Combining Expert Knowledge and Machine Learning for Analysing Width Defects in a Hot Strip Mill

Prifysgol Abertawe
Swansea University

Samuel Latham

Faculty of Science and Engineering
Swansea University

Academic Supervisors:

Prof. Cinzia Giannetti

Prof. Cameron Pleydell-Pearce

Industrial Supervisor:

Richie Hart (Tata Steel UK)

This dissertation is submitted for the degree of
Engineering Doctorate

March 26, 2025

Abstract

Thousands of steel strips are processed through the Port Talbot Steelworks' Hot Strip Mill every year. Hundreds of these, however, are either scrapped, sold at a concession, or require further processing due to a variety of defects, a significant percentage of which are width-related. Tata Steel has identified this process as being abundant with underutilised data and is currently investing in innovative research and development which aims to address this issue as well as other industry-wide challenges including decarbonisation and digitisation of legacy processes.

Currently, Artificial Intelligence and Machine Learning has become a popular technology for identifying and analysing defects in manufacturing processes. However, the scope of applications which utilise these technologies within the steel industry are currently limited and are often contained to specific subprocesses. Simultaneously, gathering expert knowledge from employees with first-hand experience is crucial to understanding defects and how they can be analysed.

This thesis aims to expand the scope of Machine Learning tools for defect analysis in steel-making processes and showcase their potential when combined with expert knowledge on a process-wide scale. This is achieved by understanding the challenges of collecting and utilising industrial data, exploring the technologies available for various types of analyses, and reviewing the history and development of these technologies in manufacturing and in the steel industry. From this, technologies and analyses specific to the Port Talbot Hot Strip Mill process are selected, providing a platform on which tools for the analysis of width-related defects are developed.

An in-depth understanding of these defects and their root causes is gained through continual communication with employees at the Port Talbot Steelworks and, using this, several Machine Learning models are created for the purpose of identifying them in several Hot Strip Mill subprocesses. Further to this, a new, data-driven decision-making process is developed for the purpose of identifying width-related defects and determining their root causes across the entire Hot Strip Mill process as they occur.

Declaration and Statements

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ... [redacted] Date ....26/3/2025................

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .. [redacted] . Date ....26/3/2025................

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loans **after expiry of a bar on access approved by the University**.

Signed .. [redacted] .. Date ....26/3/2025................

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Signed .. [redacted] .. Date ....26/3/2025................

Acknowledgements

I express my deepest gratitude to my first academic supervisor, Prof. Cinzia Giannetti for her extensive support, guidance, and encouragement since the beginning of this project. I also thank my Industrial supervisor, Richie Hart, and the team at Port Talbot Steelworks for my access to their knowledge and experience, and for allowing me the opportunity to gain industrial experience in such a key company in my homeland of Wales.

Secondly, I thank the Materials and Manufacturing Academy (M2A) for funding this project and for giving me this opportunity, which has been the platform from which I have launched my career in manufacturing research.

Thanks also to colleagues from my current employment who have supported me during the write-up stage of this project, particularly Bobby M. for his consistent encouragement since I began this role.

Last but certainly not least, I express enormous gratitude to my family and friends for their continued moral support. I thank my cohort friends, Alex B. and Sedar D., in particular for sharing both the university and EngD experience with me through many coffee-fuelled months of writing and head-scratching. Thanks also to my very good friends, Jack T. D. and Alex C., for being consistently supportive and for keeping me sane and sociable throughout my university experience. Thanks to my brothers, Joseph and Daniel, who I am always looking up to. Finally, a huge thank you to my mum and dad, Maha and Tony, for their invaluable support and for always pushing me to be my best.

Contents

List of Figures

List of Tables

Abbreviations

5-FCV	5-Fold Cross Validation
AI	Artificial Intelligence
ANC	Anti-Necking Control
ANN	Artificial Neural Network
BDA	Big Data Analytics
CNN	Convolutional Neural Network
DC	Down Coiler
DL	Deep Learning
FM	Finishing Mill
GSO	Grid Search Optimisation
HOCV	Hold Out Cross Validation
HSM	Hot Strip Mill
I4.0	Industry 4.0
K-FCV	K-Fold Cross Validation
KNN	K-Nearest Neighbour
ML	Machine Learning
RCA	Root Cause Analysis
ReLU	Rectified Linear Unit
RM	Roughing Mill
SVM	Support Vector Machine

1 Introduction

In 2023, approximately 1885 million metric tonnes of crude steel were produce globally, with this figure growing almost every year [12]. While the Tata Steel group has managed to remain among the top producing companies in the world, the United Kingdom has been steadily descending the list of top producing countries in recent years, moving down from 25th to 28th between 2021 and 2022 alone [12]. Numerous other manufacturing sectors, including construction, automotive, and electrical manufacturers [13], rely on the production and continuous supply of high quality steel products with ever-developing properties and coatings [14]. This has created a global push for large-scale but rapid innovation in steel-manufacturing such that competition between companies and countries is at an all time high. Among other reasons, including shifts towards composites and the 2008 financial crash, this has unfortunately contributed towards the United Kingdom's decline in global steel production rankings [15].

Another major challenge starting to be addressed is sustainability and decarbonisation of the steel industry. In 2023, it was reported to the UK parliament that the steel industry constitutes 14.2% of the manufacturing sector's greenhouse has emissions, and 2.4% of the UK's total greenhouse gas emissions [16]. While the UK Government and UK steel manufacturers themselves have set out road maps for decarbonisation, it was reported in 2023 that the Welsh steel industry is at risk of falling behind in these plans due to inaction [17]. The Welsh Government also reported that most increases and contributions to greenhouse gas emissions over recent years have been caused by increased emission outputs from steel production, power stations, and road transport [18].

In recent years, Neath Port Talbot has been the second highest producer of greenhouse gas emissions in the UK, second to London, due to industrial activity [19]. Within this area, Port Talbot Steelworks, owned by Tata Steel, is a major contributor to these emissions. However, Tata Steel and engineers at Port Talbot Steelworks are currently leading major efforts which attempt to address both quality and decarbonisation challenges through various technological developments. The most prominent example of this is the announcement of new electric arc furnaces to replace older blast furnaces. These are to be installed in the coming years and aim to reduce Wales' carbon emissions by over 20%. To address challenges rooted in the competition for innovation and quality, Port Talbot Steelworks is investing in research in a number of fields including materials, chemistry, and digitisation.

With regards to digitisation in steel manufacturing, applications for Artificial Intelligence

(AI) and Machine Learning (ML), are limited to a small number of use cases, namely surface defect detection, and these types of applications are not yet widespread within the steel industry. Tata Steel, as well as other steel-making companies and companies in other industries, have also determined that historical data is a valuable asset which should be capitalised on. In steel-making, the HSM process is a process that is key to ensuring that final product requirements, namely shape and surface characteristics and quality, are met to a suitable, if not exceptional, standard. Port Talbot Steelworks has identified this process to have an abundance of historical data which is currently being underutilised in the competition of innovative steel-making. This thesis therefore focuses on creating and integrating innovative digital tools into the Port Talbot HSM process with the aim of utilising both this historical data and knowledge derived from the expertise of operators and analysts who have been employed at the site for a number of years. The HSM process is described in detail in the following section. The thesis also aims to address challenges associated with limited data availability in applications in the steel industry. This is typically due to company data being, understandably, kept private and internal, and due to public datasets being largely limited to the most popular steel manufacturing applications of ML, namely surface defect detection [20]. These challenges and applications of ML in manufacturing and the steel industry are discussed in further detail in the following chapter.

1.1 Overview of the Hot Strip Mill process

In the HSM process, steel bars are collected either immediately from the casting process, or from the slab yard, and, through a number of subprocesses, are converted into a strip of a given length, width, and thickness. The strip is then wound into a coil and stored before either being further processed or sold to a customer. Port Talbot Steelworks outputs approximately three million tonnes of steel products each year, all of which go through the HSM process [21]. Figure 1.1 shows a diagram of this process. Within this process, hundreds of strips, equating to thousands of tonnes, are affect by width-related defects.

The subprocesses within the HSM process are completed in a fixed, sequential order. A bar is first passed into the HSM process along with scheduling data available after the casting process. It is then reheated in a furnace in which the bar's temperature is recorded. The bar is passed through a Horizontal Scale Breaker where scale is removed using steam. It is then passed back and forth several times between vertical edger rolls in the Roughing Mill (RM), with each pass bringing the slab closer to a required width. Width, roll force, and many other measurements are recorded across the length of the bar during each pass.

A set of spread/squeeze rules dictate whether a certain slab profile can be used to roll the required strip profile. This should be determined before the slab enters the HSM process.

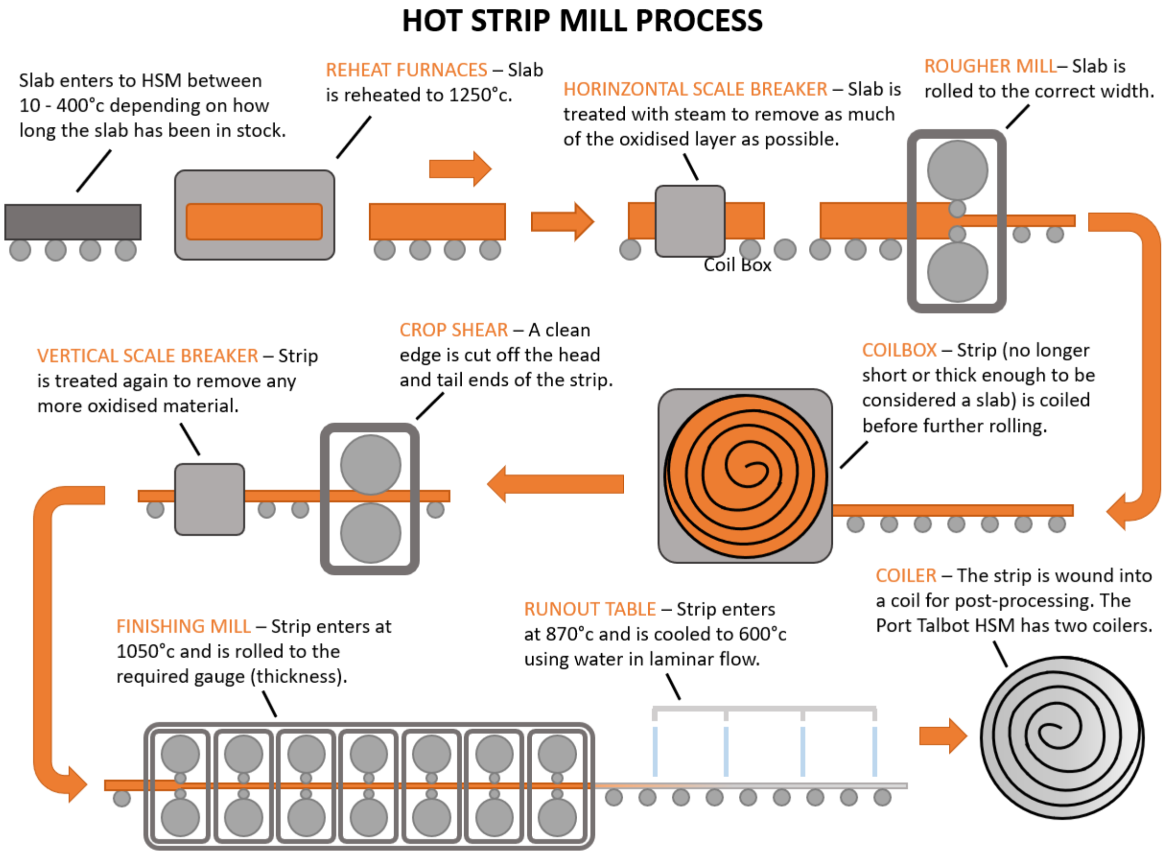

HOT STRIP MILL PROCESS
Slab enters to HSM between 10 - 400°c depending on how long the slab has been in stock.
REHEAT FURNACES – Slab is reheated to 1250°c.
HORINZONTAL SCALE BREAKER – Slab is treated with steam to remove as much of the oxidised layer as possible.
ROUGHER MILL– Slab is rolled to the correct width.
Coil Box
VERTICAL SCALE BREAKER – Strip is treated again to remove any more oxidised material.
CROP SHEAR – A clean edge is cut off the head and tail ends of the strip.
COILBOX – Strip (no longer short or thick enough to be considered a slab) is coiled before further rolling.
FINISHING MILL – Strip enters at 1050°c and is rolled to the required gauge (thickness).
RUNOUT TABLE – Strip enters at 870°c and is cooled to 600°c using water in laminar flow.
COILER – The strip is wound into a coil for post-processing. The Port Talbot HSM has two coilers.

Figure 1.1: Diagram of the Port Talbot HSM process [1].

The bar, now considered a strip due to its length, is rolled into a coil before the head and tail end are cut in the Crop Shear subprocess to create a straight edge. The Finishing Scale Breaker removes scale using steam again. The strip is rolled to the required thickness in the Finishing Mill (FM), which is made up of seven stands, labelled 5 through 11. Width, roll force, temperature, and other measurements are recorded across the length of the bar at each stand. The strip is then passed over a runout, or laminar flow, table where water is used to cool the strip. The final subprocess in the HSM process is the Down Coiler (DC), commonly referred to as the Coiler, in which the strip is wound into a coil before being stored or passed over to the Cold Rolling process. Width, temperature, and other measurements are also recorded across the length of the strip in this subprocess. A single single strip takes only a few minutes to be processed through the entire HSM, with each strip beginning the process almost immediately after the previous one finishes. This results in almost continuous production of HSM data which is added to an ever-growing data stores. These data stores and the data produced during the HSM process are described in further detail in Chapter 3 and its following chapters.

1.2 Objectives

The overall aim of thesis is to gain an understanding of the underlying concepts and challenges associated with data and applications of ML in manufacturing and the steel industry, and thus to identify gaps in research, before using this knowledge to approach practical experiments which aim to address these gaps and produce novel tools for the purpose of improving defect analysis in the HSM process. The thesis aims to accomplish this through the following objectives which are to be evaluated in the final chapter:

1. Conduct a review into the potential and challenges of collecting and utilising data in a real-world manufacturing environment. Further to this, investigate and analyse the development and current state of AI and ML in the manufacturing and steel-making industry in order to identify technologies which could be used to address challenges and improve processes related to quality control in the Port Talbot HSM.

2. Gather expert knowledge and develop novel tools for the detection and analysis of width defects and their root causes using those technologies reviewing in the previous objective.

3. Develop a tool, using the Port Talbot HSM's web infrastructure, by combining the knowledge gained and models created from the previous objective to identify width defects and their potential root causes throughout the entire HSM process.

1.3 Thesis Layout

This thesis describes the research and development carried out over the duration of the EngD project. Chapters are structured such that they correspond clearly to the research objectives stated in Section 1.2. The work and findings produced in each chapter have contributed to several research papers, two of which are conference papers, and one a journal article. The thesis follows the structure described below.

Chapter 2 introduces Industry 4.0 (I4.0) and the technologies from which it is constructed, namely BDA. Methodologies, potential, and challenges of BDA are discussed in detail, outlining some of the main challenges associated with the work carried out in this thesis and with data-driven decision-making in manufacturing. The technologies used for BDA and data-driven decision-making in industry, included those which are considered for use in the practical experiments in this thesis are described in detail. These included ML, Artificial Neural Network (ANN)s, and Computer Vision. The current state of these technologies and their applications in the steel industry are taken into account, highlighting the current research gap which the objectives of this thesis intend to address.

In Chapter 3, the approach and specific techniques used in the following chapters are described in greater detail along with justifications for their use compared other available technologies. The chosen approach follows a traditional BDA methodology of data collection, data management, modelling, and results analysis and visualisation. These provide a foundation from on which tangible steps are taken to complete the practical experiments carried out throughout the thesis.

In Chapter 4, width defects that can be found throughout the HSM process are described in detail. This includes the current process of detection and inspection carried out by operators and analysts, thus capturing expert knowledge which is integrated into a new data-driven decision-making process created in Chapter 6. For those defects which are identified to require further analysis due to their ambiguity and complexity, ML models are developed using the selected technologies described in Chapters 2 and 3. The selected technologies for ML, optimisation, and validation are reviewed and the steps required to collect data are described in detail. Data pre-processing steps are described and test results are reviewed to determine how well and reliably each model performed. A critical analysis of the findings of this chapter is provided at the end of this chapter to conclude.

In Chapter 5, the potential root causes of the defects discussed in Chapter 4 are described in detail. This also included the methods currently used by HSM operators and analysts to capture and analyse occurrences of these defects and how they may be resolved. Expert knowledge gained from these findings is also integrated into the data-driven decision-making process created in Chapter 6. The technologies selected for use in the models in this chapter are described in detail, followed by the steps for pre-processing the data used to train and test each model, and an analysis of the test results to determine how well the model performed. The findings of this chapter are also critically analysed to conclude.

In Chapter 6, the current infrastructure and procedures for collecting data for occurrences of defects in the HSM is reviewed along with its limitations and access permissions. Within this chapter, a new data-driven decision-making process is introduced and is constructed from a combination of the expert knowledge gained and ML models created in Chapters 4 and 5. This process is integrated into a practical tool, which is constructed using the same web-based infrastructure used in the Port Talbot HSM for simpler integration into the current system should the company wish to deploy it. Specifically, this tool aims to replicate, and therefore assist in, the visual inspection processes carried out by experts in the HSM. For testing purposes, and due to the access permissions and limitations discussed in the early stages of Chapter 6, the tool is run in a local web environment. The technologies used to create this environment and the tool itself, as well as the decision-making process are described in great detail throughout this chapter. In the final sections of Chapter 6, the created tool is showcased using new, unseen data collected independently after the experiments in Chapters 4 and 5. The tool, and thus the performance of the new data-driven decision-making process, is the critically analysed to conclude the chapter.

The work and findings of this thesis are summarised and reviewed in Chapter 7. These are then evaluated against the objectives established in Section 1.2 and potential future work is discussed.

1.4 Research Contributions

1.4.1 Research Papers

1. **Pre-Trained CNN for Classification of Time Series Images of Anti-Nacking Control in a Hot Strip Mill**, Samuel Latham & Cinzia Giannetti [22] (Conference Paper in Proceedings of the 9th IIAE International Conference on Industrial Application Engineering 2021 (ICIAE 2021))

2. **Root Cause Classification of Temperature-related Failure Modes in a Hot Strip Mill**, Samuel Latham & Cinzia Giannetti [23] (Conference Paper in Proceedings of the 3rd International Conference on Innovative Intelligent Industrial Production and Logistics 2022 (IN4PL 2022))

3. **A Tool to Combine Expert Knowledge and Machine Learning for Defect Detection and Root Cause Analysis in a Hot Strip Mill**, Samuel Latham & Cinzia Giannetti [1] (Journal Article in SN Computer Science)

1.4.2 Conference Presentations

1. M2A Annual Conference 2019

2. M2A Annual Conference 2021

3. International Conference on Industrial Application Engineering 2021 (ICIAE 2021)
4. M2A Annual Conference 2022
5. International Conference on Innovative Intelligent Industrial Production and Logistics 2022 (IN4PL 2022)

2 Background

In this chapter, the history and development of I4.0 and BDA is discussed in Section 2.1. Detailed overviews of ML, ANNs, and Computer Vision and Image Classification are provided in Sections 2.2, 2.3, and 2.4, respectively, including a short review of their utilisation in industrial applications. In Section 2.5, a review of the history and development of applications for defect detection and Root Cause Analysis (RCA) within the steel industry is provided.

2.1 Data-Driven Decision-Making using Big Data

2.1.1 Big Data Analytics (BDA)

BDA comprises of the processing, analysis, and perhaps most importantly, the understanding of complex large-scale data [24]. It ultimately combines big data and data analytics, encouraging the development of techniques and processes to overcome the challenges associated with these two technological fields in real-world applications, including data collection, noise, and utilisation of results.

A simple description of data analytics is the analysis and visualisation of structured data that can be comprehended and managed simply by humans. The data involved has a limited number of variables and is usually kept within the same data structure with little variety in data types. BDA is used to carry out this task on a much larger scale in more difficult scenarios. Big data is characterised by a large dataset, which is either unstructured or exists within a complex structure, to which a vast amount of data is added to frequently [25]. These characteristics are usually simplified by 5 Vs: Volume, Variety, Velocity, Value and Veracity model [26,27]. Big data is sometimes described using more than 5Vs, however, they are usually synonymous with either the five Vs described here or with the analytics process itself. Big data can also be difficult to process due to null values and anomalies in data.

Volume refers to the usually vast amount of data produced and storage required to contain it [28]. Since data is stored and thereon considered historical, the volume of data can become too large to sort and analyse manually. Depending on the age of data or the infrastructure used to collect it, various data formats and data stores may be available. This can include written documents, spreadsheets, and images, all of which can be considered big data.

While data collected using old infrastructure may result in large volumes of data in different datastores, the data stored in these different formats can all be used in the BDA process. Variety in data, particularly in data types, increases the complexity of data [29]. It should also be noted that different data types and formats can be used in conjunction in the BDA process, whether this entails integers or Boolean values, or images and time series data.

Velocity refers to the rate at which data is produced. Within manufacturing in particular, data is being continuously generated at a fast rate by numerous sources including real-time sensors and engineering reports, contributing to an ever-growing datastore. It is also important to consider the speed at which this data can be accessed and processed by the analytics process [30]. The optimal BDA process will be able to access, process, and visualise big data as quickly as possible, or in real-time if being used in parallel with a production process.

Value refers to the difficulty of determining the usefulness of data, especially in large datastores. One of the main challenges of BDA is understanding data and whether it can make an accurate contribution towards analysis. It is this however that determines its value, as simply collecting data without an analysis or without providing an insight into its use holds no value to a company [31].

Veracity is the biggest obstacle in BDA [32]. It describes the noise that may be present in big data which can determine the difficulty and time consumed by the BDA process, particularly when pre-processing data. The accuracy of data can be severely affected by imperfections including anomalies and null values. Data is carefully pre-processed in an attempt to mitigate the effect these problems have on analysis.

2.1.2 General Big Data Analytics Methodology

Generally, the first steps in the methodology of BDA, understanding and collecting the data, are the most difficult [2]. Because datasets are usually large, it can be difficult to determine where data has come from, when and why it was created, and how it relates to other data. Data must first be understood so that it can be used to its full potential in its given context [33]. The next step is to manage and process the data to make it complete. It is common that big data contains poorly formatted, missing, and anomalous values which make an analysis inaccurate [34]. It is therefore necessary to remove these values to produce a more accurate analysis or to allocate new, updated values to produce a fairer analysis which attempts to utilise as much existing data as possible. This is usually the most time-consuming step of the process.

Next, models are created and validated using a technology suited to the analysis to be performed in the given scenario [35]. For example, a simple conditional AI check or an image classification model. In this step, new data related to the previously collected and processed data is used to determine whether the created model performs accurately and reliably. An example of this is Cross Validation in ML, in which a model is trained using a set of features derived from data and is validated using the same features extracted from new, unseen data [36]. This is discussed in greater detail in Sections 2.2 and 3.3.3.

The final step is to produce feedback for the intended analysis and visualise results. How this step is completed is dependent upon the intended use case and the data outputted by the analysis. Its purpose is to assess what has been discovered from the analysis, why these discoveries are important, and how these discoveries might aid in future analysis [2]. This framework is followed throughout the course of this thesis, enabling a structured, conventional approach towards these four BDA components. Figure 1.1 illustrates this methodology.

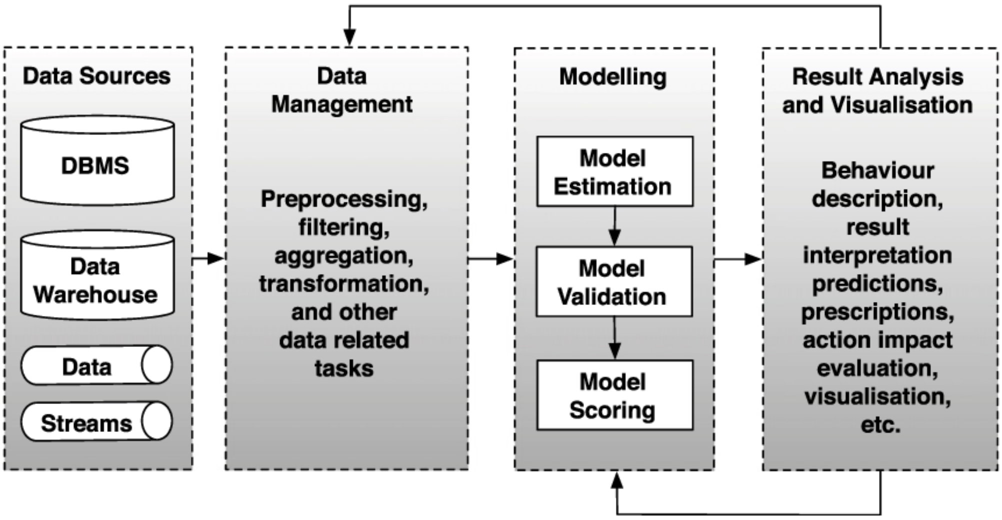

Data Sources
DBMS
Data Warehouse
Data
Streams
Data Management
Preprocessing, filtering, aggregation, transformation, and other data related tasks
Modelling
Model Estimation
Model Validation
Model Scoring
Result Analysis and Visualisation
Behaviour description, result interpretation predictions, prescriptions, action impact evaluation, visualisation, etc.

Figure 2.1: Overview of the analytics workflow for Big Data [2]. The steps are Data Sources, Data Management, Modelling, and Results Analysis and Visualisation, respectively.

2.1.3 Potential and Challenges

Many industries use BDA today including manufacturing, healthcare and education. In the entertainment industry, the success of shows is determined based on trends between episodes such as audience numbers and new memberships for streaming services [37]. Social media companies find the latest and most popular trends based on publicly available data such as retweets, mentions, and hashtags [38]. BDA is also used to improve illness diagnosis and patient condition monitoring in the healthcare industry [39], and to predict the flow of travel and arrival times of public transport using passenger-related data in the transportation industry [40]. In education, it is used to support learning processes by analysing student performance in real-time [41]. In sports, it is used to determine trends in competitor, player, and team performance such as scores, rebounds, and assists [42].

