. Mahapatra, A. Signal, A. Goel, D. Huang, K. Scardapane, S. Spinelli, I. Mahmud, M. & Hussain, A. (2023). *Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence*. Cognitive Computation, 16(3). DOI:doi:10.1007/s12559-023-10179-8
- [21] Rudin, C. & Radin, J. (2019). *Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition*. Harvard Data Science Review, 1(2). DOI: doi:10.1162/99608f92.5a8a3a3d
- [22] Zafar, A. Aamir, M. Nawi, N.M. Arshad, A. Riaz, S. Alruban, A. Dutta, A.K. & Almotairi, S. (2022). *A Comparison of Pooling Methods for Convolutional Neural Networks*. Applied Sciences, 12(17), p.8643. DOI: <https://doi.org/10.3390/app12178643>
- [23] Ioffe, S. & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, pp. 448-456. Available at: <https://arxiv.org/abs/1502.03167>
- [24] Demir-Kavuk, O. Kamada, M. Akutsu, T. & Knapp, E.W. (2011). *Prediction using step-wise L1, L2 regularization and feature selection for small data sets with a large number of features*. BMC Bioinformatics, 12, p.412. DOI: <https://doi.org/10.1186/1471-2105-12-412>.
- [25] Kingma, D.P. & Ba, J. (2014). *Adam: A method for stochastic optimization*. International Conference on Learning Representations (ICLR). Available at: arXiv:1412.6980.
- [26] Sarkar, A. Sarkar, A. & Balasubramanian, V. N. (2022). *Leveraging Test-Time Consensus Prediction for Robustness against Unseen Noise*. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp.1839-1848.
- [27] Ding, X. Lin, L. He, D. Wang, L. Huang, W. & Shao, Y. (2020). *A weight multinet architecture for bearing fault classification under complex speed conditions*. IEEE Transactions on Instrumentation and Measurement, 70, p. 3505111. DOI: <https://doi.org/10.1109/TIM.2020.3026461>
- [28] Vakili, M. Ghamsari, M. & Rezaei, M. (2020). *Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification*. 1st ed. Tehran: University of Science and Culture; Sharif University of Technology; Khorasan Institute of Higher Education. DOI:10.48550/arXiv.2001.09636