Although these are clear examples that BDA can be used to provide industries with new paths to development, efficiency, and revenue, it is still not without its challenges. As described in Section 2.1.1, the main challenges associated with BDA derive from the 5 Vs [26]. However, there are still several other challenges which arise from its use in real-world environments, including the complexity of planning and implementation, evaluation and utilisation of deployed systems, and the security of internal data [43], a problem which cyber security, another I4.0 technology, aims to address.

2.1.4 How Big Data Analytics is Used in Manufacturing

Manufacturing companies use BDA to monitor and control production processes, often utilising data collected in real-time time using advanced sensors. Steel manufacturers in particular can potentially yield huge savings using BDA technologies, both in terms of cost and material. In an article describing a prediction control process for steel production, Saarstahl AG, Krumeich et al [44] explain how businesses must tackle economic, environmental and manufacturing problems proactively to compete in today's market. It is suggested that event-driven architecture has the potential to increase the flexibility and speed of manufacturing processes. This is a reactive approach to BDA which attempts to mitigate the effects of problems in the manufacturing process as they occur. Waiting for events to occur like this can cause latency between decision-making processes and production line feedback optimisation.

Data-driven decision-making has become more favoured recently as it is capable of providing immediate feedback for process adaptation, using output data to determine the following actions in a process rather than relying solely on expert knowledge [45]. Manufacturing companies using event-driven architectures normally utilise a combination of this knowledge and historical data to make process decisions. Data-driven decision-making, however, uses real-time data for immediate feedback and optimisation. As opposed to event-driven architecture, this approach aims to proactively avoid production problems, and therefore their effects, altogether. For some manufacturing companies using event-driven architecture on a large scale, introducing new and large-scale technologies and systems into an existing and established process is not feasible, and therefore makes integrating data-driven decision-making a challenge [46]. In a study carried out in 2015, it was determined that 90% of all surveyed data at the time had only been gathered in the few years prior [47] and that many industries, especially manufacturing, attempt to use ML to improve decision-making processes by utilising this data. ML has become an extremely popular technology to use within BDA due to its ability to automate the learning of patterns and features from vast amounts of complex data.

Within the last decade companies with the means to do so have strived to understand big data and integrate its use into processes with some success [48]. To achieve this, an efficient method of analysis is necessary. I4.0, and BDA in particular, aim to increase process agility, enabling quicker adaptation to ever-changing circumstances [3]. This is illustrated in Figure 2.2. The main goals of these technologies are to maximise the use of available data while predicting and minimising process issues.

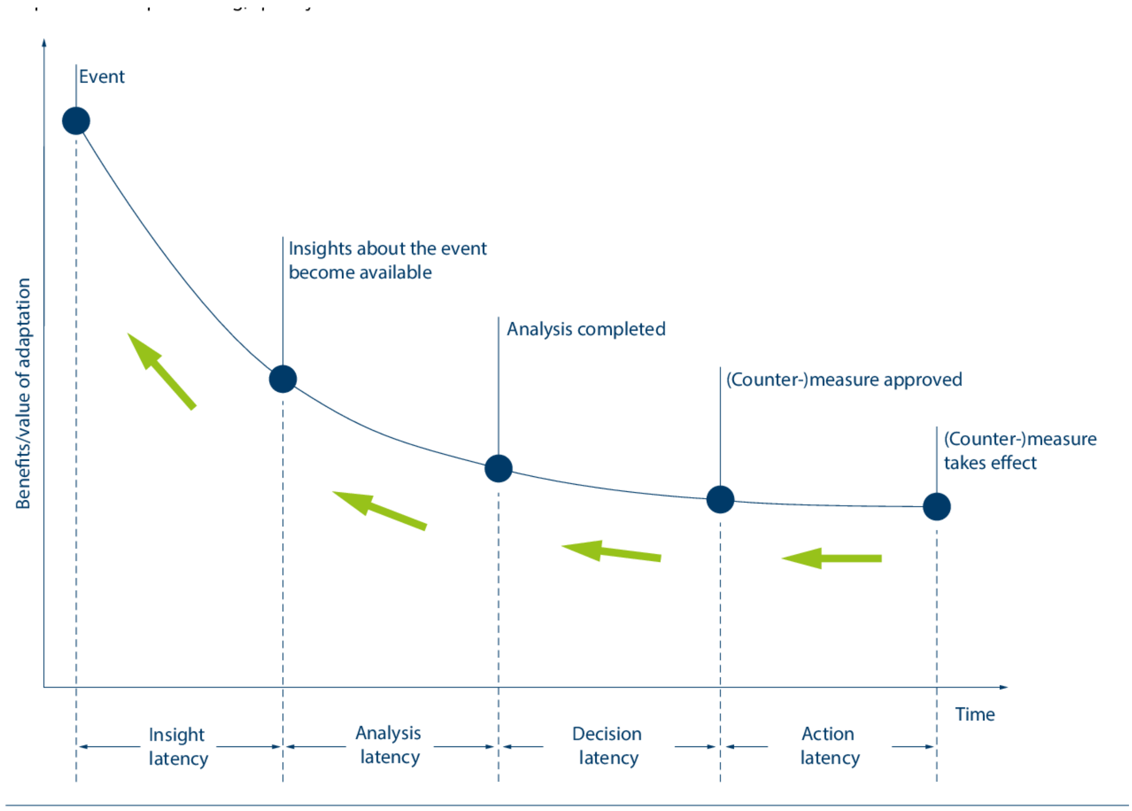

Event
Insights about the event become available
Analysis completed
(Counter-)measure approved
(Counter-)measure takes effect
Benefits/value of adaptation
Time
Insight latency
Analysis latency
Decision latency
Action latency

Figure 2.2: Corporate adaption processes. Delays occur before decisions and countermeasures can be made when events occur within a business. This is due to limited integration of information systems used to capture, process, and convey relevant and useful information across businesses [3].

2.2 Machine Learning

ML aims to use data as a means to learn rules and patterns [36] and encourage more evidence-based decision-making through the use of big data [49]. It is a branch of AI and is used to learn patterns in data to perform a task without being given exact, hard-coded instructions. AI simply aims to solve computational problems by emulating human behaviour. ML achieves this by training a model using a chosen algorithm and a training dataset from which to learn. This model can then be used to perform tasks on new, unseen data, commonly referred to a testing dataset.

Initial ideas for ML developed from the structure of neural activity in 1949 [50] effectively describing the relationship between learning and the activation strength of neurons in the brain, affecting how future decisions are made. In the following decade, new AI developments began to incorporate this idea through scoring algorithms and dynamic decision-making [51]. This also led to the creation of the first ANN [52] and other ML algorithms including K-Nearest Neighbour (KNN) [53] and K-Means Clustering [54]. ML algorithms gradually introduced new parameters and factors to be considered such as the updating of weights and errors [55] to account for the learning issues typically associated with with traditional AI [56].

There are several benefits of utilising ML within a BDA context. Firstly, it has the capability to utilise historical and big data, maximising the potential of what can be considered company assets. Secondly, it can also be used for data-driven decision-making within the scope of predictive analytics, as well as for reactive, event-driven analytics. This also increases the potential for its use in a broad range of applications in a number of industries. ML models can also be retrained, meaning that they can further use big data to their advantage. This is due to the velocity of big data and because further training and validation can only improve the reliability of the results produced during a model's evaluation.

It is important, however, to consider the drawbacks of such a rapidly evolving technology, which are simultaneously rooted in the main characteristics of big data. Due to the usually large volume of big data, as well as its velocity and veracity, it can be computational expensive and time-consuming to train ML models. This also applies when addressing the value of big data as it requires more time to gain a deep understand of collected data when its size is ever-increasing. Another drawback specific to real-world industrial applications of ML is that it can be difficult to attain or curate datasets, particularly balanced labelled datasets. This can either be due to, again, the characteristics of big data which make it difficult for companies with already large legacy data stores to maximise their use, or due to access and privacy permissions which prevent or limit such data from being shared.

ML can be used to solve or optimise various computational problems including image and speech recognition [57,58]. There are three types of ML: supervised, unsupervised, and reinforcement learning. This section focuses on the use of supervised and unsupervised learning methods since reinforcement learning involves the self-optimisation of learning techniques rather than the grouping or labelling of data [59].

2.2.1 Supervised Learning

Supervised learning algorithms are used to train models using a pre-labelled dataset. The model processes data and makes informed decisions based on what is learned from the training data. Within the model, these decisions are made using functions or weights which are choices or numbers based on the training data, and are used to decide the final output of the ML model. The output of a supervised learning model can either be a classification of input data, or a predicted number or trend depending on the type of supervised learning used [60]. Thus, the two types of supervised learning are classification and regression. It is important to note that while using pre-labelled data to train ML models can be more effective than traditional AI, such a technology is still prone to human error and overfitting [61].

Classification aims to categorise input data based on features and patterns learned from a dataset containing categorical data associated with two or more categorical classes, or labels. In a classification model, input data is processed using the algorithm the model is based on and its weights, and is assigned a probability or number which determines its class, which is the output of the classification model. Classification models have been used in a wide range of applications, including image classification and speech recognition [57,58].

The learning process of regression is similar to classification since supervised learning requires labelled training data. However, regression aims to calculate continuous or dependent outputs based on the relationship between multiple independent features [62]. Regression models are commonly used for predictive purposes such as condition monitoring and forecasting [63]. In this thesis, we have chosen to use classification for the developed ML models [1,22,23]. The driving factors behind this decision are discussed in further detail in Chapters 4 and 5.

2.2.2 Unsupervised Learning

As opposed to supervised learning, unsupervised learning uses unlabelled data to train a ML model. The model learns to categorise data based on its own decisions about what it perceives as distinguishable characteristics [64]. This sometimes requires data dimensionality reduction depending on the complexity of the training data. This entails processing data with high dimensionality, such as data with a large number of variables or features, into lower dimensionality data to remove noise or irrelevant information. Since decisions are derived directly from information provided by the data's characteristics and not labelled data, human error is eliminated from the unsupervised learning process. Semi-supervised learning is also an option in which models are trained using both labelled and unlabelled data [65]. The process of using unsupervised learning to group unlabelled data is called clustering.

In clustering, a number of classes is decided beforehand, and an algorithm is chosen to determine the natural patterns which exist within the data. Since there are no labels involved in unsupervised learning, data is clustered by making decisions based on related properties between samples [54]. Simply, clustering is an informed estimation of data groupings based on characteristics shared between data points. Groupings can be based on density, hierarchy, or simply extrema. Clustering is sometimes used in applications related to performance quality evaluation and customer analysis [66].

2.2.3 Machine Learning in Industry

ML has become very popular in a range of industrial fields. Diagnosis, fault detection, and forecasting are notably among the most popular of industrial applications, especially for those utilising Supervised Learning [67]. Examples include forecasting product demand using a Support Vector Machine (SVM) [68], automated visual inspection of machine components using Naïve Bayes [69], and condition monitoring for mechanical equipment using an ANN [70]. While there is much development on this front, big data and model reliability have been, and often still are, a consistent challenge in manufacturing applications [71]. Some researchers suggest that future research should address this by creating more readily-available datasets and methodologies for solving domain-specific tasks [72].

2.3 Artificial Neural Networks

An ANN is a ML architecture which is based on the architecture of the human brain, as opposed to traditional ML algorithms. They are used to train and create models for recognizing patterns based on trends shown in the data it is representing. Feature extraction or selection is typically performed beforehand when using ANNs, as is the case with most other ML algorithms. Feature extraction can, however, be performed by Deep Learning (DL) ANN architectures which are discussed in the following subsection. ANNs are made up of a number of linked neurons, or nodes, which are combined to form layers, thus creating a network structure [73]. Each node contains a weight value which determines the direction in which data flows through the network. An example of a basic ANN neuron is shown in 2.3 Using an optimisation algorithm, these weights can be updated when processing data through the network, resulting in a trained ANN model.

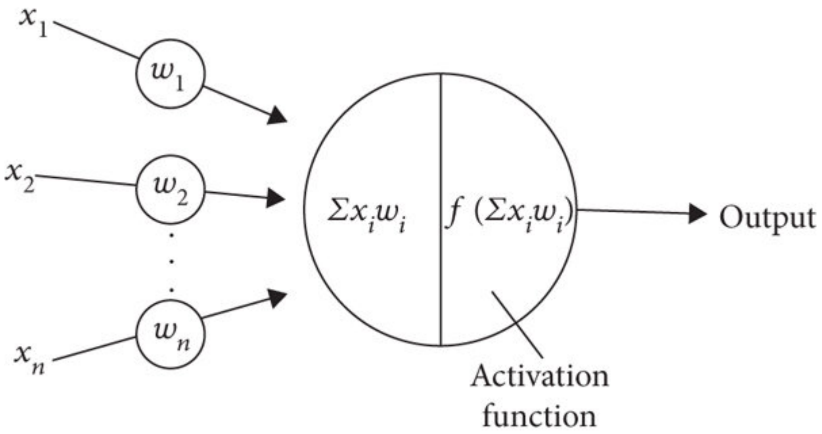


Figure 2.3: Model of an artificial neuron [4] which receives input values, x_n, which are then multiplied using an activation function, $f(\Sigma x_i w_i)$, where w_i is each values corresponding weight value.

While ANNs are defined by their architecture, they still require an algorithm to optimise themselves. Stochastic Gradient Descent with Momentum is one of the most commonly used optimisation algorithms. This method utilises backpropagation in which a training data sample is passed through the network and an output error is calculated. The error is then passed backwards through the network and used to update the weights [74]. This is the training process for ANNs. It is usually safer to train in batches where some or all training data samples are passed through the network before updating its weights. An iteration over all records in the training dataset is called an epoch. There are many hyperparameters, including the maximum number of epochs, which can be specified in neural network training algorithms [75]. Commonly adjusted hyperparameters include learning rate, which decides how much weights are updated by, momentum, which determines how much a weight can change in the same direction based on the previous weight, and learning rate decay, which gradually decreases learning rate over epochs. During training, samples from the dataset are also tested through the ANN to determine whether the output node produces the intended response. This is called validation, and it is performed after each epoch. If the output node produces an incorrect response, the validation error, which is calculated by the sum of errors over all epochs, will increase. The training process ends when the specified number of iterations is reached. This is based on the number of epochs and the batch size, usually specified before training. It can also end when the validation error increases or reaches a specified limit. The network architecture and the resulting weights form the final ANN model which is used to make predictions with test data. A network is tested when new, unseen data is passed through the network without updating the weights.

There are a number of ANN types which can be applied to a range of applications. A CNN, for example, is typically used in image classification due to its sequential structure of nodes which utilise kernels and pooling. The type of ANN used to create the image classification models in this thesis is a CNN, the structure of which is described in greater detail in Section 2.4.2. ANNs, like many other ML models, can also be used to analyse large, complex datasets using trained and evaluated models, addressing some of the main challenges which arise from the characteristics of big data. These factors also make them useful for analysing data which may be ambiguous to humans or non-dynamic AI methods. As with other general ML models, the reliability of ANN-based models will typically improve with further training and validation data. It can still, however, be computationally expensive and time-consuming to train and evaluate ANN-based models. This also applies to the collection and pre-processing of the data used to carry these out. A large amount of training data is usually required to produce a reliable model which can be difficult to collect and understand in real-world or novel applications, although this sometimes depends on the domain and application.

2.3.1 Deep Learning

An ANN's width is determined by the largest number of neurons used in any given layer while its depth is defined by the number of layers it contains, including input and output. An ANN with a depth of 3 or greater is generally referred to as DL, rather than traditional ML. These layers are typically referred to as input, hidden, and output layers [73]. Data is passed from the input layer through the hidden layers for processing, otherwise known as feature extraction, and outputted at the end of the network. This means feature selection is performed by the DL architecture itself during the learning process. Apart from the input layer, all neurons have a weight and an activation function which tells the neuron how exactly to process the data at that point. Data must also be pre-processed to be compatible with the specified network input. For example, an image of a particular size or format.

2.3.2 Artificial Neural Networks and Deep Learning in Industry

Since ANNs and DL are a subset of ML, they follow the same trend in areas in which applications are becoming increasingly popular and adopted in industry [67]. ANNs and DL are thus being used in many fault diagnosis and detection applications as well as for predictive maintenance. There are, however, many other ways in which these technologies can be, and are starting to be, utilised in industrial settings including predictive control in pharmaceutical manufacturing using Recurrent Neural Networks [76], fault classification and diagnosis in semi-conductor manufacturing using CNNs [77], and estimating the remaining useful life of rollers in the steel strip manufacturing using Convolutional-Long Short Term Memory networks [78]. While DL applications within manufacturing and the optimisation of DL algorithms are being continually researched, their popularity within this industry has been steadily increasing over the last decade [79]. Future applications within manufacturing aim to further develop their integration into cloud technologies such that ANN and DL technologies can be developed or accessed using time, cost, and resource efficient on-demand computing services [80].

2.4 Computer Vision and Image Classification

2.4.1 Computer Vision

Computer Vision technologies aim to emulate human vision by extracting and processing image data in order to understand what it represents. There are several Computer Vision technologies including image classification, object detection, and image segmentation. The aim of image classification is to be able to determine what an image represents, whether it is an image of a physical object, an encoding, or any other image representation. This is achieved by creating a classifier, usually with a ML algorithm, which can distinguish between an intended set of classes. Object detection is a subset of image classification in which an image is dissected into local areas. These areas are then treated as individual tasks for image classification. Image segmentation also localizes objects but its aim is to create a pixel-wise map to distinguish objects in detail from the original image.

There are many existing applications of Computer Vision across a wide range of industries. Common day-to-day examples include object detection for self-driving cars [81], classification of medical issues using MRI images [82], and facial recognition in biometric systems [83]. There are also many existing applications of Computer Vision in the manufacturing industry such as bar code reading, quality inspection for circuits, and object detection and tracking for robotics. DL, particularly with CNNs, has become an extremely popular method of training image classifiers over recent years [84].

2.4.2 Convolutional Neural Networks and Image Classification

CNNs are one of the most commonly used neural network models for image classification [85] and are designed to recognise patterns in image data, much like the human eye [86]. Statistical models can also be used to solve image classification problems although they are limited by their ability to reuse contextual information when deriving patterns in data, which CNNs are designed to retain. This is not to say that CNNs are without their drawbacks. Due to the amount of data processing necessary to train such a model, CNNs can be computationally expensive and difficult to implement with challenges such as class imbalances and overfitting being present [87]. These can however be solved with more computational power and data pre-processing. Once data is transformed into the required format, CNNs use multiple hidden layers to extract features and its final layers to output a label for the data based on the identified features. This process is completed over several layers which make up the CNN's structure [88]. In image classification, each layer acts as a filter which changes the network's perspective on the image in order to learn particular features.

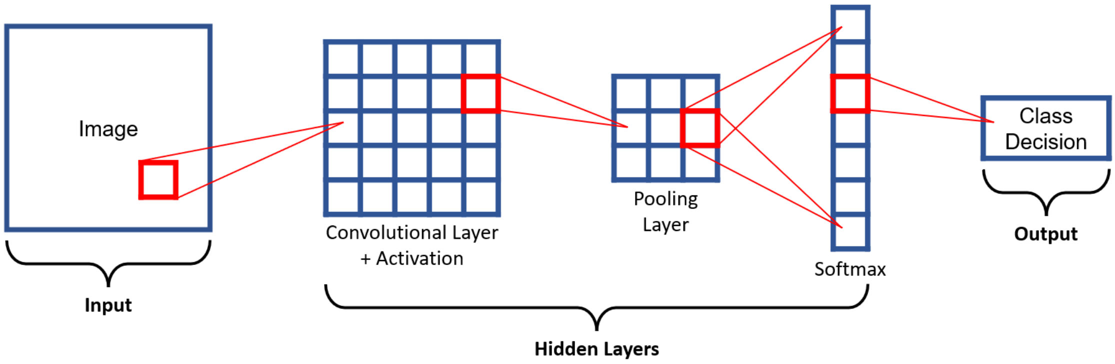


Figure 2.4: Diagram of a basic CNN architecture [1] consisting of an input layer, and output layer, and hidden layers between them consisting of convolutions, activations, pooling, and softmax.

The first layer is an input layer where image data enters the network. Next is a convolutional layer which reveals features by applying filters to the image, converting it into a readable format. Third is a Rectified Linear Unit (ReLU) or activation layer which eliminates redundant data by changing negative values in the image to zero. The fourth layer is a pooling layer which reduces data dimensionality, making features easier to recognise in image sections. The second to last layer is a fully connected layer which outputs the probabilities of the image belonging to each class. The last layer of the network is a softmax layer which outputs the classification decided by the network [1]. A basic CNN structure is illustrated in Figure 2.4. This however is the basic structure of a CNN. There are many architectures which use a mix of combinations tailored to specific classification problems. No one CNN architecture is best for every classification problem [89].

CNN architectures can produce accurate image classification models using a variety of training and optimisation approaches, which are discussed further in Section 3.3. This versatility can result in time and cost-efficient training and deployment. CNN architectures are also adjustable, meaning that parameters for nodes and layers, including the input and output, can be adjusted to suit the application. These include the size of input images and convolutions. As with any ML applications, however, CNNs are not without their drawbacks. These include standard data access and privacy concerns, dependence on data infrastructure in real-world applications, and the possibility that some level of human interaction may still be necessary after deployment to assure model reliability or in cases where data, and thus decision-making, is ambiguous.

In the last decade, CNNs have become widely popular in a range of applications including robotics, bioinformatics, and cyber security [90]. Within manufacturing, they are typically used in object tracking, and part and quality inspection [91]. CNNs are now more commonly used for condition monitoring and defect identification with many use cases being able to process real-time data [92]. CNNs have been used to determine the condition of tools in a CNC machine using time series data in the form of plot images [93]. Similarly, they have been used to classify power quality disturbances based on voltage signals [94, 95]. Transfer learning and other neural network architectures can be combined with CNNs to create hybrid architectures which attempt to utilise the properties and benefits of both architectures.

As with general ML, CNN and Image Classification technologies are being pushed towards integration within cloud services so that computational resources and required functionality can be accessed on-demand. This is already present in today's available technologies, most prominently with large cloud services and packages such as [96] and [97], as well as smaller, application-specific software such as Edge Impulse [98].

2.5 Development of Defect Detection and Root Cause Analysis in Manufacturing and the Steel Industry

While the detection of defects and their root causes has always been a part of manufacturing processes, standardised frameworks became particularly popular in the 1980s soon after the development of Lean 6 Sigma [99]. Such frameworks include a range of defect detection, RCA, and quality control techniques including the 5 Whys [100], an example of an older technique that has been incorporated into the framework. These techniques enable manufacturers to optimise product quality and eliminate various types of waste and inefficiencies from their processes, while providing them with a deeper understanding of these processes and their underlying problems. As discussed in the previous sections of this chapter, tools for defect detection and RCA in manufacturing processes have evolved over the past 75 years to include the use of advanced AI [101]. This has enabled manufacturers to automate data-driven decision-making processes, leading to the development of diagnosis and prescription tools, typically referred to as expert systems [102].

One of the major challenges associated with data in manufacturing processes is its continuous production at such a vast rate. This can make it difficult to analyse and understand data efficiently, and to make effective decisions in a continuously running production line [103]. Such vast amounts of data, particularly legacy data which often contains a lot of noise, is often unstructured and unorganised, and requires a great deal of processing before it can be analysed [104]. Over the last few decades, ML has become an increasingly popular technology to use for automated analytics, enabling users to create bespoke modes for specific analyses and to quickly analyse complex data, sometimes in real-time [105–108]. This can be approached similarly to expert systems in which basic AI, expert knowledge, and ML for ambiguous data can be combined to create powerful tools for BDA, defect detection, and RCA [1].

Within the steel industry, applications for defect detection and RCA typically focus on individual subprocesses and machines. In particular, applications which utilise AI and ML are not yet widespread outside of part quality inspection monitoring, roller model optimisation, and most commonly surface defect detection. In recent decades, Computer Vision has become a popular approach in steel surface defect detection applications [109]. The focus of the research behind some of these applications varies, highlighting that there are various factors to be considered when creating or emulating a visual inspection process. In the literature, a CNN model to determine whether oxidation, pitting, and other defects are present in steel strips [110–112]. Another approach has been to use Object Detection alongside CNNs to capture individual instances of stains, cracks, and other steel surface defects [113–115]. These, and many other applications, demonstrate that emulating and reproducing visual inspection processes using these technologies can have a major impact on the accuracy and efficiency of defect detection and analysis in steel-making, creating potential for further investigations into their occurrences in a repeatable and time-effective manner [116].

Existing applications are also not yet used on such a scale that defect detection, RCA, and operator feedback is an automated, end-to-end task or system [1]. It is important to consider that in AI and ML applications within the steel industry or HSM process, while there is much potential benefit, unreliable systems and poor decision-making can lead to wasted time and materials, as well as negative knock-on effects on the production line, as is the case in any manufacturing process [117].

In the Port Talbot HSM, approximately 7% of strips are found to contain defects annually with approximately 0.4% of these accounting for width defects, equating to thousands of tonnes of steel each year. Depending on the severity of the defect, knock-on effects can include extra time spent on further processing, profit loss from concessions, material wasted by scrapping products, and possible downtime in the HSM. Managers and team leaders estimate that if no width defects were to occur in the Port Talbot HSM, the time and energy saved on further processing and waste could equate to £2.4 million annually.

2.6 Summary and Conclusions

In this chapter, I4.0 was introduced along with the set of technologies from which it is constructed. From this, it was possible focus on BDA, for which a broad but detailed overview has been provided. This served as a major aid towards understanding characteristics of big data and approaches to data collection, processing, and utilisation. Through this, the potential and challenges associated with these steps in real-world environment has been highlighted. Following this, various technologies were considered for use in the practical work in later chapters, including ML, ANNs, and Computer Vision. These were discussed in detail, taking into account their advantages and drawbacks with regards to both their general practicality and their use in industrial applications. Specific technologies were also considered further to these, including Supervised Learning, CNNs, and Image Classification. Finally, the history of these technologies for the purpose of defect detection and RCA within manufacturing has been considered, highlighting the current state and limitations of applications within the steel industry, and the gap in research and development which this thesis aims to address.

This chapter aims to provide reasoning for the pursuit of both the research direction and practical output of this thesis. So far, the main focuses of applications for defect detection and root cause analysis in the HSM process have been surface defect detection, part quality inspection monitoring, and roller model optimisation. The review as highlighted gaps in AI and ML applications for width-related defects. This thesis aims to expand the scope of applications to add steel defects by focusing on these defects and their root causes over the entire HSM process.

Image classification in particular has been chosen to emulate visual analyses of ambiguous time series data which may contain more complex or ambiguous contextual information. This will be performed alongside conditional checks based on expert knowledge in order to construct a well-informed decision on whether defects and their root causes are present in the HSM process.

In the following chapter, Chapter 3, the overall approach to the practical experiments carried out in later chapters, as well as the specific methods and techniques used, are described in greater detail along with justifications which take into account their advantages and disadvantages. This chapter also aims to serve as an introduction to the main technologies which encompass these methods and techniques.

3 Methods and Techniques

The work in this thesis has been approached by using the BDA framework described in Section 2.1.2. In this chapter, the methods and technologies chosen for each BDA component have been discussed in greater detail along with justifications for their use in the development of the proposed tool [1, 22, 23]. Section 3.1 provides a brief overview of the software used to store and collect the raw data from the HSM process. Section 3.2 discusses the chosen methodology for processing and readying time series data for the Modelling stage. While basic AI will be used to perform some checks within the tool, the processing and analysing of time series data is the key to making it a success, hence why it is the main focus of this section. Section 3.3 describes how models will be created using technologies further to those discussed in Chapter 2. It will also outline the available and chosen techniques for optimising and validating these models. Section 3.4 will introduce the technologies used to visualise data and integrate models into the tool.

This methodology enables the development of a structured process for understanding and processing data and expert knowledge, creating models, and feeding back their results. This is illustrated on a high level in Figure 3.1, and the details of this process are described thoroughly in Chapters 4, 5, and 6.

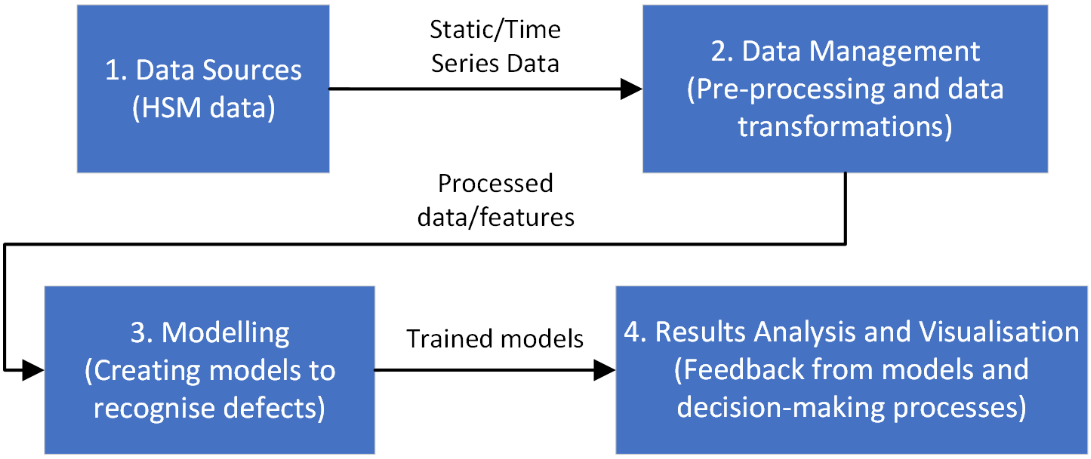


Figure 3.1: High-level flowchart describing how the BDA methodology discussed in this chapter is applied.

3.1 Data Sources

The first step towards building a reliable system for analytics is to assess and understand the origins of the data it utilises. In this section, an overview is provided for the mediums in which data for the Port Talbot HSM is stored, as well as the software used to query and retrieve it. A detailed review of the process for querying and extracting data from these mediums is provided in Chapter 4 prior to practical experiments.

3.1.1 IBA Analyzer and SQL Databases

Within the Port Talbot HSM, there are two main modes of data storage used for different types of data. The first is a database which is stored on a local server within the steelworks site. Other servers and databases exist but only one is used to store HSM data. Within this database, a number of tables are used to store primitive data types. Primitive data types are those which are defined by a single value, including characters, integers, floats, and booleans [118]. The names of the tables in this database and the data they contain are related to the various subprocesses or machines within the HSM process, such as width tolerances, chemical compositions, and order specifications. This data can be accessed using SQL database query software, such as HeidiSQL [119], and a secure connection to the database itself. From this, specific tables and values can be queried and exported into a number of file formats, including .csv and .xlsx, for further use in other development environments.

The other mode of storage is through a data analysis software named IBA Analyzer [120]. This software is primarily used to visualise time series data from HSM processes which is accessed from software-specific .iba files created by data read from sensors in the HSM process which have also been linked to the software. The data stored here can range from width measurements, to temperature levels, to the product's position within the process. This data can be accessed with a secure connection via the IBA Analyzer software. However, in order to export it into another file format useful in other development environments, a separate tool must be used for queries and extraction. Currently, however, a limited number of employees have the permissions to use this tool as too many simultaneous connections and extractions can slow down general access speeds to IBA Analyzer, and thus time series data, for all employees. To query and extract this data, it must therefore be requested directly from an employee with the required permissions. Figure 3.2 illustrates the flow of data when stored and accessed.

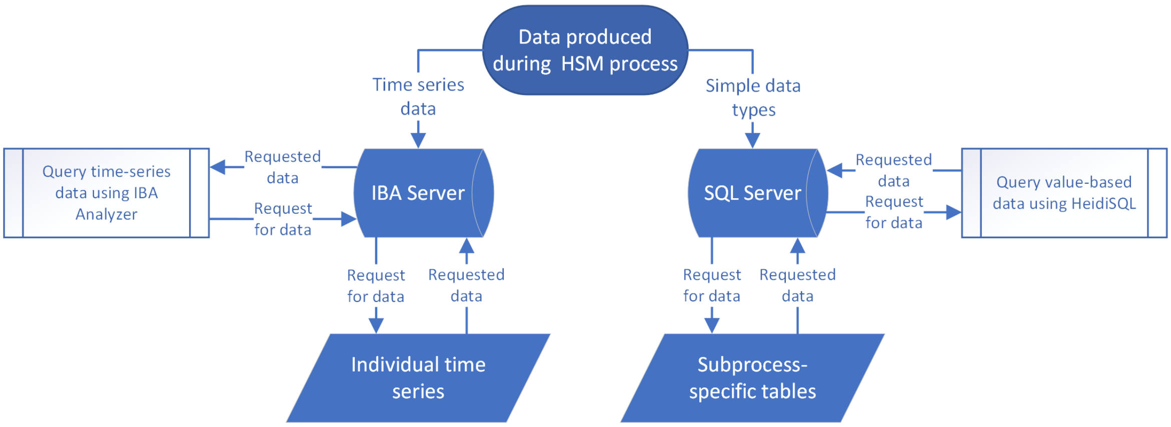


Figure 3.2: Overview of data storage and access in the HSM.

The specific data used in the experiments and proposed tool within this thesis are extracted from these platforms, and the specific data gathered and its use is discussed in further detail in Chapters 4, 5, and 6.

3.2 Data Management

After understanding and capturing relevant data, suitable methods must be chosen to process it and extract important features and information. When choosing suitable processing methods, the type and nature of the data should be considered. In this section, an overview is provided for the nature of time series data, which is the type of a large majority of the data in the Port Talbot HSM, and existing approaches and applications for processing time series data are discussed.

3.2.1 Time Series Images for Classification using Convolutional Neural Networks

A time series is a set of observations that is collected sequentially over a period of time [121]. There are a number of approaches that can be used for processing and analysing time series data within the fields of AI and ML based on the properties of the raw data and how important characteristics might be best projected for AI and ML problems [122, 123], while also considering how specific the data is to the domain in which it is being used.

There are four types of time series behaviours: Trending, cyclical, seasonal, and random. A trending time series shows a linearly increasing or decreasing pattern over the length of the time series. Values in both cyclical and seasonal time series increase and decrease while both types can still be affected by external factors, changes in seasonal time series typically occur over a fixed period of time or frequency, while changes in cyclical time series occur over random or non-fixed frequencies. Random time series exhibit none of the characteristics of trending, cyclical, or seasonal time series, and thus exhibit random value changes over non-fixed time frequencies [124]. Time series can also be either univariate or multivariate, the former meaning that only a single observation is recorded, and the later meaning that multiple observations are recorded.

Before analysis of time series data can be carried out by AI or ML algorithms, feature extraction or selection must be performed. In feature selection, data is processed such that characteristics of the original data are projected, while in feature extraction the original data is processed such that new information or representations of the data are produced [125]. For time series data, the simplest methods of feature selection is to calculate minimum, maximum, and other basic statistical functions. There result of each of these functions can then be considered a feature of the given time series. It is often the case, however, that time series data must first be transformed onto a new feature plane. This is because, in many cases, raw data does not contain predictable values within a known range, meaning basic feature selection may not produce a reliable representation of time series features for AI and ML algorithms. It is therefore common to use transformation functions on the time series before further processing and analysis of its data can take place. Since data is projected onto a new feature space as a result, this is considered feature extraction. Popular time series transformation functions include Fast Fourier Transform, which represents time series or signal data in the frequency domain [126], and Dynamic Time Warping, which computes the distance of two time series of different lengths [127]. It is always important to consider which information may be lost from the raw data when applying feature extraction [128].

Following feature selection or feature extraction, AI and ML algorithms can be used to identify the intended features. Basic conditional checks, or traditional ML algorithms such as Trees or KNN, can make use of simple mathematical features gathered during a feature selection step. However, an increasingly popular approach for processing and analysing time series data is to generate images and perform image classification. Commonly used examples of this are Gramian Angular Fields, which encodes temporal correlations between multivariate time series [129], and spectrograms, which encode frequency transformations such as Fast Fourier Transform [130]. As previously mentioned, the transformation functions used in the encoding process may sacrifice valuable information from the original data.

Another approach, which will be the focus of the image classification tasks carried out in this thesis, is to encode the time series data itself without using complex transformation functions, although basic transformations may be used to remove noise and standardise value ranges across samples. This approach can be used when time series data is random but still shares similar characteristics with other samples in the dataset [1]. While the type of time series is not a majorly impacting factor when using this approach, it is usually limited to domain-specific applications with unique or niche processes from which the data is recorded [1,22,131,132]. This approach provides a computationally efficient way to quickly transform time series data and produce image datasets that can then be used alongside ML and DL to create effective image classification models, which is the approach chosen for those created in Chapters 4 and 5.

3.3 Modelling

Using approach to processing data chosen in Section 3.2, other technologies can be used to create models capable of providing an analysis of this data. In this section, the technologies used to create models for analysing Port Talbot HSM data within this project are discussed. The methods used to optimise and validate these models are also described.

3.3.1 Transfer Learning

3.3.1.1 Transfer Learning using Pre-Trained Artificial Neural Networks

In traditional machine and DL, models are trained from scratch. This can take a long time when large datasets are used. It is however possible to use a pre-trained network for a chosen classification problem given that this network has been trained on relevant data. This is known as Transfer Learning and it is used when a pre-trained ANN, typically a CNN can boost model training times and accuracy when trained using new data due to the knowledge gained from the data it was originally trained on. For example, if a model is to be trained on a new dataset to classify images of vegetables, it may be possible to use a network pre-trained with images of fruit as it may extract features similarly between these two datasets. The pre-trained component of a network is usually used for more fundamental feature extraction such as edge and shape detection. “The objective of Transfer Learning is to take advantage of data from the first setting to extract information that may be useful when learning or even when directly making predictions in the second setting” [89]. This process is illustrated in Figure 3.3.

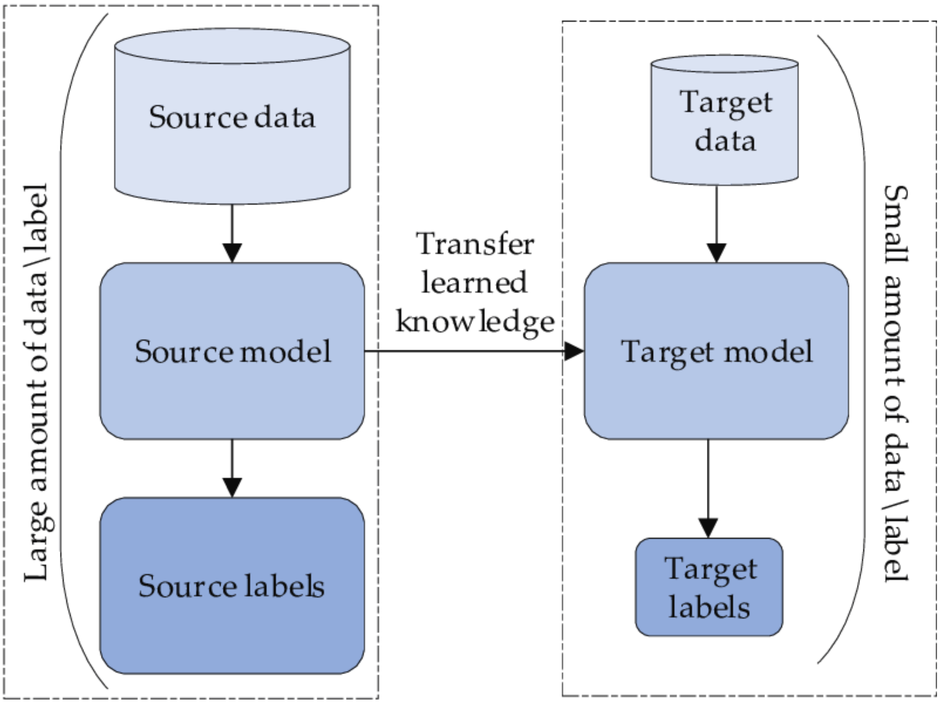

Large amount of data\label
Source data
Source model
Source labels
Transfer learned knowledge
Target data
Target model
Target labels
Small amount of data\label

Figure 3.3: The concept of transfer learning [5].

To achieve this, a pre-trained network is chosen and, unless already specifically trained for the new problem, is amended with new layers for further feature extraction more specific to the new data including output layers to suit the ML problem. This means that only the new layers must be trained using the new data, saving massively on training times and computational power since it is no longer necessary to train the network from scratch [58]. While Transfer Learning can produce CNN-based models which perform generally well with a fraction of the data, time, and computational power necessary to train one from scratch [133], it is still important to consider the potential drawbacks of such a powerful technology. Layers, nodes, and weights of pre-trained networks, for example, can sometimes remain unchanged resulting in a model not generalising data well during re-training [134]. This, however, is typically subject to which layers are altered and re-trained, as well as the veracity of the new training dataset. For both small and large datasets, unclear distinctions between samples can lead to this poor generalisation when re-training and, thus, an unreliable model. Transfer Learning is the approach used to train the image classification models discussed in Chapters 4 and 5, and to address the challenges that also arise from the collection and processing of real-world industrial data [1, 22, 23].

Transfer Learning is becoming increasingly adopted in a number of manufacturing domains including, but not limited to, anomaly detection, time series classification, Computer

Vision, and fault diagnosis due to its ability to assist in training where limited domain-specific data is available. Simultaneously, real-world examples of these applications are scarce due to either their limited scope of being demonstrated on a small, simple tasks, or due to issues related to the sharing and security of private datasets [135].

3.3.1.2 Pre-Trained Convolutional Neural Network Architectures

There are a large number of pre-trained CNN architectures available for use in Transfer Learning, each varying in depth and breadth, and with their own advantages and disadvantages. Selecting a pre-trained CNN architecture is dependent upon both the domain and complexity of the use case. Popular architectures include AlexNet, ResNet, and GoogLeNet [90].

AlexNet, illustrated in Figure 3.4, has a depth of 8 layers, which is relatively short compared to other pre-trained networks. It contains five convolutional layers and ends with three fully connected layers. Only the first, second, and last convolutional layers are followed by a ReLU activation function. AlexNet was the first CNN architecture to expand the number of layer in a CNN from a single convolutional and fully connected layers to increase depth. It was also the first CNN architecture to include the ReLU function [6] which, as previously described, eliminates negative values in the network's weights and neurons. AlexNet is trained on ImageNet [136], an ever-expanding dataset which currently contains over 14 million images and over 1000 classes. AlexNet was also the winner of the ImageNet Large Scale Visualization Recognition Challenge (ILSVRC) [137] 2012, a competition in new CNN architectures attempt to classify an updated version of the ImageNet dataset with the highest accuracy and lowest error. The disadvantage of this network, however, is that it still has a limited depth compared to other architectures, meaning that it may take longer to train or learn features.

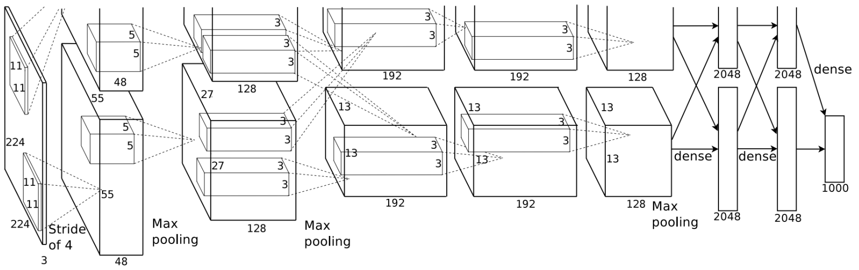


Figure 3.4: AlexNet's Architecture [6].

Residual Network, more commonly referred to as ResNet, has several variations including ResNet18, Resnet50, and ResNet101. The number in each variation describes how many layers deep the network is. It contains a series of convolutional layers which gradually use an increased number of kernels. The unique feature of ResNet, however, is that it includes a skip function, or Residual Block, which allows the network to bypass a remaining convolutional layers based on the current weight update at a given point in the network, addressing the problem of irrelevant updates and overfitting [7, 138]. However, a deeper variation of this network may take longer to train, and a shorter variation may reduce its ability to learn features. These factors should be taken into account if using a variation of ResNet. The ResNet architecture and an example of its Residual Block are illustrated in Figures 3.5 and 3.6, respectively.

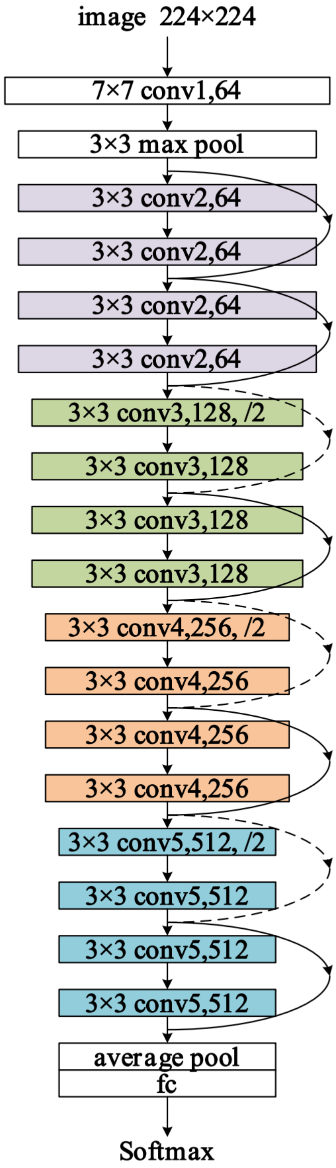


Figure 3.5: ResNet18's Architecture [7].

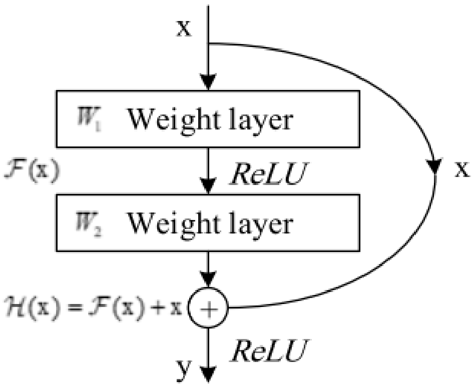


Figure 3.6: ResNet's Residual Block [7].

One of the most commonly used pre-trained CNN architectures is GoogLeNet. GoogLeNet is chosen as the pre-trained CNN architecture used in the image classification tasks in this thesis due to its ability to maintain a low computational expense when training regardless of its width and depth [8]. More specifically it uses smaller convolutional filters to reduce the total parameters within the network, allowing for a larger architecture. An excessive number of neurons and layers however can lead to extreme underfitting or overfitting. GoogLeNet's Inception module balances this through its Inception module which contains sequential, fixed sized convolutional filters that enable it to generalize features more easily and avoid overfitting. GoogLeNet was also the winner of the ILSVRC 2014. An example of GoogLeNet's Inception module is illustrated in Figure 3.7 while the full architecture is shown in Figure 3.8.

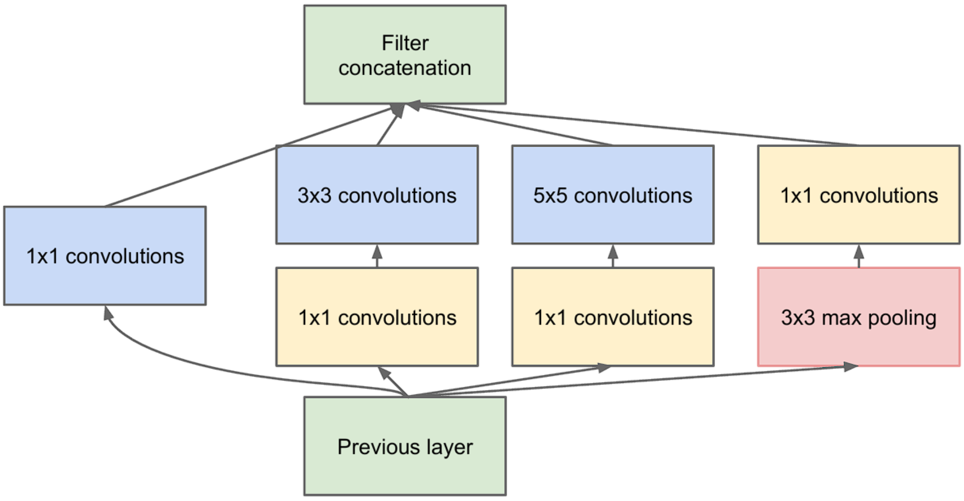


Figure 3.7: GoogLeNet's Inception Module [8].

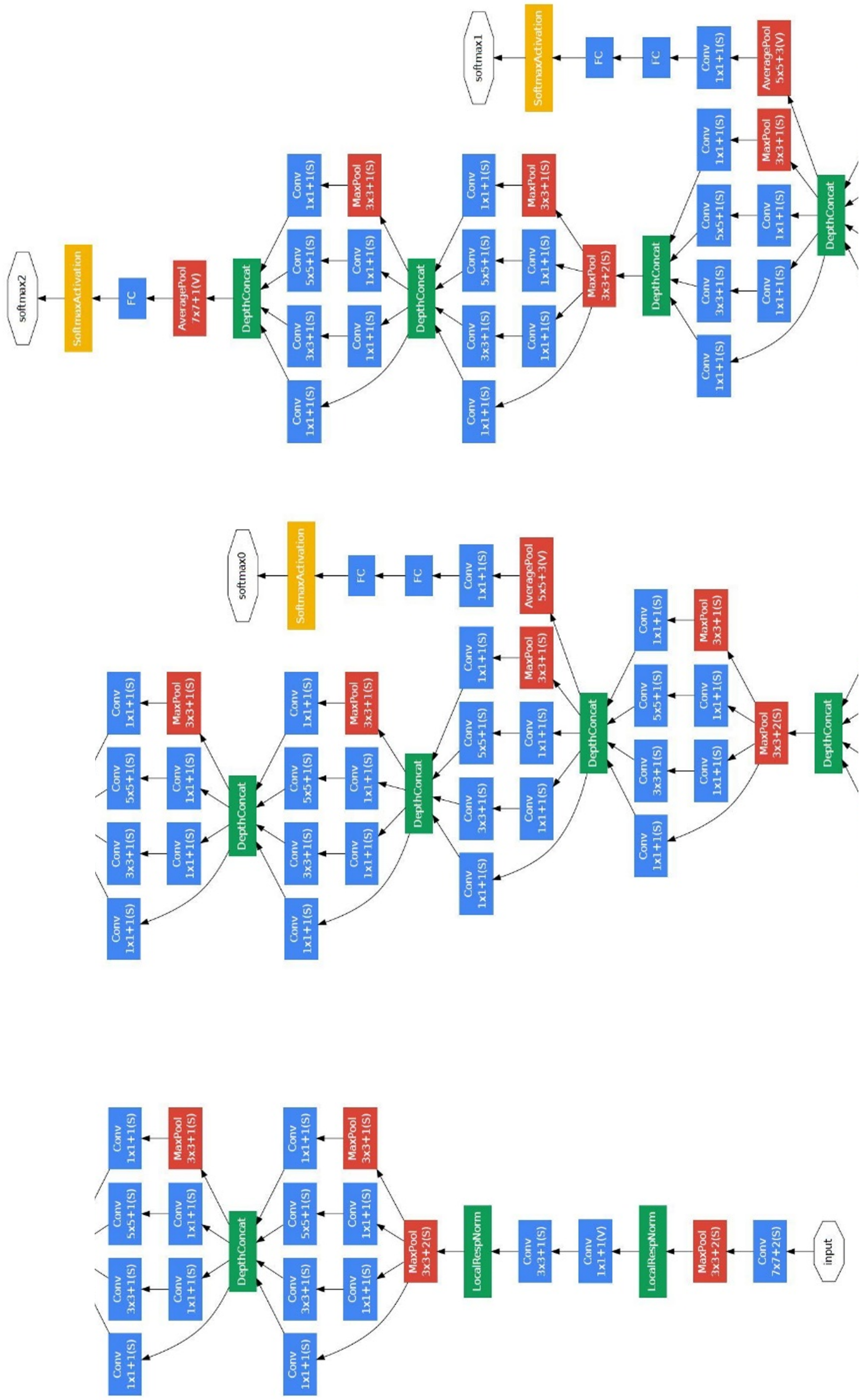


Figure 3.8: GoogLeNet's Architecture [8].

The final two layers used for classification in the standard GoogLeNet architecture have been replaced by a custom fully-connected layer and are followed by a classification output layer to enable further training using images specific to a given classification task.

3.3.2 Hyperparameter Optimisation

Hyperparameter optimisation aims to determine the optimal hyperparameters for training a neural network in order to produce the best performing model. Hyperparameters are sometimes referred to as training options since they are a configurable setting. There are a number of hyperparameters which can be configured, and some have a larger impact on performance than others.

The number of epochs, or Max Epochs, is the number of iterations over the entire dataset. This determines the number of times all samples are observed and, therefore, contribute towards the decisions made by the final model [139]. A lower number of epochs, or fewer iterations, gives a network less opportunity to observe training samples and to construct a reliable model, usually resulting in poor training and testing performance. A larger number of epochs, or more iterations, allows an algorithm to observe training samples multiple times which helps it to confirm its association of features to particular labels. If the number of epochs is too large, however, the algorithm may lose its ability to generalise by overfitting to the training data, resulting in normal, or good, training performance but poor testing performance. An optimal Max Epochs value would achieve good training and testing performance without overfitting [139, 140].

Mini Batch Size is another configurable hyperparameter of neural networks. It is the number of samples observed by the network during training before its weights are updated [141]. A larger Mini Batch Size will allow the network to observe a larger amount of data at once which has been argued to deteriorate generalisation. A smaller Mini Batch Size means that weights are updated more frequently while observing less data. If the Mini Batch Size is too small, however, generalisation can still deteriorate due to conflicting observations in each batch. An optimal Mini Batch Size depends on the size of the training dataset and the complexity of the data. The most thorough way to determine the best Mini Batch Size for a neural network is, however, a hyperparameter optimisation algorithm.

Initial Learn Rate is a factor by which weights are multiplied when they are updated. This determines how drastically the value of the weights changes and, therefore, how quickly or slowly the network learns [141, 142]. A lower Initial Learn Rate can allow the network more time to learn which can help when datasets are large, and require many iterations and more batches although this can lead to longer training times. A larger Mini Batch Size can shorten training times, but can also cause the network to update its weights too quickly and potentially decresase performance.

Other configurable hyperparameters include Learn Rate Drop Factor, a factor by which the learn rate is gradually decreased, and L2Regularization, a form of weight decay which prevents the network from overfitting. These, and other, configurable hyperparameters, however, do not have as significant an impact on performance as Max Epochs, Mini Batch Size, and Initial Learn Rate in this project. This was demonstrated in a GSO experiment prior to creation of the ANC stroke timing classification task in Chapter 5 [22].

Optimal hyperparameter values can vary between different models and classification tasks. When different models with similar or identical architectures are used for different classification tasks, however, optimal values may not differ by as much, particularly as models become more reliable with more quality training data. The effects and changes between each hyperparameter and their values are reviewed for each of the optimised classification models created in Chapters 4 and 5.

3.3.2.1 Grid Search Optimisation

GSO is an exhaustive method of hyperparameter optimisation in which a number of values to be tested for each hyperparameter are used to train a network for every possible combination [143, 144]. Table 2 displays an illustration of GSO in which O_n^m describes a hyperparameter value where O_n is a given hyperparameter and m is a chosen value. The final set of hyperparameter value combinations is constructed from the Cartesian product of this table. GSO is beneficial for providing an overview of performance across a wide range of training options. This, however, comes at the cost of time and computational power.

Table 2: Theoretical selection of hyperparameter values, or training options, for use in GSO.

Hyperparameter	O_1	O_2	...	O_n
Value	O_1^1	O_2^1	...	O_n^1
	O_1^2	O_2^1	...	O_n^2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$

3.3.2.2 Bayesian Optimisation

Another hyperparameter optimisation method is Bayesian Optimisation. This is a probabilistic approach in which the effect of hyperparameter values on model performance are evaluated. The result of this evaluation is used to train further models which are guided by error and probability to gradually improve performance [144,145]. Limits can also be set for hyperparameter values and for the error for each training run before running the algorithm. Bayesian Optimisation can dramatically decrease the time taken to find hyperparameter values that perform well with its calculated approach, however, it is less exhaustive than the GSO algorithm, meaning it is possible that the resulting hyperparameter values may not be the most optimal [146].

3.3.3 Cross Validation

Two datasets, training and testing, are typically used when creating a neural network [147]. The test dataset is used after training to determine the performance of a model when analysing new, unseen data. Cross-validation is commonly used as a training and testing methodology, and varying splits of training and testing data can be used to create optimal models and determine how well they classify new, unseen data [147]. There are a number of Cross Validation methods available, each with advantages and disadvantages. Hold Out Cross Validation (HOCV) and K-Fold Cross Validation (K-FCV) are two of the most commonly used Cross Validation methods.

3.3.3.1 Hold Out Cross Validation

HOCV randomly splits samples in a dataset into two subsets for training and testing. These are usually 70% and 30%, respectively, however, splits of 80% and 20%, respectively, are also commonly used [148]. It should be noted that this split applies across labels such that 70% of the total samples for each label in a classification task are selected for the training dataset. This method is typically performed once, but can be run multiple times with different randomised splits to confirm repeatability. HOCV has a lower computational expense than other Cross Validation methods due to its lower number of training and testing runs, particularly when combined with a hyperparameter optimisation algorithm. However, it is difficult to determine the reliability of the training dataset with a single, or only a few, training and testing runs when compared to other Cross Validation methods, namely K-FCV [149].

3.3.3.2 K-Fold Cross Validation

K-FCV, illustrated in Figure 3.9, splits the entire dataset into a specified number, K, of folds [150, 151]. A commonly used value for K is 5, which means that the dataset is split into five folds of 20%. In this case, five training and testing runs are then performed using each possible combination of four folds, or 80% of the total data, as the training dataset and the remaining fold, or 20%, as the testing dataset. When K is 5, this means that a total of five training and testing runs are performed. The performance of these runs is then averaged and used to evaluate the overall performance of the model. While this method is more time-consuming than HOCV, it is more widely used as it provides an insight into how well the model generalises when given the opportunity to train of all possible samples in the dataset. K-FCV should also only be used to evaluate the model during development, unless a sufficient amount of data becomes available after deployment to re-train and thus update it [149].

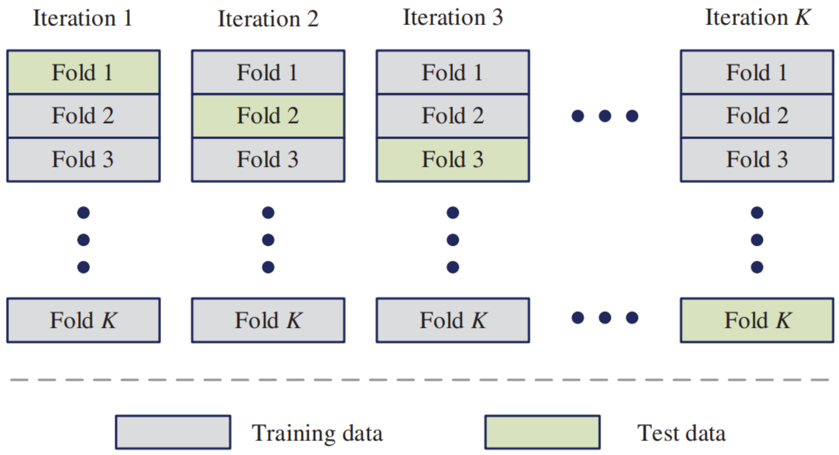


Figure 3.9: K-Fold Cross Validation [9].

3.3.3.3 Evaluation Metrics

To evaluate the performance of classification models throughout this thesis, several metrics are used. These include accuracy, precision, recall, and F1 Score. Accuracy simply describes the percentage of samples in the test dataset that have been assigned the correct label and have therefore been classified correctly. Accuracy is a useful metric which provides a high-level description of a model's performance. It is calculated using the following formula 1.

$$Accuracy = \frac{TruePositives}{TotalNumberofSamples} \tag{1}$$

Accuracy, however, is not reliable on its own as it does not provide any information with regards to false positive and false negative rates. Precision, recall, and F1 Score are therefore calculated and used to provide further insight into model performance alongside accuracy. Precision describes the number of samples assigned to a class which truly belong while recall describes the percentage of samples belonging to one class which are classified correctly. These metrics are calculated for each class and can be reviewed independently but are typically averaged to provide a final score. F1 Score provides further insight into model performance as it takes both precision and recall into account. It calculates the performance of a model based on true positives while factoring in false positives and false negatives [152]. Precision, recall, and F1 Score are calculated using formulas 2, 3, and 4, respectively.

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)} \tag{2}$$

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)} \tag{3}$$

$$F1Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{4}$$

These four metrics can be used in conjunction to determine whether a model performs suitably for a classification task with regards to both the entire dataset and each class.

3.4 Results Analysis and Visualisation

The feedback of the models created using the technologies described in Section 3.3 can be integrated into an analytics platform for further review. In this section, the technologies and software used to create a platform for the conveyance and visualisation of the results produced by the previously described models and a further decision-making process, described in Chapter 6, are discussed.

3.4.1 MATLAB and Technologies for a Local Web Environment

To implement the final component of the BDA framework described in Section 2.1.2, it is necessary to combine the models created in the previous component with a platform capable of visualising both data and model outputs, and to enable further AI to be implemented for basic conditional checks on simple types. This is approached by implementing a local web environment to load data in such a manner that it simulates it being collected and visualised as if a steel strip were progressing through the HSM process [1]. The technologies used to achieve this are MATLAB, XAMPP, and the mark-up and programming languages HTML, PHP, and JavaScript.

MATLAB [153] is a powerful programming tool which allows users to easily keep track of workspace data as it is being processed. It also includes a large number of packages which provide advanced data processing and AI capabilities. MATLAB is used for the pre-processing of data, and for the creation and evaluation ML models [1,22,23] discussed in the previous framework steps, but it is also used in this final step of the framework to generate an executable file which can be triggered by the tool when necessary to utilise the model and retrieve their output for new data [1].

HTML, PHP, and JavaScript are the languages used to create web-based applications in the Port Talbot HSM. By using these to create a local web environment, integration into the current Port Talbot HSM web infrastructure would be a simple process with minimal additions to the source code, should the tool be deployed in the future. XAMPP [154] is an open-source web server solution which enables users to create local, offline environments in which to run web applications. This is used to test the final web tool locally while navigating the limitations of data access permissions and deployment delays [1]. Examples of applications include the estimation of roller force using ANNs and several traditional ML algorithms [155], predicting the remaining useful life of rollers using ANNs [156], and the recognition of edge defects using CNNs [157].

3.5 Summary and Conclusions

In this chapter, a structured approach has been considered and is used to carry out the practical experiments in the following chapters. Firstly, an overview of data sources in the HSM and their access was given in Section 3.1, providing an understanding of where process data is currently stored, how analysts access and review it, and how it could be extracted for use in the practical experiments. Classification for time series images has also been selected as an approach to analysing ambiguous time series data, namely due to its random nature, as described in Section 3.2. Further to this, Transfer Learning has been selected as an approach to training CNN image classification models to address the challenge of limited data availability while providing other possible benefits including shorter training times and better generalisation, and several pre-trained CNN architectures were also compared in great detail. Techniques for hyperparameter optimisation and Cross Validation were then discussed and compared. Specific approaches for these, as well as a specific pre-trained CNN architecture, are selected in the following chapters prior to the practical experiments. Finally, MATLAB, XAMPP, and several programming languages have also been considered as a means of creating a local web environment, based on the Port Talbot HSM web infrastructure, in which the tool created in Chapter 6 is run.

In Chapter 4, width defects found in the HSM are described in detail and the current process used by analysts and operators to identify and analyse them is reviewed. This knowledge is used to derive a set of rules to be integrated into a new, data-driven decision-making process, described in Chapter 6, and to develop ML models from a local data structure in MATLAB. The specific process for collecting HSM data and establishing this local data structure is also described in Chapter 4, prior to the development of these models.

4 Detecting Width-Related Defects in the Hot Strip Mill Process using Deep Learning and Expert Knowledge

The experiments in this and the following chapters are carried out by adhering to the BDA framework described in Section 2.1.2 and the specific methodology to follow this framework described in Chapter 3. The purpose of this chapter is to understand the width-related defects that can occur throughout the HSM process, and to create and evaluate ML models for those defects which cannot be identified using simple tolerances and thus conditional checks. In Section 4.1, an overview of all potential width-related defects in the HSM process, how they can occur, and how they are currently resolved is provided. In Section 4.2, the process used for Data Collection directly from the HSM data sources described in Section 3.1 is outlined. In Sections 4.3.1, 4.4.1, and 4.5.1, the Data Management stage is described, including how data is processed and transformed in MATLAB in order to produce the required datasets, in this case, image datasets. In Sections 4.3.2, 4.4.2, and 4.5.2, the Modelling and Results and Analysis stages are carried out using selected hyperparameter optimisation and Cross Validation techniques. The findings and results of this chapter form part of the published journal article titled 'A Tool to Combine Expert Knowledge and Machine Learning for Defect Detection and Root Cause Analysis in a Hot Strip Mill' [1], mentioned in Section 1.4.1.

4.1 Width-Related Defects in the Hot Strip Mill Process

There are several possible width-related defects that can occur throughout the HSM process. They are first detected within the Roughing Mill, Finishing Mill, or Coiler subprocess, and their root causes may originate in the same or previous subprocesses. An overview of width-related defects and the subprocess they originate from is shown in Figure 4.1. Operators and analysts use both rules and expert knowledge based on whether the data is ambiguous to confirm the exact nature of the defect. This procedure is described below and is used to derive the decision-making process of the final web tool described in Chapter 6.

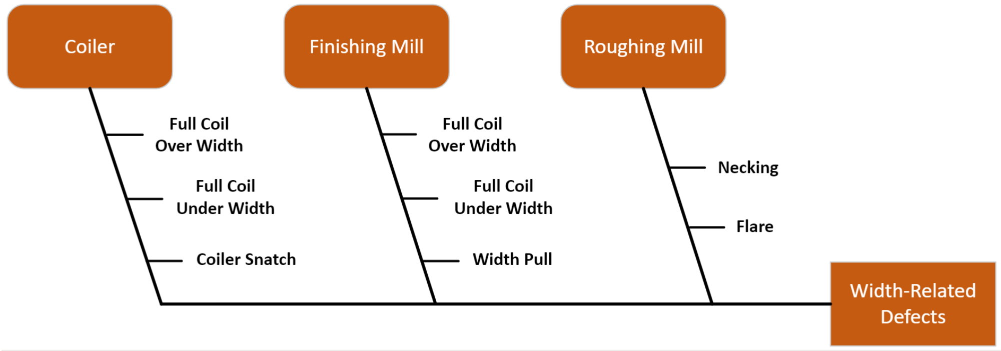


Figure 4.1: Fish bone diagram showing possible width-related defects in the HSM [1].

4.1.1 Roughing Mill Subprocess

The Roughing Mill, illustrated in Figure 4.2, is the last point of direct width control in the HSM process, using vertical edge rollers to guide a bar towards a required width, and is the first subprocess in which a width-related defect can be identified. There are two possible width-related defects that can occur in the Roughing Mill which can also affect width performance of the bar in subsequent subprocesses: Necking and Flare.

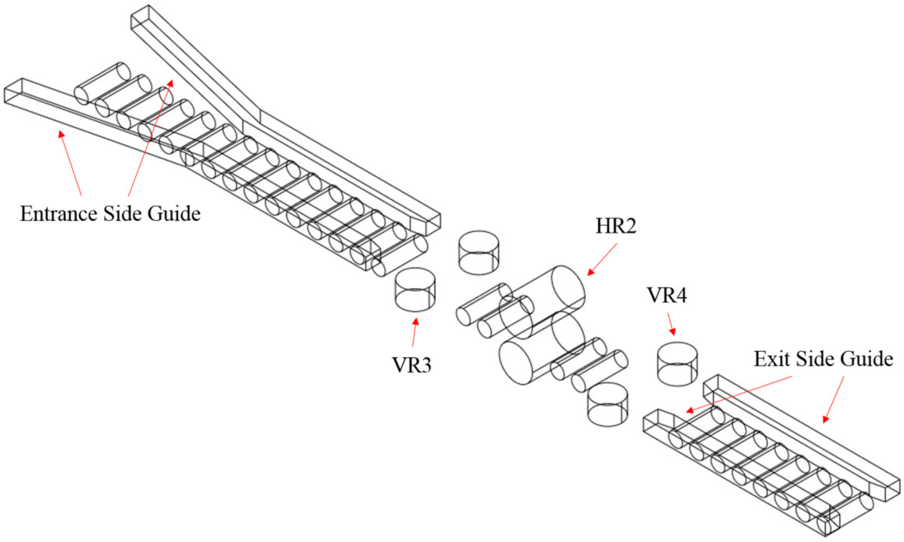


Figure 4.2: Schematic modelling of roughing rolls, edging rolls, and side guides [10].

Necking occurs when the head or tail end of the bar, or both, is narrower than the main body [22, 158]. Analogously, Flare occurs when the head or tail end of the bar, or both, is wider than the main body [22, 158]. Both can be identified through visual inspection of the Roughing Mill width deviation times series. While factors that may contribute towards these defects may be identified during Scheduling, and thus before entering the HSM process, it is currently only possible to determine whether these defects are truly present after they occur. This also applies to defects in subsequent subprocesses. Operators currently address identified instances of Necking and Flare by checking for potential root causes including early ANC stroke timing and high or low Roughing Mill model error. Examples of width deviation signals showing behaviour of Necking and Flare are shown in figures 4.3 and 4.4, respectively. Within these figures, the x-axis represents time while the y-axis represents width deviation from the width aim.

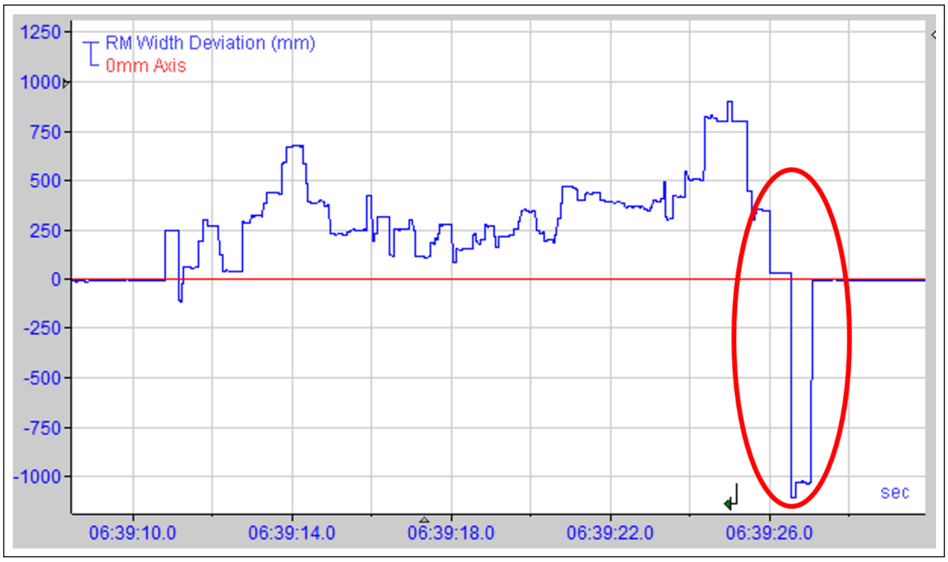


Figure 4.3: Width deviation signal of a bar with Necking at the tail end in the Roughing Mill.

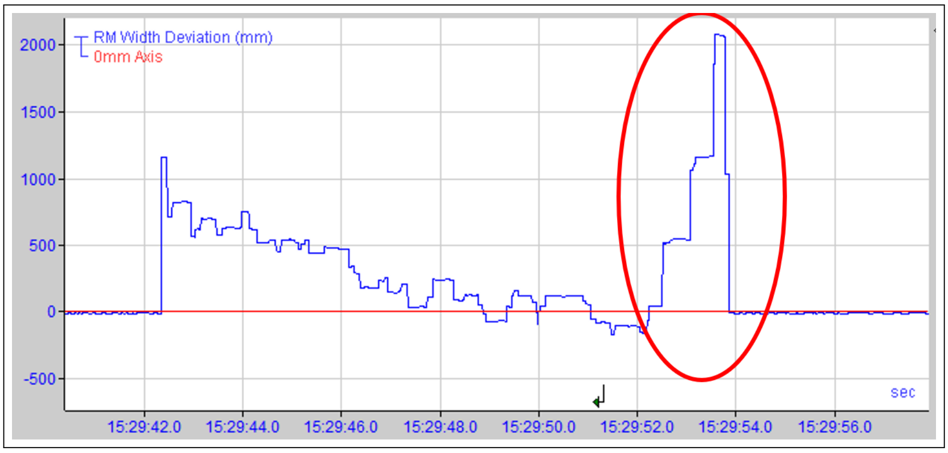


Figure 4.4: Width deviation signal of a bar with Flare at the head end in the Roughing Mill.

4.1.2 Finishing Mill Subprocess

In the Finishing Mill, the bar, now considered a strip due to its elongation and thinning in this subprocess, is rolled to a required thickness through a series of stands containing horizontal rollers. Figure 4.5 provides an illustration of a Finishing Mill stand. Positive and negative tolerances exist to determine whether a strip is performing optimally with regards to its width. There are several width-related defects which can occur in this subprocess: Full Coil Under Width, Full Coil Over Width, Width Pull, and Flare.

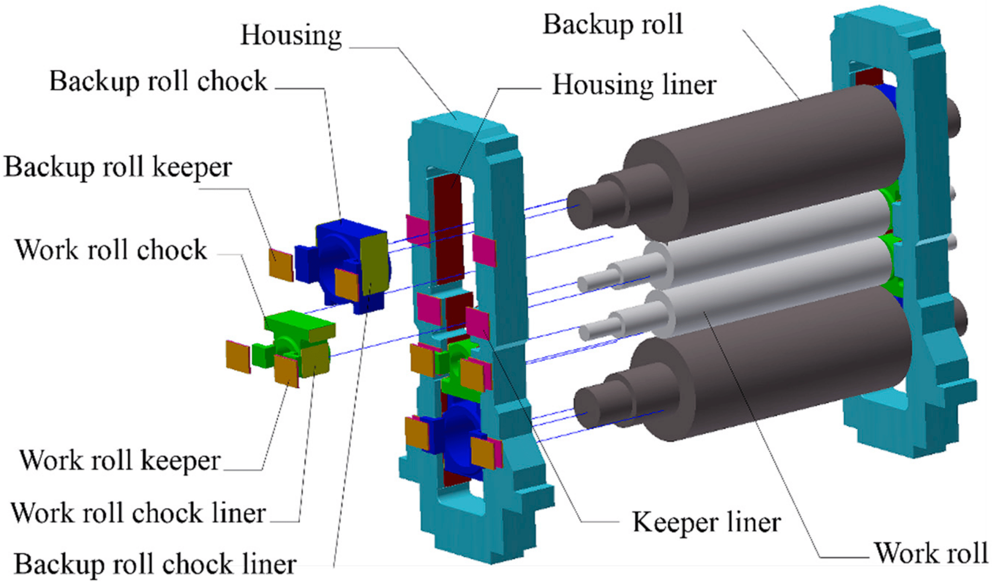


Figure 4.5: View of a finishing stand [11].

A strip is categorised as Full Coil Under Width if 10% or more of its width is less than the negative tolerance, and categorised as Full Coil Over Width if 50% or more of its width is greater than the positive tolerance. The existing tolerances can be used to determine whether Full Coil Under Width and Full Coil Over Width are present in the Finishing Mill subprocess. Generally, defects in which width exceeds positive tolerances are not considered as severe as those which fall below negative tolerances or deviate to lower width measurements. This is because post-processing can be carried out to remove excess width as opposed to lost width, which cannot be recovered. Width Pull occurs when the width of the head end of a strip, or both, suddenly deviate to lower measurements from the main body [159, 160]. While an elongated Width Pull can occasionally result in Full Coil Under Width, it is often the case that width measurements in a Width Pull either fall below the negative tolerance only for a very small percentage of the strip's length or not at all. Each of these defects can be identified through visual inspection of the Finishing Mill width deviation signal. Operators currently address these issues by identifying potential root causes including high or low Finishing Mill model error, high or low strip temperatures, and those in from previous subprocesses. Examples of width deviation signals showing behaviour of Full Coil Under Width, Full Coil Over Width, and Width Pull are shown in figures 4.6, 4.7, and 4.8, respectively. Within these figures, the x-axis represents time while the y-axis represents width deviation from the width aim.

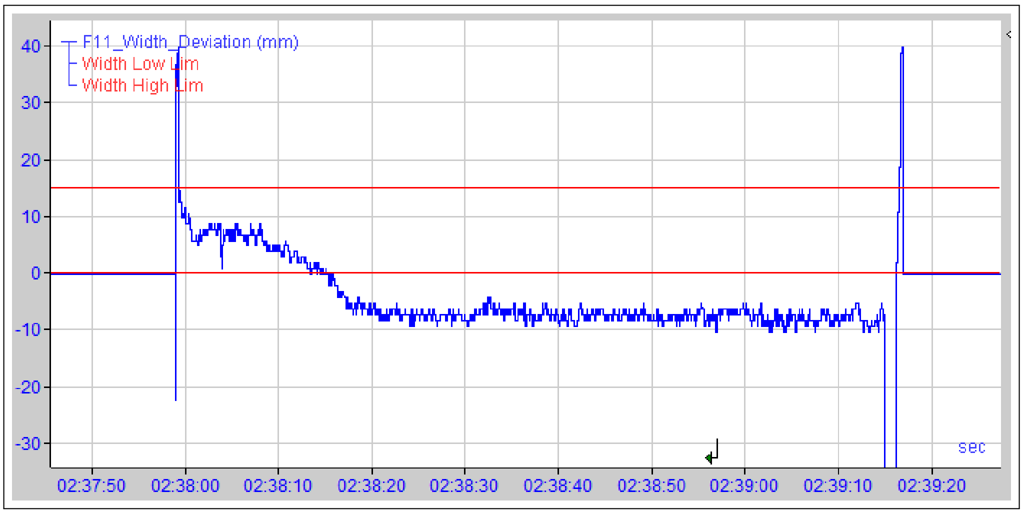


Figure 4.6: Width deviation signal of a strip with Full Coil Under Width in the Finishing Mill.

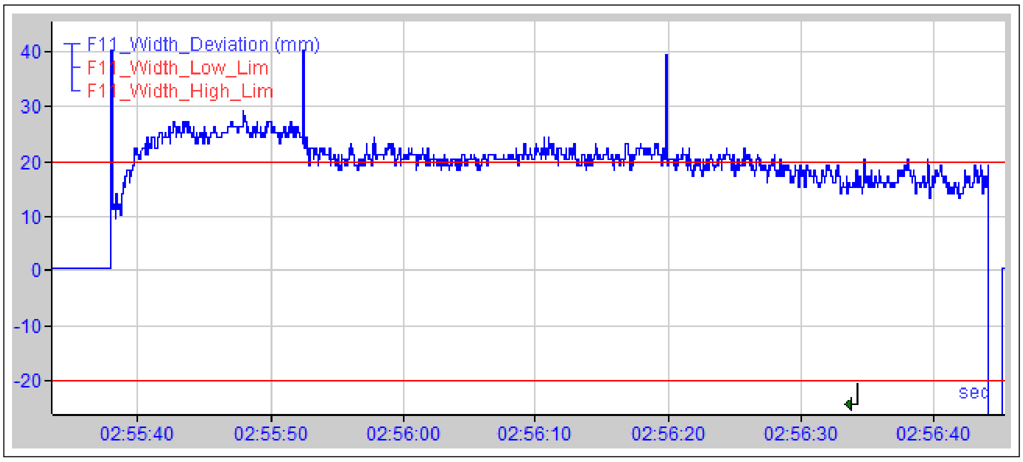


Figure 4.7: Width deviation signal of a strip with Full Coil Over Width in the Finishing Mill.

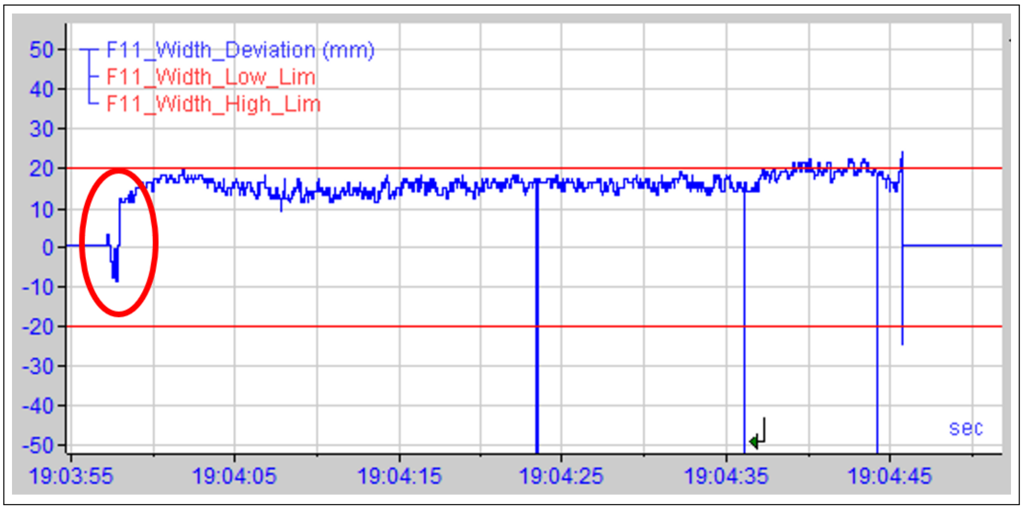


Figure 4.8: Width deviation signal of a strip with Width Pull in the Finishing Mill.

4.1.3 Coiler Subprocess

In the Coiler subprocess, illustrated in Figure 4.9, the strip is wound into a coil before being stored or passed on to the Cold Rolling Mill process. Width-related defects which can occur in this subprocess are largely the same as those in the Finishing Mill. The same positive and negative tolerances from the Finishing Mill are used here, and the same conditions are used to determine whether Full Coil Under Width, Full Coil Over Width, or Flare are present. The last width-related defect which can occur in the Coiler subprocess is Coiler Snatch. Similarly to Width Pull in the Finishing Mill, Coiler Snatch is a sudden deviation to lower measurements in width. However, this can occur at several points throughout the strip and may not necessarily occur at its head. While Coiler Snatch may originate in the Coiler itself, it may become present as a result of Width Pull in the Finishing Mill, or as a result of other width defects from the Roughing Mill. Each of these defects can be identified through visual inspection of the Coiler width deviation time series. Operators currently address these issues by identifying potential root causes, including high or low strip temperatures, and those from previous subprocesses. Any issues found in the HSM process are relayed to the Scheduling team and analysed accordingly. Manual offsets are then made to the models in subsequent subprocess or strips if necessary. An example of a width deviation signal showing behaviour of Coil Snatch is shown in Figure 4.10. Within this figure, the x-axis represents time while the y-axis represents width deviation from the width aim. Table 3 provides a summary of these defects, how they are found and addressed, and their potential root causes.

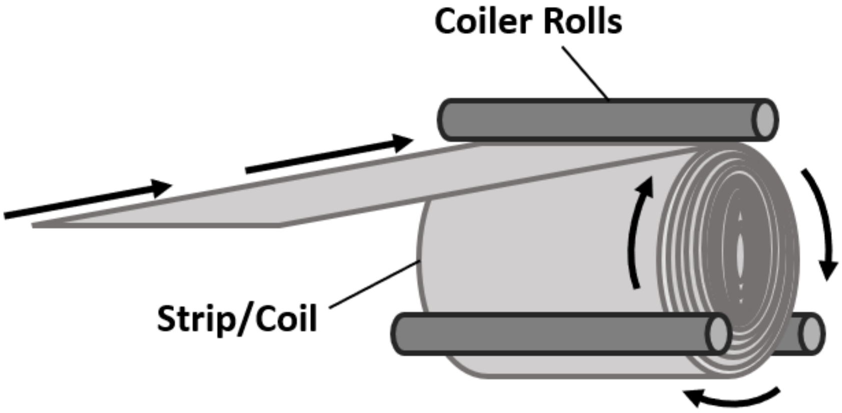


Figure 4.9: Illustration of the Coiler subprocess.

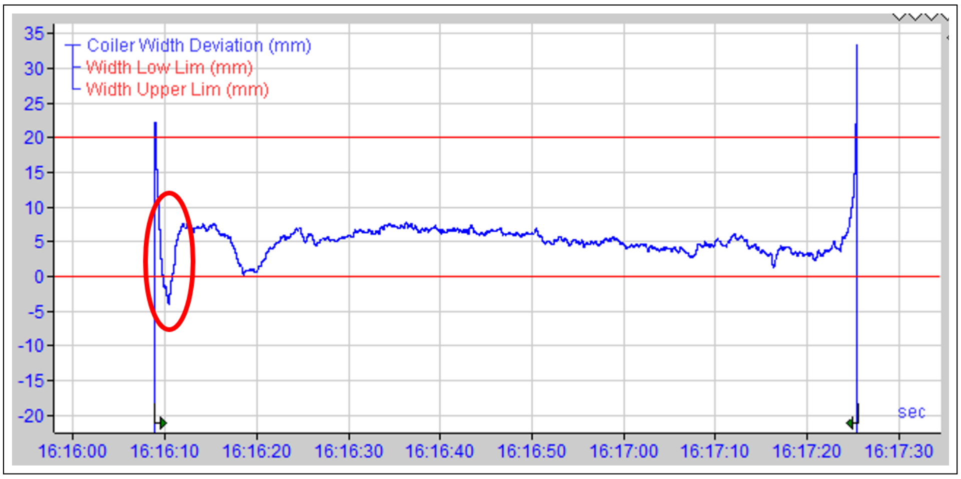


Figure 4.10: Width deviation signal of a strip with Coiler Snatch.

Table 3: Summary of possible width-related defects that can be found in the HSM process.

Defect	**Description**	**How it's currently found and addressed**	**Potential Root Causes**
		Roughing Mill	
Necking	Head or tail end of bar becomes narrower than main body.	Visual inspection of Roughing Mill width deviation time series data. It should be noted that, in this thesis, the terms	Problematic Grade, Spread/Squeeze rules not followed, Delay, Single Furnace Operation,

Flare	Head or tail end of bar becomes wider than main body.	time series and signal are used interchangeably. ANC stroke timing and Roughing Mill model error, as well as potential root causes from Scheduling and Furnaces, are checked and any issues found are relayed to the scheduling team and analysed accordingly.	ANC Stroke Early/Disabled, High/Low Roughing Mill Model Error
		Finishing Mill	
Full Coil Under Width	10% or more of strip's width is below lower tolerance.	Visual inspection of Finishing Mill width deviation time series data. Finishing Mill Model Error and temperature, as well as potential root causes from previous subprocesses, are checked and any issues found are relayed to the scheduling team and analysed accordingly.	High/Low Finishing Mill Model Error, High/Low Temperature, potential root causes from the previous subprocesses.
Full Coil Over Width	50% or more of the strip's width is above upper tolerance.		
Width Pull	Width of strip's head or tail end, or both, deviates to lower measurements than the main body.		
		Coiler	
Full Coil Under Width	10% or more of strip's width is below lower tolerance.	Visual inspection of Coiler width deviation time series data. Temperature, as well as potential root causes from previous subprocesses, are checked and any	High/Low Temperature, potential root causes from the previous subprocesses.
Full Coil Over Width	50% or more of the strip's width is above upper tolerance.		

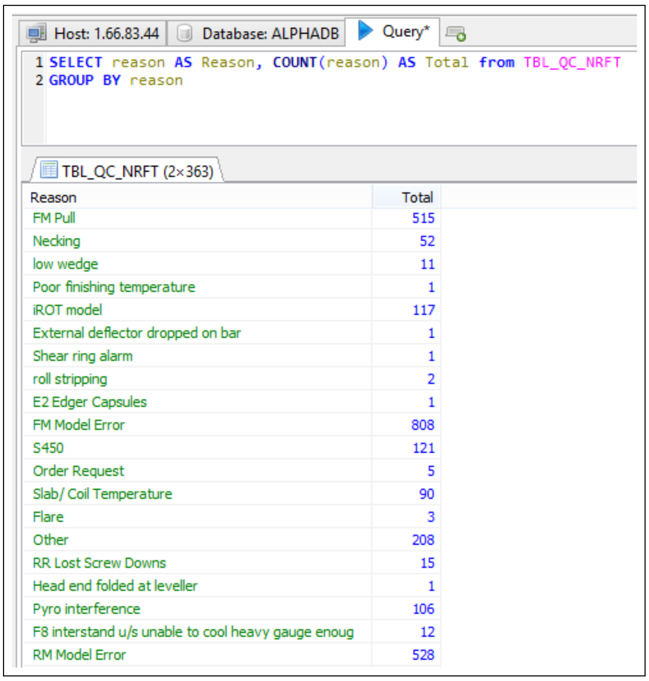
Coiler Snatch	Width of a strip suddenly deviates from the main body.	issues found are relayed to the scheduling team and analysed accordingly.

In this chapter, DL models to classify images which represent width deviation time series data are created using the methods and techniques described in Chapter 3. Through this, the ability to determine whether width-related defects which cannot be reviewed using conditional checks and tolerances are present in the HSM process is enabled, and a foundation is created upon which further classifications and checks for root causes of width-related defects in Chapter 5 are introduced. The three individual models created in this chapter determine the presence of Necking and Flare in the Roughing Mill subprocess, Width Pull in the Finishing Mill subprocess, and Coiler Snatch in the Coiler subprocess. The methodology for these models specifically is derived from the current process, described above, of visually inspecting time series data, hence image classification [1]. The rules, and other expert knowledge used to determine whether other defects are present, are combined with these models and integrated into the tool created in Chapter 6.

4.2 Methodology Following Big Data Analytics Framework

4.2.1 Data Collection and Management

As described in Chapter 3, a conventional BDA methodology [2] is followed for each of the experiments carried out in this thesis. The first step towards creating the DL models for defect detection, as well as the models created in Chapter 5, was extracting data from the data sources described in Section 3.1 and compiling this collection into a MATLAB workspace, the chosen tool for data pre-processing and modelling. To achieve this, the necessary static data found in the EngServer2 database was first extracted. Defects are labelled in the 'reason' column of the 'TBL_QC_NRFT' table, described in Table 4. A complete list of recorded defects was queried, and the defect names which matched or reflected those width-related defects focused on in this thesis were reviewed with analysts and used as a filter in the subsequent query to gather the relevant, labelled data. The possible root cause of the identified defect is also recorded in the 'root_cause' column of the 'TBL_QC_NRFT' and sometimes described in the 'comment' column. It should be noted that these columns are sometimes left blank either if the cause of a defect was found to be obvious from time series data, or if the root cause was not identified at all.

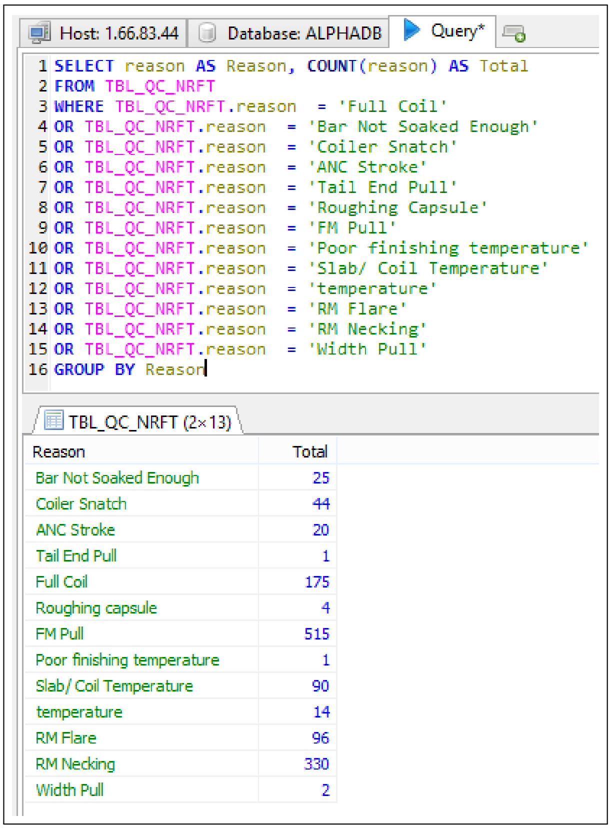


```sql
SELECT reason AS Reason, COUNT(reason) AS Total from TBL_QC_NRFT
GROUP BY reason
```

Reason	Total
FM Pull	515
Necking	52
low wedge	11
Poor finishing temperature	1
iROT model	117
External deflector dropped on bar	1
Shear ring alarm	1
roll stripping	2
E2 Edger Capsules	1
FM Model Error	808
S450	121
Order Request	5
Slab/ Coil Temperature	90
Flare	3
Other	208
RR Lost Screw Downs	15
Head end folded at leveller	1
Pyro interference	106
F8 interstand u/s unable to cool heavy gauge enoug	12
RM Model Error	528

Figure 4.11: HeidiSQL query to retrieve the names and counts of all unique defects in the TBL_QC_NRFT table.

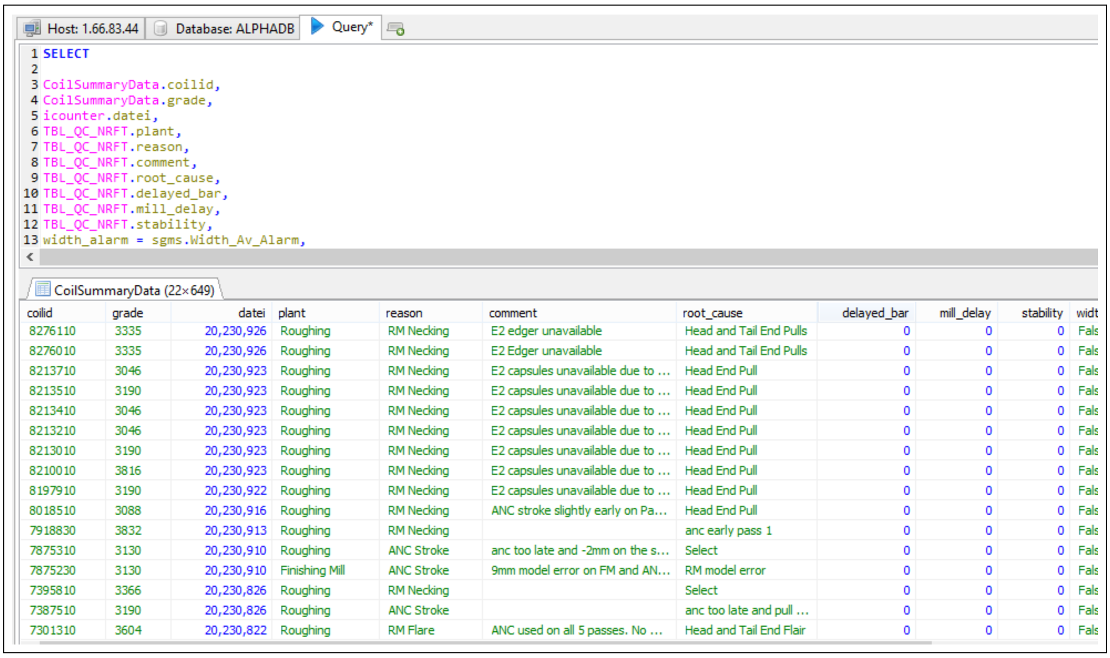
Derived from a list of all existing entries of defects in TBL_QC_NRFT, shown in Figure 4.11, a final list of the fields required from the EngServer2 database, shown in Table 4, was reviewed with analysts and a query was used to retrieve all possible records of these fields which also exist within TBL_QC_NRFT and have a 'reason' matching one of those shown in Figure 4.12. The resulting data table, shown in Figure 4.13, accounts for samples that are used to teach ML models defective behaviour. A similar number of instances that do not exist in the TBL_QC_NRFT table are queried to form those samples used to train models to learn normal behaviour. In this case, the 'reason' column and other fields taken from this table are allocated an 'Okay' value. For the remainder of this thesis, notably within experiments, the term 'Okay' is used to denote any samples with no defective behaviours. For example, in an experiment performed in Section 4.3, a model is created to distinguish between steel strips which show characteristics of Width Pull in the FM subprocess and others which show no abnormal characteristics and are therefore 'Okay'.



```sql
SELECT reason AS Reason, COUNT(reason) AS Total
FROM TBL_QC_NRFT
WHERE TBL_QC_NRFT.reason  = 'Full Coil'
OR TBL_QC_NRFT.reason  = 'Bar Not Soaked Enough'
OR TBL_QC_NRFT.reason  = 'Coiler Snatch'
OR TBL_QC_NRFT.reason  = 'ANC Stroke'
OR TBL_QC_NRFT.reason  = 'Tail End Pull'
OR TBL_QC_NRFT.reason  = 'Roughing Capsule'
OR TBL_QC_NRFT.reason  = 'FM Pull'
OR TBL_QC_NRFT.reason  = 'Poor finishing temperature'
OR TBL_QC_NRFT.reason  = 'Slab/ Coil Temperature'
OR TBL_QC_NRFT.reason  = 'temperature'
OR TBL_QC_NRFT.reason  = 'RM Flare'
OR TBL_QC_NRFT.reason  = 'RM Necking'
OR TBL_QC_NRFT.reason  = 'Width Pull'
GROUP BY Reason
```

Reason	Total
Bar Not Soaked Enough	25
Coiler Snatch	44
ANC Stroke	20
Tail End Pull	1
Full Coil	175
Roughing capsule	4
FM Pull	515
Poor finishing temperature	1
Slab/ Coil Temperature	90
temperature	14
RM Flare	96
RM Necking	330
Width Pull	2

Figure 4.12: HeidiSQL query to retrieve the names and counts of those defects in the TBL_QC_NRFT table that are relevant to the experiments in this thesis.



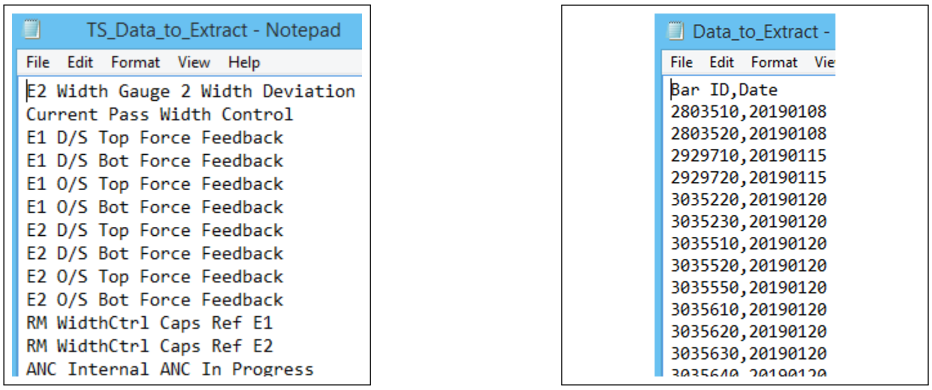
```sql
SELECT

CoilSummaryData.coilid,
CoilSummaryData.grade,
icounter.datei,
TBL_QC_NRFT.plant,
TBL_QC_NRFT.reason,
TBL_QC_NRFT.comment,
TBL_QC_NRFT.root_cause,
TBL_QC_NRFT.delayed_bar,
TBL_QC_NRFT.mill_delay,
TBL_QC_NRFT.stability,
width_alarm = sgms.Width_Av_Alarm,
```

coilid	grade	datei	plant	reason	comment	root_cause	delayed_bar	mill_delay	stability	widt
8276110	3335	20,230,926	Roughing	RM Necking	E2 edger unavailable	Head and Tail End Pulls	0	0	0	Fals
8276010	3335	20,230,926	Roughing	RM Necking	E2 Edger unavailable	Head and Tail End Pulls	0	0	0	Fals
8213710	3046	20,230,923	Roughing	RM Necking	E2 capsules unavailable due to ...	Head End Pull	0	0	0	Fals
8213510	3190	20,230,923	Roughing	RM Necking	E2 capsules unavailable due to ...	Head End Pull	0	0	0	Fals
8213410	3046	20,230,923	Roughing	RM Necking	E2 capsules unavailable due to ...	Head End Pull	0	0	0	Fals
8213210	3046	20,230,923	Roughing	RM Necking	E2 capsules unavailable due to ...	Head End Pull	0	0	0	Fals
8213010	3190	20,230,923	Roughing	RM Necking	E2 capsules unavailable due to ...	Head End Pull	0	0	0	Fals
8210010	3816	20,230,923	Roughing	RM Necking	E2 capsules unavailable due to ...	Head End Pull	0	0	0	Fals
8197910	3190	20,230,922	Roughing	RM Necking	E2 capsules unavailable due to ...	Head End Pull	0	0	0	Fals
8018510	3088	20,230,916	Roughing	RM Necking	ANC stroke slightly early on Pa...	Head End Pull	0	0	0	Fals
7918830	3832	20,230,913	Roughing	RM Necking		anc early pass 1	0	0	0	Fals
7875310	3130	20,230,910	Roughing	ANC Stroke	anc too late and -2mm on the s...	Select	0	0	0	Fals
7875230	3130	20,230,910	Finishing Mill	ANC Stroke	9mm model error on FM and AN...	RM model error	0	0	0	Fals
7395810	3366	20,230,826	Roughing	RM Necking		Select	0	0	0	Fals
7387510	3190	20,230,826	Roughing	ANC Stroke		anc too late and pull ...	0	0	0	Fals
7301310	3604	20,230,822	Roughing	RM Flare	ANC used on all 5 passes. No ...	Head and Tail End Flair	0	0	0	Fals

Figure 4.13: Results given by the final HeidiSQL query for static data.

Following this, a final list of time series required from IBA Analyzer, shown in Table 5, was also reviewed with analysts. This list, along with all of the coil IDs and corresponding dates was, sent to an analyst with the permissions required to perform the extraction. The data for each coil was received in individual .csv files and split between subprocesses, as shown in Figure 4.14 (a) and (b) and Figure 4.15 (a) and (b).



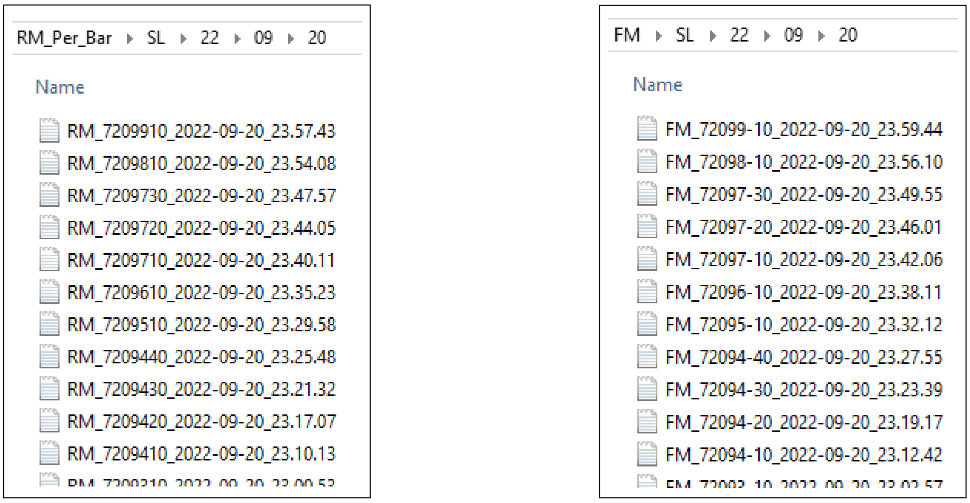
```
E2 Width Gauge 2 Width Deviation
Current Pass Width Control
E1 D/S Top Force Feedback
E1 D/S Bot Force Feedback
E1 O/S Top Force Feedback
E1 O/S Bot Force Feedback
E2 D/S Top Force Feedback
E2 D/S Bot Force Feedback
E2 O/S Top Force Feedback
E2 O/S Bot Force Feedback
RM WidthCtrl Caps Ref E1
RM WidthCtrl Caps Ref E2
ANC Internal ANC In Progress
```

```
Bar ID,Date
2803510,20190108
2803520,20190108
2929710,20190115
2929720,20190115
3035220,20190120
3035230,20190120
3035510,20190120
3035520,20190120
3035550,20190120
3035610,20190120
3035620,20190120
3035630,20190120
3035640,20190120
```

(a) Text file containing the required time series data.

(b) Text file containing the IDs and dates of the required strips.

Figure 4.14: Screenshots of text files containing the requested information of the required strips and time series data, sent to an analyst for extraction.



```
RM_7209910_2022-09-20_23.57.43
RM_7209810_2022-09-20_23.54.08
RM_7209730_2022-09-20_23.47.57
RM_7209720_2022-09-20_23.44.05
RM_7209710_2022-09-20_23.40.11
RM_7209610_2022-09-20_23.35.23
RM_7209510_2022-09-20_23.29.58
RM_7209440_2022-09-20_23.25.48
RM_7209430_2022-09-20_23.21.32
RM_7209420_2022-09-20_23.17.07
RM_7209410_2022-09-20_23.10.13
RM_7209310_2022-09-20_23.00.53
```

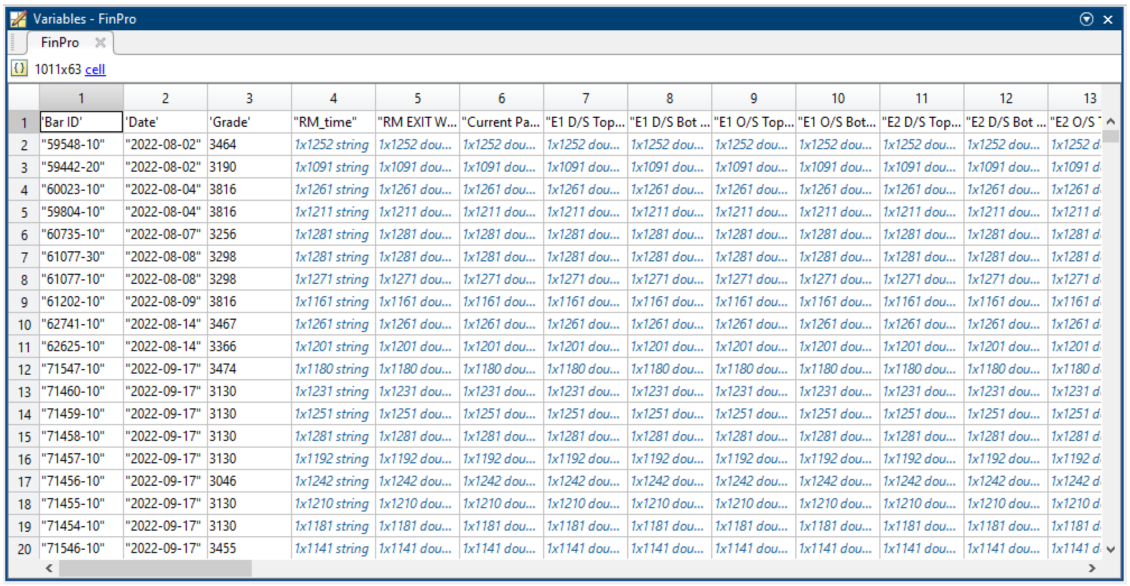
```
FM_72099-10_2022-09-20_23.59.44
FM_72098-10_2022-09-20_23.56.10
FM_72097-30_2022-09-20_23.49.55
FM_72097-20_2022-09-20_23.46.01
FM_72097-10_2022-09-20_23.42.06
FM_72096-10_2022-09-20_23.38.11
FM_72095-10_2022-09-20_23.32.12
FM_72094-40_2022-09-20_23.27.55
FM_72094-30_2022-09-20_23.23.39
FM_72094-20_2022-09-20_23.19.17
FM_72094-10_2022-09-20_23.12.42
FM_72093-10_2022-09-20_23.02.57
```

(a) Files containing time series data related to the Finishing Mill subprocess.

(b) Files containing time series data related to the Finishing Mill subprocess.

Figure 4.15: Screenshots of text files containing the requested information of the required strips and signals, sent to an analyst for extraction.

A new table data structure was created within MATLAB and used to combine all gathered data from the .csv files from the previous steps. Firstly, the .csv exported from the EngServer2 database was read into the table structure in the same format. This enabled iteration through the file structure of the received IBA extraction data to match the coil IDs and dates found in the file names to those already in the table. New columns were created in the table structure to accommodate this data. It should be noted that both coil IDs and dates are used as sample identifiers since coil IDs are recycled in the HSM every 9 to 12 months. The result of this process is a single table structure containing all relevant static and time series data for each coil, indexed by coil ID and date. From this, rows, columns and samples can be queried by subprocess, defect label, or any other desired field, to simplify the subsequent pre-processing stages. The resulting table structure in the MATLAB environment is displayed in Figure 4.16, while the names and descriptions of all static data retrieved from the database and time series data retrieved from IBA are shown in Tables 4 and 5, respectively.



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	'Bar ID'	'Date'	'Grade'	"RM_time"	"RM EXIT W...	"Current Pa...	"E1 D/S Top...	"E1 D/S Bot ...	"E1 O/S Top...	"E1 O/S Bot...	"E2 D/S Top...	"E2 D/S Bot ...	"E2 O/S
2	"59548-10"	"2022-08-02"	3464	1x1252 string	1x1252 dou...	1x1252 dou...	1x1252 dou...	1x1252 dou...	1x1252 dou...	1x1252 dou...	1x1252 dou...	1x1252 dou...	1x1252 d
3	"59442-20"	"2022-08-02"	3190	1x1091 string	1x1091 dou...	1x1091 dou...	1x1091 dou...	1x1091 dou...	1x1091 dou...	1x1091 dou...	1x1091 dou...	1x1091 dou...	1x1091 d
4	"60023-10"	"2022-08-04"	3816	1x1261 string	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 d
5	"59804-10"	"2022-08-04"	3816	1x1211 string	1x1211 dou...	1x1211 dou...	1x1211 dou...	1x1211 dou...	1x1211 dou...	1x1211 dou...	1x1211 dou...	1x1211 dou...	1x1211 d
6	"60735-10"	"2022-08-07"	3256	1x1281 string	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 d
7	"61077-30"	"2022-08-08"	3298	1x1281 string	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 d
8	"61077-10"	"2022-08-08"	3298	1x1271 string	1x1271 dou...	1x1271 dou...	1x1271 dou...	1x1271 dou...	1x1271 dou...	1x1271 dou...	1x1271 dou...	1x1271 dou...	1x1271 d
9	"61202-10"	"2022-08-09"	3816	1x1161 string	1x1161 dou...	1x1161 dou...	1x1161 dou...	1x1161 dou...	1x1161 dou...	1x1161 dou...	1x1161 dou...	1x1161 dou...	1x1161 d
10	"62741-10"	"2022-08-14"	3467	1x1261 string	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 dou...	1x1261 d
11	"62625-10"	"2022-08-14"	3366	1x1201 string	1x1201 dou...	1x1201 dou...	1x1201 dou...	1x1201 dou...	1x1201 dou...	1x1201 dou...	1x1201 dou...	1x1201 dou...	1x1201 d
12	"71547-10"	"2022-09-17"	3474	1x1180 string	1x1180 dou...	1x1180 dou...	1x1180 dou...	1x1180 dou...	1x1180 dou...	1x1180 dou...	1x1180 dou...	1x1180 dou...	1x1180 d
13	"71460-10"	"2022-09-17"	3130	1x1231 string	1x1231 dou...	1x1231 dou...	1x1231 dou...	1x1231 dou...	1x1231 dou...	1x1231 dou...	1x1231 dou...	1x1231 dou...	1x1231 d
14	"71459-10"	"2022-09-17"	3130	1x1251 string	1x1251 dou...	1x1251 dou...	1x1251 dou...	1x1251 dou...	1x1251 dou...	1x1251 dou...	1x1251 dou...	1x1251 dou...	1x1251 d
15	"71458-10"	"2022-09-17"	3130	1x1281 string	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 dou...	1x1281 d
16	"71457-10"	"2022-09-17"	3130	1x1192 string	1x1192 dou...	1x1192 dou...	1x1192 dou...	1x1192 dou...	1x1192 dou...	1x1192 dou...	1x1192 dou...	1x1192 dou...	1x1192 d
17	"71456-10"	"2022-09-17"	3046	1x1242 string	1x1242 dou...	1x1242 dou...	1x1242 dou...	1x1242 dou...	1x1242 dou...	1x1242 dou...	1x1242 dou...	1x1242 dou...	1x1242 d
18	"71455-10"	"2022-09-17"	3130	1x1210 string	1x1210 dou...	1x1210 dou...	1x1210 dou...	1x1210 dou...	1x1210 dou...	1x1210 dou...	1x1210 dou...	1x1210 dou...	1x1210 d
19	"71454-10"	"2022-09-17"	3130	1x1181 string	1x1181 dou...	1x1181 dou...	1x1181 dou...	1x1181 dou...	1x1181 dou...	1x1181 dou...	1x1181 dou...	1x1181 dou...	1x1181 d
20	"71546-10"	"2022-09-17"	3455	1x1141 string	1x1141 dou...	1x1141 dou...	1x1141 dou...	1x1141 dou...	1x1141 dou...	1x1141 dou...	1x1141 dou...	1x1141 dou...	1x1141 d

Figure 4.16: Resulting table structure in the MATLAB workspace.

Table 4: Summary of static HSM data collected by querying database tables used to form the dataset for the proposed web tool.

Name	**Description/Purpose**	**Storage Location /Table in Database**
coilid	ID number of the strip/coil.	CoilSummaryData
grade	Identified for the strip/coil's chemical composition.	CoilSummaryData
datei	The date on which the strip was rolled.	icounter
COIL_TEMP _TOL_PLUS	Upper tolerance for temperature in the Coiler subprocess.	hfncpdi
COIL_TEMP _TOL_MINUS	Lower tolerance for temperature in the Coiler subprocess.	hfncpdi
width_alarm	True/False alarm indicating whether Spread/Squeeze rules have been broken.	sgms
S_S	The maximum width to which the bar can be rolled to.	scheduling_pdi
sum_of_offsets	Offsets made by both operators and models throughout the HSM process.	hffmmisc
RM_model_error	Difference between the Roughing Mill model's aim and the width of the strip at the end of the Roughing Mill subprocess.	hfrmadlog
FM_model_error	Difference between the Finishing Mill model's aim and the width of the strip at the end of the Finishing Mill subprocess.	hffmadlog
pos_tol1	Upper tolerance for width in both the Finishing Mill and Coiler subprocesses.	hfncpdi
neg_tol1	Lower tolerance for width in both the Finishing Mill and Coiler subprocesses.	hfncpdi
plant	Subprocess in which a defect originates from, according to an analyst.	TBL_QC_NRFT

reason	A defect which occurred in the HSM process, according to an analyst.	TBL_QC_NRFT
root_cause	The root cause of a defect, according to an analyst.	TBL_QC_NRFT
comment	Further reasoning for why a defect occurred, according to an analyst.	TBL_QC_NRFT
delayed_bar	True/False alarm indicating whether a bar is delayed after being heated in the Furnace and before entering the Roughing Mill subprocess.	TBL_QC_NRFT
mill_delay	True/False alarm indicating whether the HSM is currently using Single Furnace Operation.	TBL_QC_NRFT

Table 5: Summary of HSM time series data collected from the IBA server and used to form the dataset for the proposed tool.

Name	**Description/Purpose**
Current Pass Width Control	Number of the current Roughing Mill pass.
E1 D/S Top Force Feedback	Roller force at the top of the driver side roller of Edger 1.
E1 D/S Bot Force Feedback	Roller force at the bottom of the driver side roller of Edger 1.
E1 O/S Top Force Feedback	Roller force at the top of the operator side roller of Edger 1.
E1 O/S Bot Force Feedback	Roller force at the bottom of the operator side roller of Edger 1.
E2 D/S Top Force Feedback	Roller force at the top of the driver side roller of Edger 2.
E2 D/S Bot Force Feedback	Roller force at the bottom of the driver side roller of Edger 2.

E2 O/S Top Force Feedback	Roller force at the top of the operator side roller of Edger 2.
E2 O/S Bot Force Feedback	Roller force at the bottom of the operator side roller of Edger 2.
RM WidthCtrl Caps Ref E1	Capsule position of Edger 1.
RM WidthCtrl Caps Ref E2	Capsule position of Edger 2.
ANC Internal ANC In Progress	Binary signal indicating whether ANC is currently active.
E2_WidthMeter _InMeasure	Width deviation of the bar at Edger 2 used for Roughing Mill exit width.
F11 Temp Upper Limit	Upper tolerance for temperature at the exit of the Finishing Mill subprocess.
F11 Temp Lower Limit	Lower tolerance for temperature at the exit of the Finishing Mill subprocess.
F11_Width_Deviation	Width deviation at the exit of the Finishing Mill subprocess.
F11_MIM	Binary signal indicating whether the strip is currently present at the exit of the Finishing Mill subprocess.
F11_Exit _Temperature_A	Temperature of the top of the strip at the exit of the Finishing Mill subprocess.
F11_Exit _Temperature_B	Temperature of the bottom of the strip at the exit of the Finishing Mill subprocess.
C4_Width _WidthDeviation	Width debiation of the strip in the Coiler subprocess.
Coiling_Temperature _Top_A	Temperature of the top of the strip in the Coiler subprocess.
Coiling_Temperature _Top_B	Temperature of the bottom of the strip in the Coiler subprocess.

4.2.2 Modelling and Analysis of Results

4.2.2.1 Algorithm and Hyperparameter Optimisation

For the image classification tasks in this chapter, CNN and Transfer Learning has been chosen. Specifically, the pre-trained network architecture, GoogLeNet [8], is used and the final two fully connected and output layers are replaced and retrained with the respective classes in each classification task's image dataset. Image classification as a methodology is also selected due to its ability to emulate the visual inspection of images which is typically carried out by humans, having also previously being applied in numerous time series classification applications, including those which use images of time series plots [22, 131, 132, 161]. A Transfer Learning approach is chosen due to its ability to utilise knowledge from previous image training data and to apply it to a new classification task [89, 162, 163], and GoogLeNet is selected due to its depth and its use of inception modules which allow it to use a larger number of convolutional layers with less data loss [8, 90, 164].

To determine the optimal hyperparameters for each of the CNNs created in this chapter, GSO [143] is used. This technique has been chosen due to its exhaustive method of providing the broadest possible evaluation by testing every possible combination. While some literature argues that initial learning rate can have a major impact on the training and testing performance of a neural network [142], from experiments performed prior to the creation of the image classification model in Chapter 5, it was determined that drastic changes in its value result only in negative performance changes [22]. From this experiment, it was also determined that the Learn Rate Drop Factor and L2 Regularization values do not have a significant impact on performance, while Max Epochs and Mini Batch Size have the most significant impact. By using a reduced number of hyperparameter values, the total training time has also greatly been reduced. The Cartesian product of Table 6, which contains a range of values for Max Epochs and Mini Batch Size, is therefore used as the set of hyperparameter values used to train each model in this chapter. Default values of 0.001, 0.3, and 0.0001 were used for the Initial Learn Rate, Learn Rate Drop Factor, and L2 Regularization values, respectively.

Table 6: Hyperparameter values used for GSO in the experiments carried out in this chapter.

Max Epochs	**Mini Batch Size**
5	2
10	4
15	8
20	16
25	32

4.2.2.2 Cross Validation

In this chapter, K-FCV is used to determine how well and how reliably each model classifies new, unseen data [150]. This method of Cross Validation is chosen for the experiments in this chapter as it is more likely to avoid overfitting than other types of Cross Validation such as HOCV due to its extensive method of splitting the dataset into folds. Although a single K-FCV experiment is more computationally expensive than a single HOCV experiment, the total number of training and testing runs to be completed across all three experiments is generally low [165]. A K value of 5 is chosen such that the datasets used in each classification task are split into five parts, each amounting to 20% of the total samples belonging to each label for each task. This is performed in parallel with GSO such that 5-FCV is performed for each hyperparamter value combination.

4.3 Detecting Necking and Flare in the Roughing Mill Subprocess

In this section, an image classification model is created to distinguish between Necking, Flare, and normal width behaviour within the Roughing Mill subprocess. To achieve this, a dataset consisting of samples having these defects, and a corresponding number of samples with no defective behaviour are extracted from the dataset acquired in Section 4.2.1. These samples are pre-processed and used to create images which display the important visual elements of the processed time series data. These images form the final dataset used to train CNN image classification models, which are then evaluated to determine the optimal configuration and, therefore, which model is to be used in the decision-making tool proposed in Chapter 6. This process is illustrated in Figure 4.17 and each step is described in detail in Sections 4.3.1 and 4.3.2.

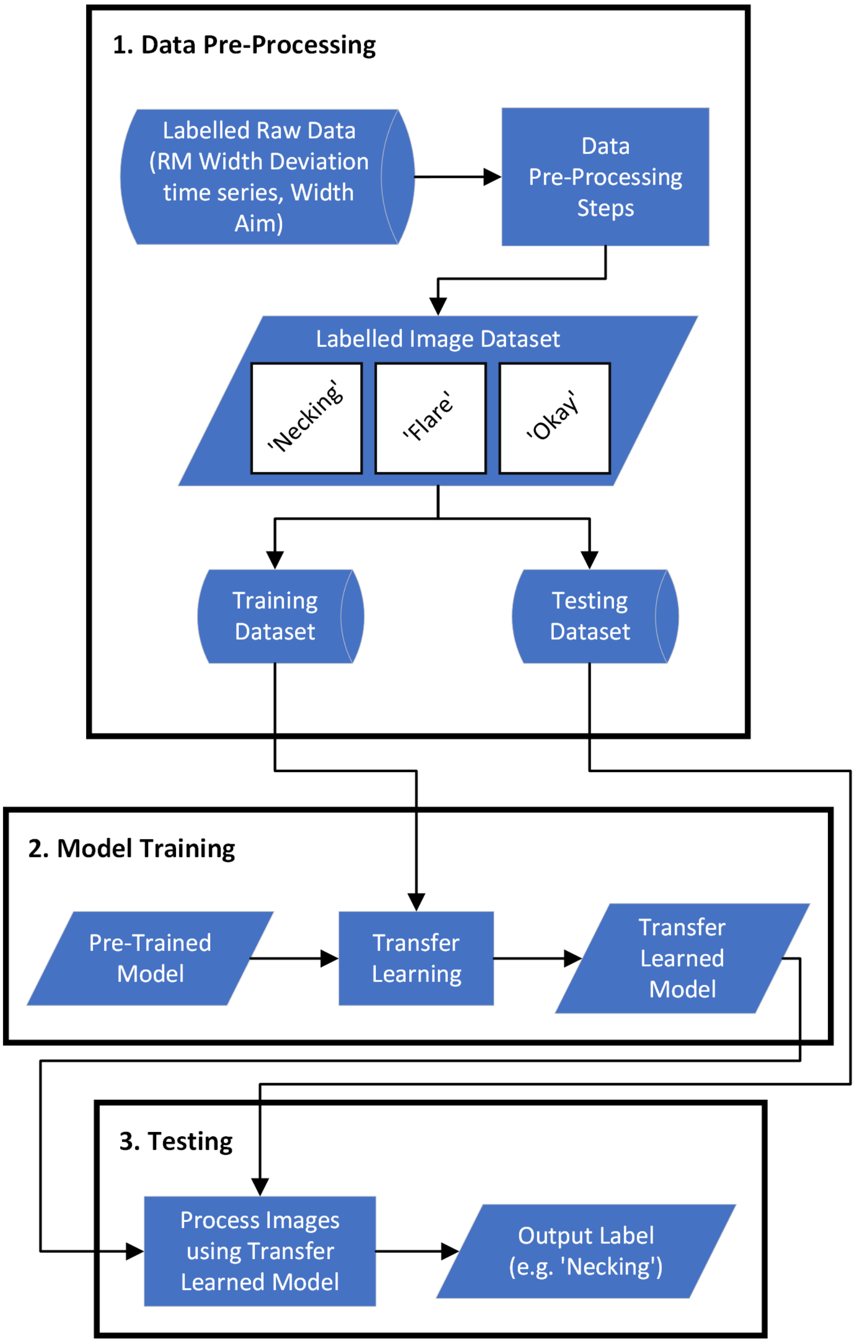


Figure 4.17: High-Level flowchart describing the steps taken to conduct the Roughing Mill Necking and Flare classification task.

4.3.1 Data Pre-Processing

The datasets used to train and test all models in each chapter consist of samples which are labelled as having the corresponding defective behaviours, as well as normal behaviour, and are compiled using real historical data from the Port Talbot HSM, as discussed in the previous section. The dataset used in this particular experiment contains 189 samples which display the behaviour of Necking, 200 samples which show the behaviour of Flare, and 171 samples which show normal behaviour. Table 7 illustrates the distribution of samples between labels as well as how samples for each label are split in order to perform 5-FCV.

Table 7: Number of samples belonging to each label for each fold in the Roughing Mill Necking and Flare classification task.

Cross Validation Fold	**Necking**	**Flare**	**Okay**
Fold 1	37	40	34
Fold 2	37	40	34
Fold 3	37	40	34
Fold 4	38	40	34
Fold 5	38	40	35
Total (Label)	189	200	171

The images used in this classification task must first be constructed from raw signal data collected during the HSM process. For this classification task, each sample is represented using an image constructed from the Roughing Mill width deviation signal ‘E2_WidthMeter_InMeasure’ and a constant y-axis value of 0 to represent the width aim. Limits of 50mm and -50mm can be set on the y-axis of the time series as values outside of this range are usually out of the capability of automatic adaption and thus show that there is a fundamental issue with width or the process which requires further examination by analysts. Redundant data indicating that the bar is not present in the Roughing Mill is also removed. Next, the time series is normalised to a range between 0 and 1, using formula (5), such that bars with different width aims and ranges are conventionally mapped on the same scale. In this case, -50mm and 50mm are mapped to 0 and 1, respectively.

$$\tilde{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{5}$$

From this, the first and last 15% of the times series, the head and tail ends of the bar, are considered as separate classification problems as it is possible for a defect to occur only on one end of the bar or for Necking and Flare to occur independently on opposite ends of the bar. The tail end data is also represented in reverse such that these defects are represented in the same direction as they would be on the head end. This reduces generalisation errors during training and enables the creation of a larger image dataset for each class, as opposed to a class for each defect on each end of the bar, which is ultimately unnecessary. It should also be noted that analysts consider a larger portion of the bar, 15%, for its head and tail ends in the Roughing Mill, as opposed to the standard 10% considered in later subprocesses, due to its usually large increase in length over time. An example of this process being carried out on a the time series data for a bar labelled as having Necking in its head end is shown in Figure 4.18.

The resulting figures are 224 pixel x 224 pixel x 3 colour channel image representations of the remaining, and relevant, time series data and the plotted constant y-axis value of 0, which constitute the final dataset used in the image classification task in this experiment. An example image from each of the Okay, Flare, and Necking classes is shown in Figure 4.19 (a), (b), and (c) from left to right, respectively. As discussed in the opening section of this chapter, Necking can be identified by a decrease in width at the head or tail end of a bar, while Flare can be identified by an increase in width. These defects are easier to identify visually rather than numerically as raw time series values can fluctuate and noisy data can be more difficult to separate without complex statistical analysis. Normal behaviour shows steady performance throughout the strip.

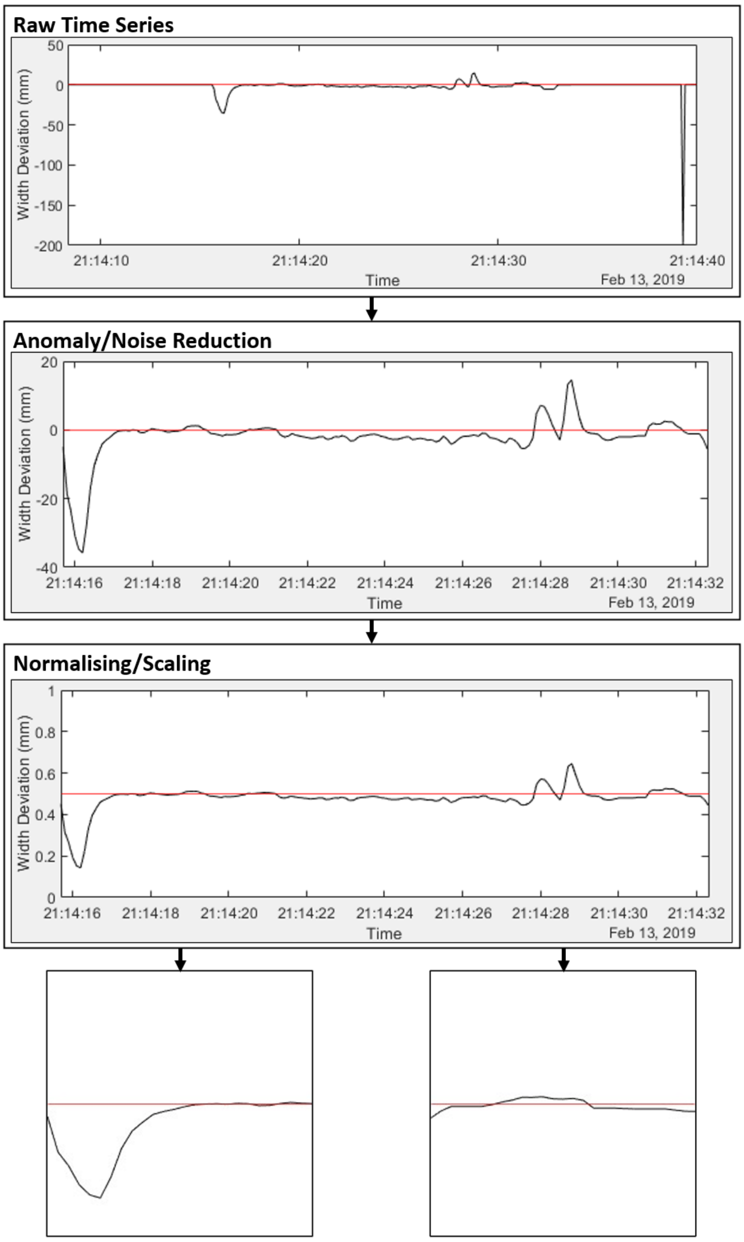
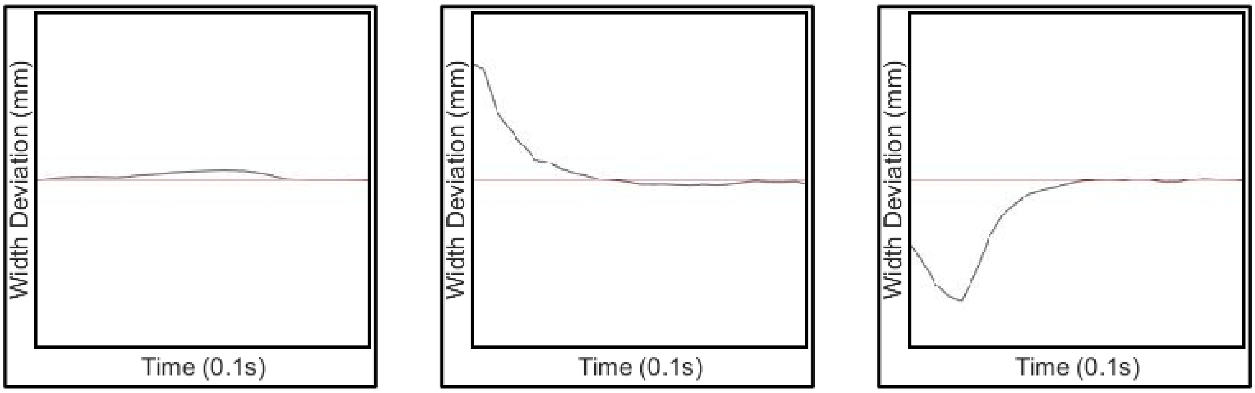


Figure 4.18: Steps for pre-processing Roughing Mill time series data into images for the Necking and Flare classification task.

GoogLeNet's input layer is configured to receive images with a width and length of 244 by 244 pixels. While this layer can be reconfigured to receive any image size, the layer is left in the default configuration since the image size is suitable for displaying the time series data without losing much resolution. Reducing the image size may begin to degrade the quality of the data represented in the image, while increasing the image size may add to both data pre-processing and model computation times.



(a) A sample showing no signs of defective behavior.

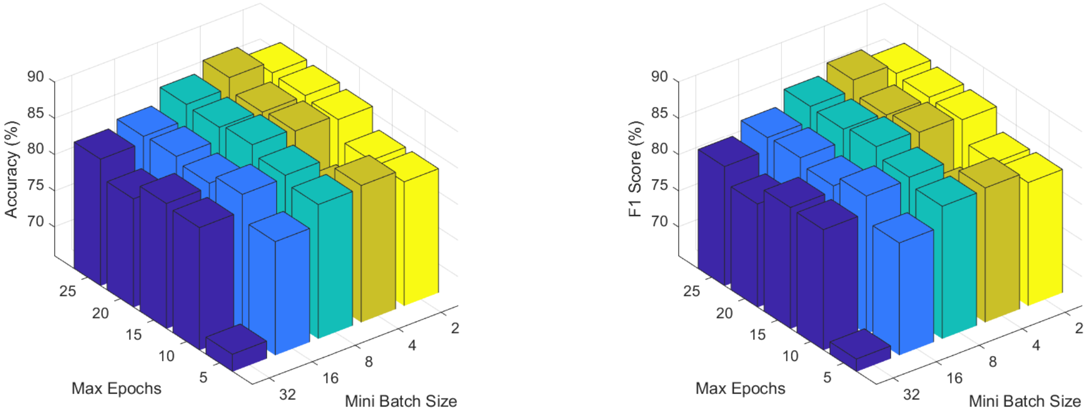
(b) A sample showing behaviours of Flare.

(c) A sample showing behaviours of Necking.

Figure 4.19: Examples of images contained in the dataset used for the Roughing Mill Necking and Flare classification task.

4.3.2 Modelling and Analysis of Results

GSO was performed simultaneously with 5-FCV using the folds described in Table 7. Figure 4.20 shows the average testing performance of all folds for each hyperparameter value for the Roughing Mill Necking and Flare classification task. Within Figure 4.20, (a) shows accuracy while (b) shows F1 Score which provides an insight into the precision and recall of the model.



(a) Average accuracy.

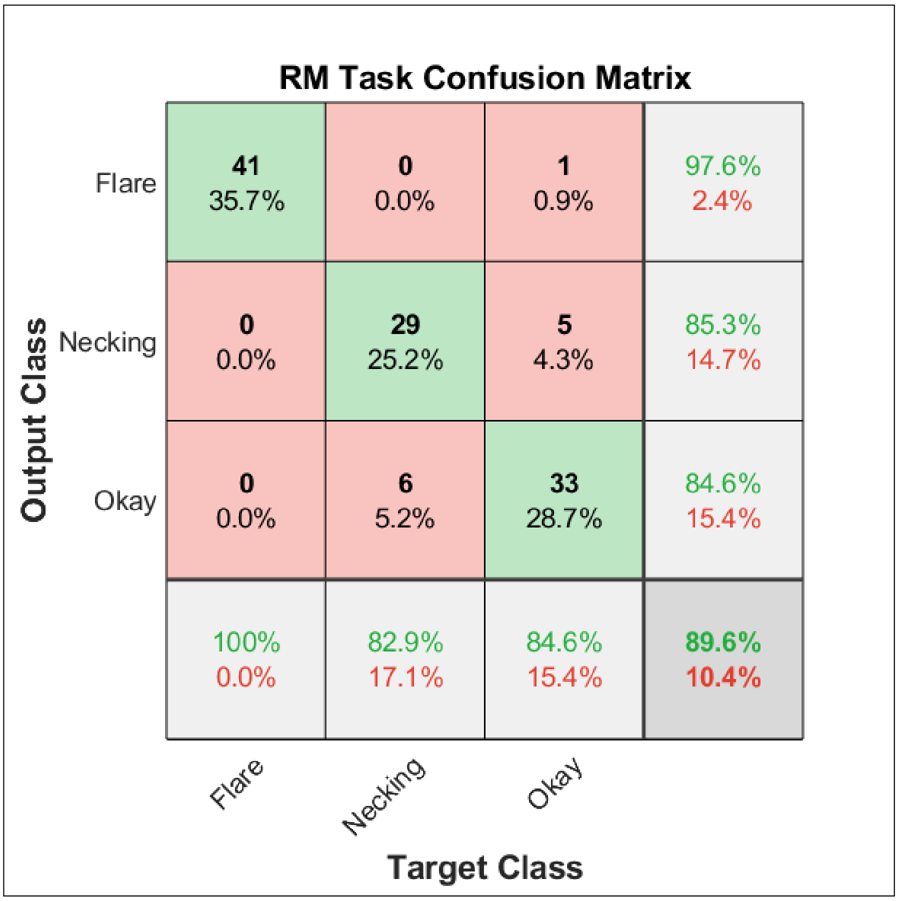
(b) Average F1 Score.

Figure 4.20: Average 5-FCV performance of each hyperparameter value combination for GSO in the Roughing Mill Necking and Flare classification task.

Figure 4.20 and Table 8 show that as Max Epochs increase, so does model performance. This may be because a higher max epochs increases the number of iterations, allowing the model to continue learning for a longer time. This can be confirmed by the results of using a Max Epochs value of 5, which shows the lowest performance across all Mini Batch Size values. The performance of each Mini Batch Size does not show a completely linear pattern across all Max Epoch values. Despite this, it is clear that as Max Epochs increases, a lower Mini Batch Size value becomes more favourable for this classification task. It should also be noted that when using a low Mini Batch Size value, model performance slightly decreases when moving from a Max Epochs value of 20 to 25. It is possible that this is because the model begins to overfit due to the higher number of iterations. Based on these findings, a Max Epochs value of 20 and a Mini Batch Size of 2 have therefore been selected as the final hyperparameter values for the final Roughing Mill Necking and Flare image classification model. Table 8 also shows this to be the best overall performing hyperparameter value combination with both the average accuracy and F1 Score for the 5-FCV splits being 88%. Figure 4.21 show a confusion matrix which shows the classification results of a randomly selected fold from this experiment using the selected hyperparameter values.

Table 8: Averaged 5-FCV results for GSO in the Roughing Mill Necking and Flare classification task.

		Max Epochs									
		5		10		15		20		25	
		Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
Mini Batch Size	2	83.49	82.47	84.37	84.20	86.50	86.23	87.93	87.60	86.32	86.23
	4	80.98	80.27	84.56	84.38	86.31	86.06	86.13	85.73	86.14	85.91
	8	83.30	82.75	83.82	83.46	86.86	86.60	86.48	86.30	85.79	85.76
	16	82.93	82.65	85.25	85.02	85.61	85.34	80.64	80.29	83.12	82.85
	32	68.41	67.75	81.68	81.51	84.53	84.30	84.89	84.56	83.12	83.02

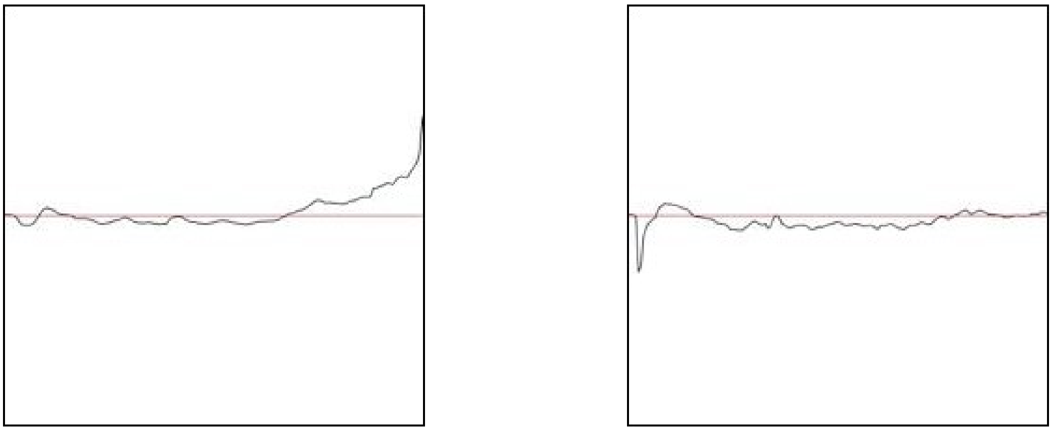


RM Task Confusion Matrix

Output Class \ Target Class	Flare	Necking	Okay	
Flare	**41** 35.7%	**0** 0.0%	**1** 0.9%	97.6% 2.4%
Necking	**0** 0.0%	**29** 25.2%	**5** 4.3%	85.3% 14.7%
Okay	**0** 0.0%	**6** 5.2%	**33** 28.7%	84.6% 15.4%
	100% 0.0%	82.9% 17.1%	84.6% 15.4%	**89.6%** **10.4%**

Figure 4.21: Classification results of a randomly selected fold using the selected hyperparameter values for the Roughing Mill Necking and Flare classification task.

While the model performed relatively well, it is important to consider the misclassifications that were made, why they may have happened, and, from this, how they may be addressed in the future. Figure 4.22 (a) and (b) show two such examples that were classified incorrectly. The target class of both images was Okay, however, the model classified Figure 4.22 (a) as having Flare and (b) as having Necking. There are a few possible explanations for this. These are not very extreme cases of Necking or Flare, and, while the model generally performs well, this could also be the result of limited data availability preventing the model from generalising between the defective classes and the Okay class. Another possibility is that these are indeed instances of Necking and Flare, however they either may have been missed by operators or left unrecorded by analysts. This particular example was reviewed with an analyst who confirmed, from their visual assessment, that this may be the case. However, it may not have been recorded as there were no other width-related issues with this bar in later subprocesses, meaning that the final strip was within specification by the end of the HSM process. This also highlights the potential issues associated with human error in data capture for high-throughput manufacturing process. To improve the model's reliability, more images should be collected over time with the intention of retraining and monitoring of its performance.



(a) An image belonging to the Okay class, labelled as having Flare.

(b) An image belonging to the Okay class, labelled as having Necking.

Figure 4.22: Examples of images which were misclassified by the Roughing Mill Necking and Flare classification model.

4.4 Detecting Width Pull in the Finishing Mill Subprocess

In this section, an image classification model is created to distinguish between Width Pull and normal behaviour within the Finishing Mill subprocess. To achieve this, the same process described in Section 4.3 and Figure 4.17 has been applied. A dataset consisting of samples labeled as having 'Width Pull', and a corresponding number of samples with no defective behaviour, labeled 'Okay', are extracted from the dataset acquired in Section 4.2.1. Corresponding images are created and a CNN image classification model is trained and optimised, similarly to the previous experiment. The process used for this experiment is illustrated in Figure Figure 4.23 and each step is described in detail in Sections 4.4.1 and 4.4.2.

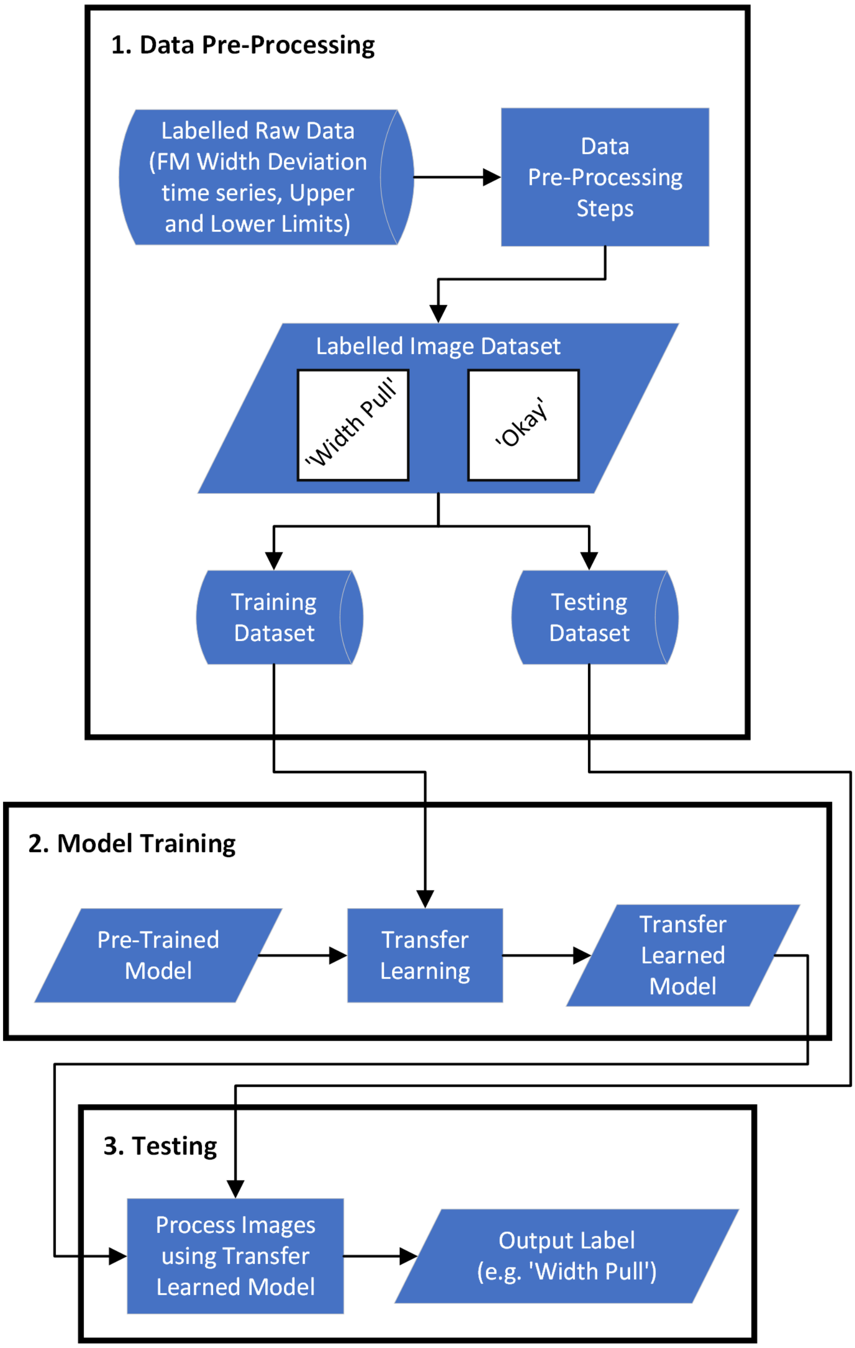


Figure 4.23: High-Level flowchart describing the steps taken to conduct the Finishing Mill Width Pull classification task.

4.4.1 Data Pre-Processing

The dataset used in this particular experiment contains 223 samples which display the behaviour of Width Pull, and 225 samples which show normal behaviour. Table 9 illustrates the distribution of samples between labels as well as how samples for each label are split in order to perform 5-FCV.

Table 9: Number of samples belonging to each label for each fold in the Finishing Mill Width Pull classification task.

Cross Validation Fold	**Width Pull**	**Okay**
Fold 1	45	45
Fold 2	45	45
Fold 3	45	45
Fold 4	44	45
Fold 5	44	45
Total (Label)	223	225

For this classification task, each sample is represented using an image constructed from the Finishing Mill width deviation signal 'F11_Width_Deviation' and two constant y-axis values representing upper and lower tolerances, gathered from the static data fields 'pos_tol1' and 'neg_tol1', respectively. Upper and lower limits are then set to remove anomalies and noise. These limits are set at the lower tolerance minus 20 and the upper tolerance plus 20. This range was discussed with analysts and convened on due to the reduced range of deviation in subprocesses subsequent to the Roughing Mill. Values outside of these limits are usually outside of the capability for automatic adaption and thus show that there is a fundamental issue with width or the process which requires further examination by analysts. Redundant data where the strip is not present in the Finishing Mill is also removed. Next, the time series is normalised to a range between 0 and 1, using formula 5, such that strip with different width aims and ranges are conventionally mapped on the same scale. In this case, the lower and upper limits are mapped to 0 and 1, respectively. From this, the first 10%, the head end of the bar, is segmented. Only the head end is required in this classification problem as this is the only portion of the bar in which Width Pull occurs. An example of this process being carried out on the time series data for a bar labelled as having Width Pull in its head end is shown in Figure 4.24.

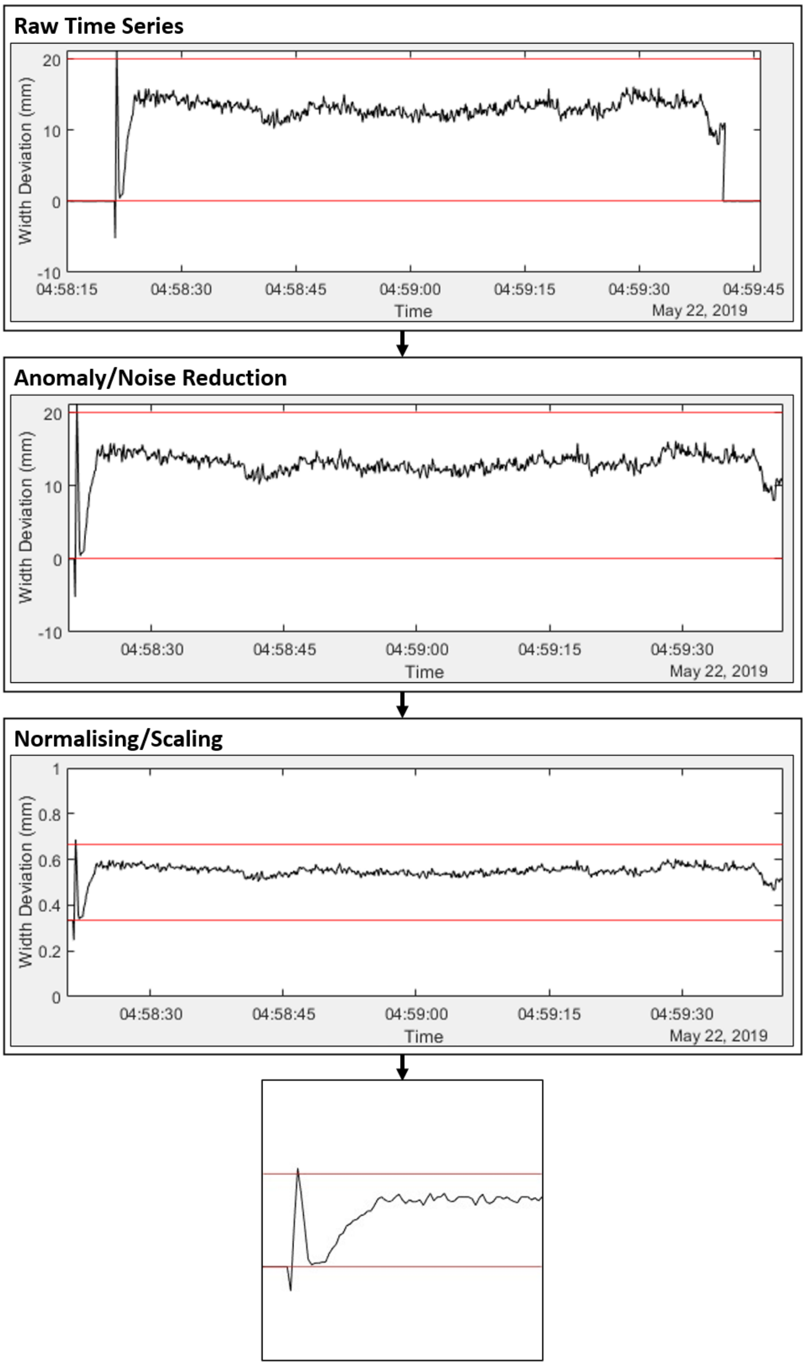
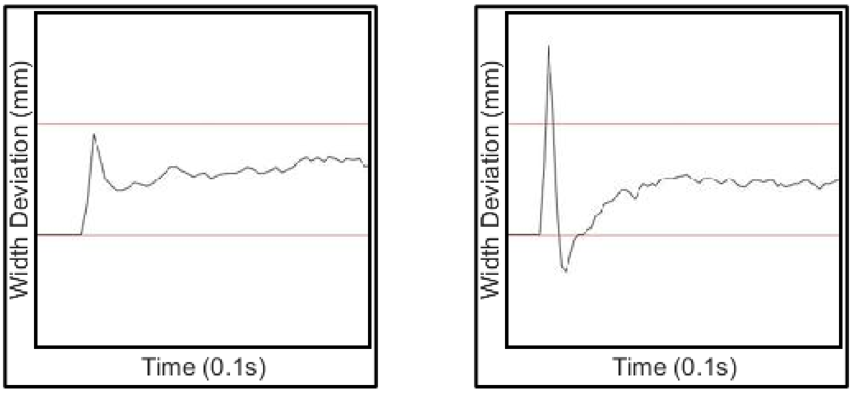


Figure 4.24: Steps for pre-processing Finishing Mill time series data into images for the Width Pull classification task.

The resulting figures are 224 pixel x 224 pixel x 3 colour channel image representations of the remaining, and relevant, time series data and the two plotted constant y-axis values representing the upper and lower width tolerances specified for each strip, which constitute the final dataset used in the image classification task in this experiment. An example image from each of the Okay and Width Pull classes is in Figure 4.25 (a) and (b) from left to right, respectively. As discussed in the opening section of this chapter, Width Pull can be identified by a sudden or gradual decrease in width but may, however, vary in magnitude and stay within tolerance. Normal behaviour shows steady performance throughout a strip, including the head end.

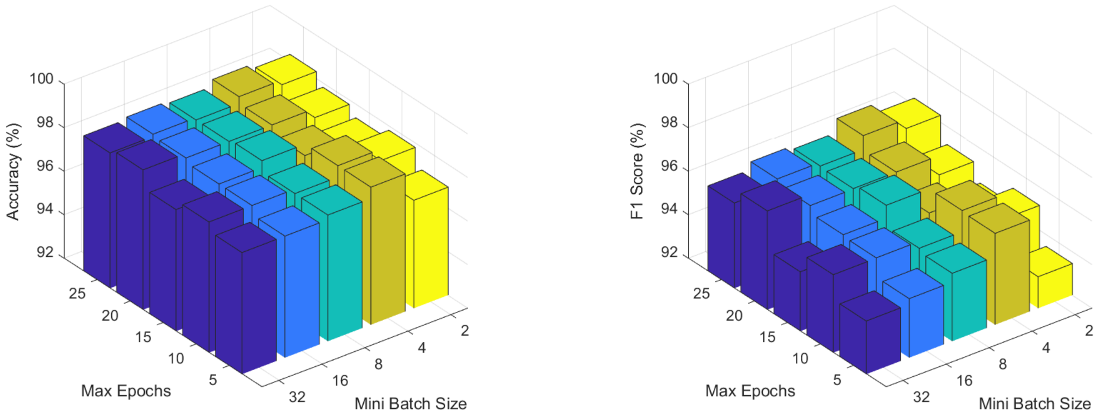


(a) A sample showing no signs of defective behavior.

(b) A sample showing behaviours of Width Pull.

Figure 4.25: Examples of images contained in the dataset used for the Finishing Mill Width Pull classification task.

4.4.2 Modelling and Analysis of Results



(a) Average accuracy.

(b) Average F1 Score.

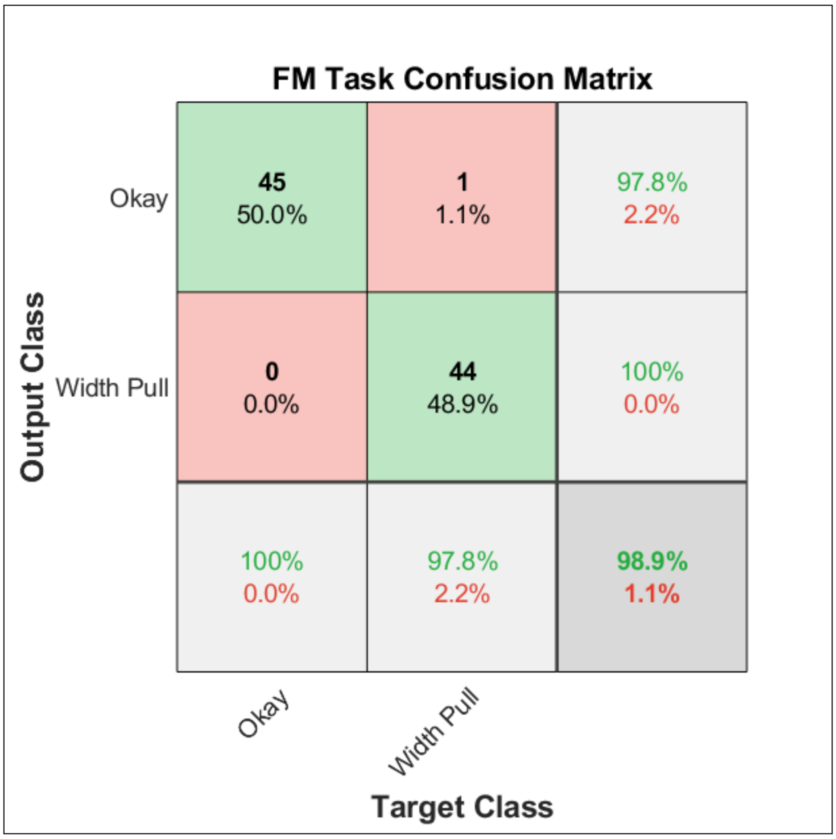
Figure 4.26: Average 5-FCV performance of each hyperparameter value combination for GSO in the Finishing Mill Width Pull classification task.

Figure 4.26 (a) and (b) show the average testing accuracy and F1 Score, respectively, of all folds for each hyperparameter value for the Finishing Mill Width Pull classification task.

Figure 4.26 and Table 10 show that, while there is only a small difference between the results of different hyperparamter values, higher Max Epochs values generally show the best overall performance. This may be because this classification task has a similar number of samples per label to the Roughing Mill Necking and Flare classification task, however, learning to distinguish between only two classes is less time consuming, when considering a low and similar number of samples. This would explain the smaller distribution in performance between high and low Max Epoch values, as opposed to the Roughing Mill Necking and Flare classification task which had a larger distribution. In addition to this, a Max Epoch value of 25 shows a higher performance result than a Max Epoch value of 5 across all Mini Batch Size values. Similarly to the Roughing Mill Necking and Flare classification task, performance generally increases as Mini Batch Size decreases. It should be noted that model performance again slightly decreases when increasing the Max Epochs value of 25, possibly due to overfitting as a result of continued training after the model already having generalised the training data. A Max Epochs value of 20 and a Mini Batch Size of 2 have therefore been selected as the final hyperparameter values for the final Finishing Mill Width Pull image classification model. Table 10 also shows this to be the best performing model overall with the average accuracy and F1 Score for the 5-FCV splits being 99% and 97%, respectively. Figure 4.27 show a confusion matrix which shows the classification results of a randomly selected fold from this experiment using the selected hyperparameter values.

Table 10: Averaged 5-FCV results for GSO in the Finishing Mill Width Pull classification task.

		Max Epochs									
		5		10		15		20		25	
		Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
Mini Batch Size	2	98.22	95.92	98.32	96.27	98.22	96.10	98.53	96.74	98.32	96.31
	4	98.43	96.54	98.22	96.03	98.22	96.05	98.22	96.05	97.80	95.17
	8	97.59	94.73	98.01	95.68	98.32	96.27	97.80	95.16	97.38	94.38
	16	98.01	95.59	98.01	95.60	97.80	95.28	98.32	96.26	97.80	95.13
	32	97.59	94.44	97.59	94.71	97.80	95.12	98.32	96.20	96.96	93.45



	Okay	Width Pull	
Okay	**45** 50.0%	**1** 1.1%	97.8% 2.2%
Width Pull	**0** 0.0%	**44** 48.9%	100% 0.0%
	100% 0.0%	97.8% 2.2%	**98.9%** **1.1%**

Figure 4.27: Classification results of a randomly selected fold using the selected hyperparameter values for the Finishing Mill Width Pull classification task.

Although the model performed well, it is again important to consider misclassifications and why they occurred. Figure 4.28 shows an example that was classified incorrectly. The target class of this image was Width Pull, but it was classified as Okay by the model. Upon initial visual inspection of this particular strip, and with further consultation from a HSM analyst, it appears to have fundamental shape issues derived from pulls and snatches along its length. This can be seen from the several small but sharp deviations in strip while it still maintains a relatively flat profile overall. After further investigation with the analysts, it was revealed that this particular strip had a low temperature and, therefore, became more difficult to roll or spread to the required width, which explains why it is largely below tolerance in this subprocess. It was originally, however, classified as having Width Pull by analysts because of the initial, elongated deviation which suggests this behaviour. This further highlights the challenges of analysing and creating models for decision-making in ambiguous case in which expert knowledge may still be required, suggesting that human interaction in the decision-making process is not necessarily counterproductive. This will be considered when designing the set of rules for decision-making in the web tool in Chapter 6, which these models will accompany.

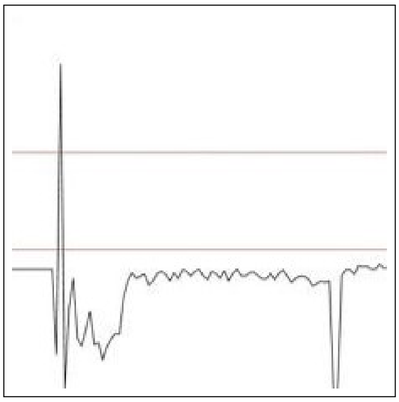


Figure 4.28: Example of an image showing Width Pull which was misclassified by the Finishing Mill Width Pull classification model as being Okay.

4.5 Detecting Snatch in the Coiler Subprocess

In this section, an image classification model is created to distinguish between Coiler Snatch and normal behaviour within the Coiler subprocess. To achieve this, the same process described in Section 4.3 and Figure 4.17 has been applied. A dataset consisting of samples labeled as having ‘Coiler Snatch’, and a corresponding number of samples with no defective behaviour, labeled ‘Okay‘, are extracted from the dataset acquired in Section 4.2.1. Corresponding images are created and a CNN image classification model is trained and optimised, similarly to the previous experiment. The process used for this experiment is illustrated in Figure Figure 4.29 and each step is described in detail in Sections 4.5.1 and 4.5.2.

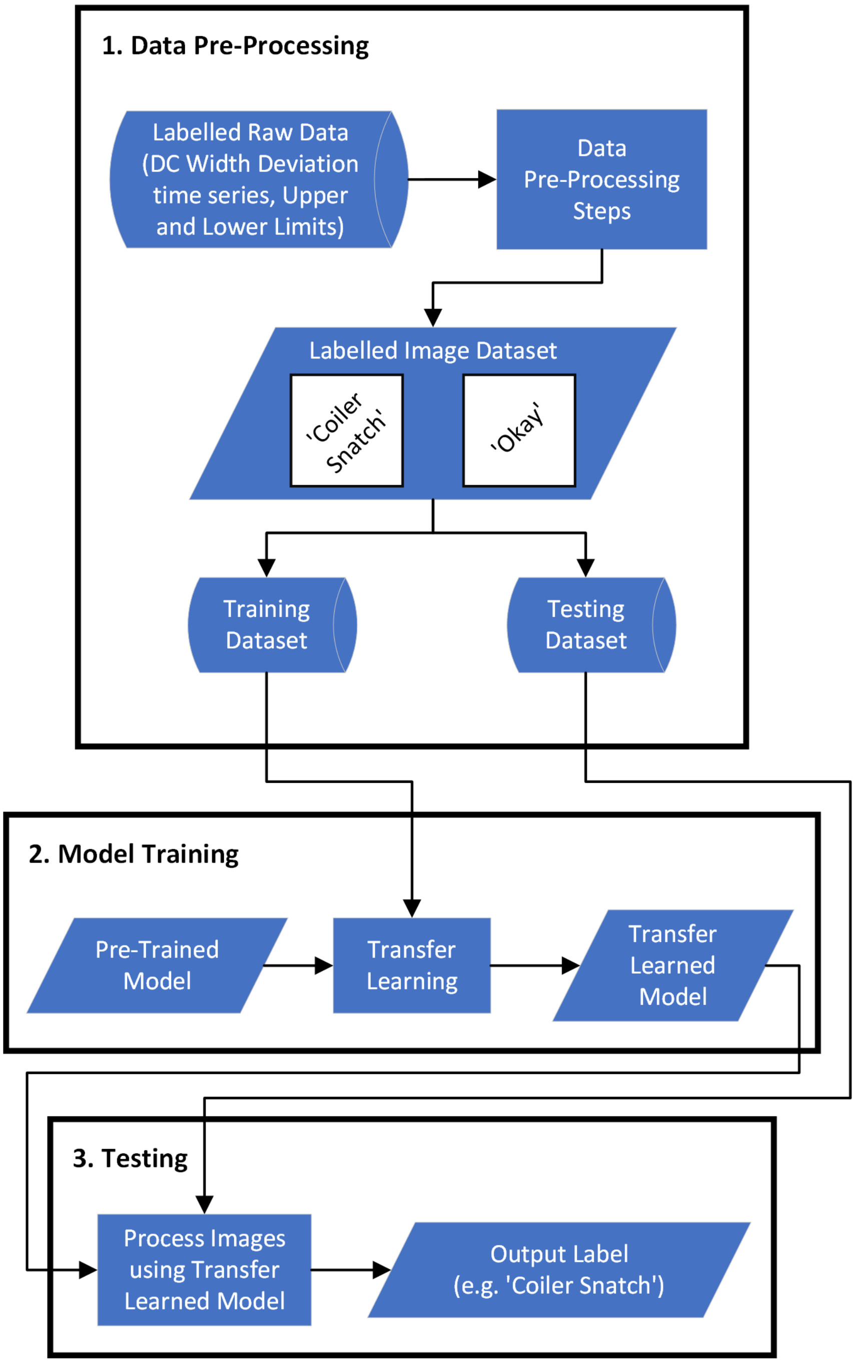


Figure 4.29: High-Level flowchart describing the steps taken to conduct the Coiler Snatch classification task.

4.5.1 Data Pre-Processing

The dataset used in this particular experiment contains 44 samples which display the behaviour of Coiler Snatch, and 44 samples which show normal behaviour. Table 11 illustrates the distribution of samples between labels as well as how samples for each label are split in order to perform 5-FCV.

Table 11: Number of samples belonging to each label for each fold in the Coiler Snatch classification task.

Cross Validation Fold	**Coiler Snatch**	**Okay**
Fold 1	9	9
Fold 2	9	9
Fold 3	9	9
Fold 4	9	9
Fold 5	8	8
Total (Label)	44	44

For this classification task, time series data consisting of a Coiler width deviation signal and two constant y-axis values representing upper and lower tolerances are used to represent each sample. Note that these tolerances are the same as those used in the Finishing Mill subprocess. Since Coiler Snatch occurs locally in the strip, the signal data for the entire strip is analysed in 10% segments using a sliding window and each window is considered as an individual classification problem. This can also help to deduce whether defects from previous subprocesses, such as Necking or Width Pull, have caused or contributed to Coiler Snatch depending on where it occured in the length of the strip. Limits of 40mm and -40mm can again be set on the y-axis of the strip to determine whether there is a fundamental issue with width or the process which requires further examination by analysts.

For this classification task, each sample is represented using an image constructed from the Coiler width deviation signal ‘C4_Width_WidthDeviation’ and two constant y-axis values representing upper and lower tolerances, gathered from the static data fields ‘pos_tol1’ and ‘neg_tol1’, respectively. Upper and lower limits are then set to remove anomalies and noise. These limits are set at the lower tolerance minus 20 and the upper tolerance plus 20, similarly to those in the Finishing Mill classification problem. Values outside of these limits indicate that there is a fundamental issue with width, or the process, which requires further examination by analysts. Redundant data where the strip is not present in the Coiler is also removed. Next, the time series is normalised to a range between 0 and 1, using formula 5, such that strip with different width aims and ranges are conventionally mapped on the same scale. In this case, the lower and upper limits are mapped to 0 and 1, respectively.

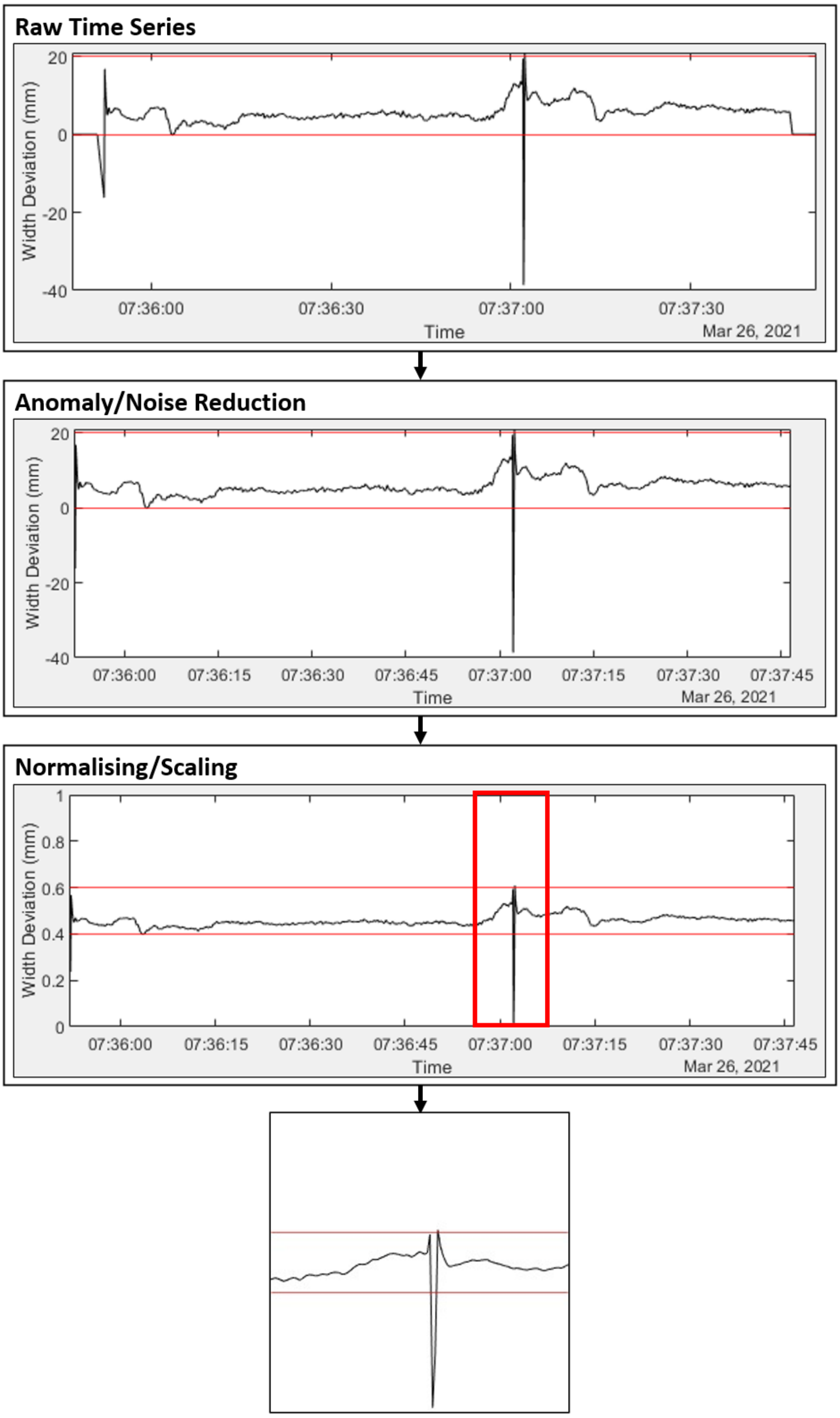
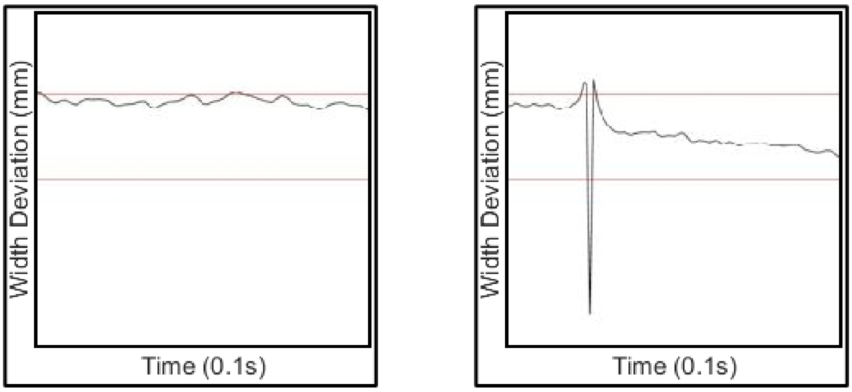


Figure 4.30: Steps for pre-processing Coiler time series data into images for the Coiler Snatch classification task.

Because Coiler Snatch can occur anywhere in the length of the strip, a sliding window is used in which segments of 10% of the strip, beginning with the head end, are considered as individual classification problems. The beginning point of this sliding window is incemented by 5% each time to ensure that there is no data missed by splitting an instance of a defect between images. It should be noted that while this has the potential to produce multiple images for the datasets used in the classification problem, only those portions of the strip in which Coiler Snatch is present have been used to represent the class in the image dataset. For example, if Coiler Snatch has only occurred once throughout the bar, only one image which represents this will be added to the Coiler Snatch class. An example of this process being carried out on the time series data for a bar labelled as having Coiler Snatch is shown in Figure 4.30.

The resulting figures are 224 pixel x 224 pixel x 3 colour channel image representations of the remaining, and relevant, time series data and the two plotted constant y-axis values representing the upper and lower width tolerances specified for each strip, which constitute the final dataset used in the image classification task in this experiment. An example image from each of the Okay and Coiler Snatch classes is in Figure 4.31 from left to right, respectively. As discussed in the opening section of this chapter, Coiler Snatch can be identified by a sudden decrease in width. Normal behaviour shows steady performance throughout a strip.



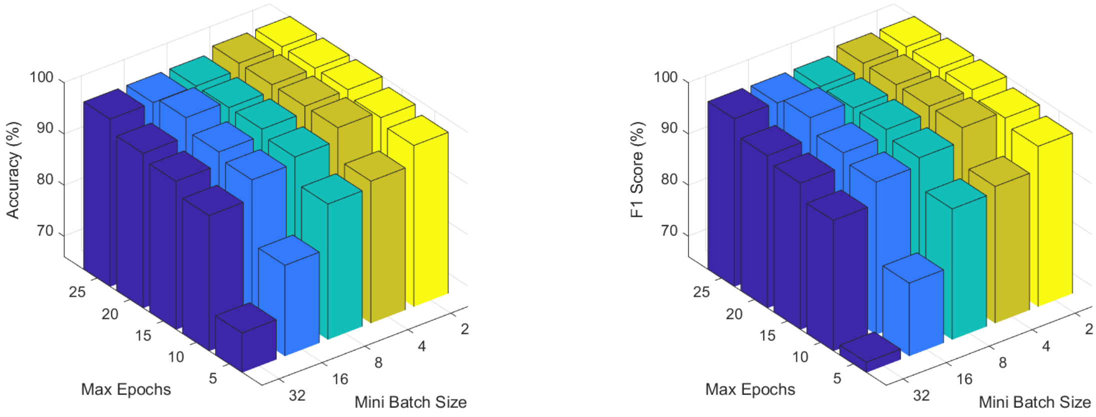
(a) A sample showing no signs of defective behavior.

(b) A sample showing behaviours of Coiler Snatch.

Figure 4.31: Examples of images contained in the dataset used for the Coiler Snatch classification task.

4.5.2 Modelling and Analysis of Results

Figure 4.32 shows the average testing performance of all folds for each hyperparameter value for the Coiler Snatch classification task. Within Figure 4.32, (a) shows accuracy while (b) shows F1 Score.

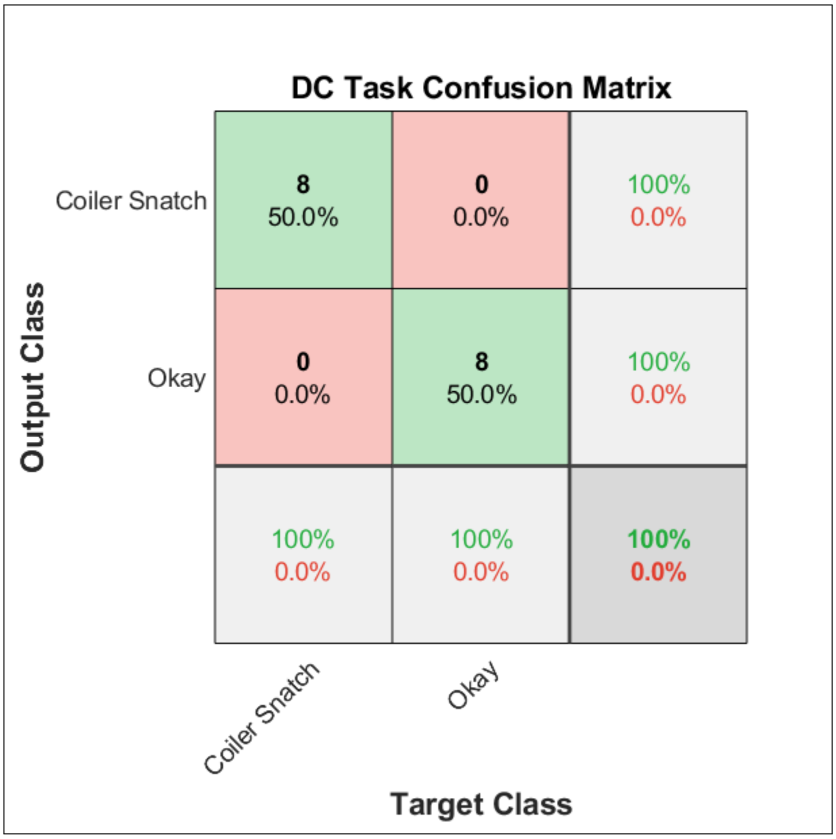


(a) Average accuracy.

(b) Average F1 Score.

Figure 4.32: Average 5-FCV performance of each hyperparameter value combination for GSO in the Coiler Snatch classification task.

Figure 4.32 and Table 12 show that higher Max Epochs values generally show the best overall performance. This may be because the full dataset used in this classification task is relatively small, meaning that lower Max Epochs values and, therefore, shorter training times, may not allow the algorithm to learn for long enough to produce a sufficient model. Performance is again shown to increase as Mini Batch Size value decreases. This may also be due to the small size of the dataset, as a smaller Mini Batch Size would allow the algorithm to learn gradually from smaller proportions of the dataset, as opposed to a larger batch size which may allow the algorithm to learn too quickly from too many different samples at once. As shown by the results of the previous two experiments, high Max Epochs values may subject models to overfitting when using a limited number of training samples. Taking this into account, a Max Epochs value of 20 and a Mini Batch Size of 2 have therefore been selected as the final hyperparameter values for the final Coiler Snatch image classification model to ensure that one of the best performing hyperparameter value combinations is still selected while mitigating the possibility of overfitting. Table 12 shows that the average accuracy and F1 Score for the 5-FCV splits using these hyperparameter values are 100%. Figure 4.33 shows a confusion matrix which shows the classification results of a randomly selected fold from this experiment using the selected hyperparameter values.



DC Task Confusion Matrix

Output Class \ Target Class	Coiler Snatch	Okay	
Coiler Snatch	**8** 50.0%	**0** 0.0%	100% 0.0%
Okay	**0** 0.0%	**8** 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	**100%** **0.0%**

Output Class
Target Class

Figure 4.33: Classification results of a randomly selected fold using the selected hyperparameter values for the Coiler Snatch classification task.

While this model produced no incorrect classifications, the limited availability of data in this particular experiment suggests that future work should aim to collect further images of Coiler Snatch instances once more data becomes available. This would allow for further training, testing, and evaluation of the model, and most importantly, further confirmation of its reliability in this classification task.

Table 12: Averaged 5-FCV results for GSO in the Coiler Snatch classification task.

		Max Epochs									
		5		10		15		20		25	
		Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
Mini Batch Size	2	98.75	98.82	98.75	98.67	98.75	98.67	100.00	100.00	100.00	100.00
	4	96.25	95.81	100.00	100.00	98.75	98.67	100.00	100.00	100.00	100.00
	8	95.00	94.63	97.50	97.33	98.75	98.67	100.00	100.00	100.00	100.00
	16	92.50	91.55	96.25	95.81	97.50	97.33	100.00	100.00	98.75	98.67
	32	73.75	68.03	83.75	80.28	92.50	91.55	93.75	92.72	97.50	97.33

4.6 Summary and Conclusions

In this chapter, the wide array of width defects which can occur throughout the HSM process was identified and discussed. By doing this, a deeper understanding of their origins and causes, as well as how they are currently identified and addressed, has been gained. From this, rules have been derived to identify some of these defects, which are outlined in Chapter 6. For those defects requiring the analysis of ambiguous data where simple conditional rules cannot be applied, ML models have been created to identify them and are also utilised in the web tool described in Chapter 6 [1]. Specifically, three models were created: One to identify instances of Necking and Flare in the Roughing Mill subprocess, one to identify instances of Width Pull in the Finishing Mill subprocess, and one to identify instances of Coiler Snatch in the Coiler subprocess. Prior to creating these models, the process of Data Collection from HSM data sources for static data from SQL databases and times series data from IBA Analyzer was carried out, creating a local data structure in MATLAB. The chosen ML technique was DL using GoogLeNet, a Pre-Trained CNN architecture, for image classification. For each model, the pre-processing steps for creating the image datasets required to train them were described. Each model was trained and tested using a combination of 5-FCV and GSO using a range of hyperparameter values. Each model showed optimal performance and will be combined with rules derived from expert knowledge, to create the data-driven web tool proposed in Chapter 6 which will ultimately have the capability of detecting width defects and their potential root causes, which are described in Chapter 5.

By gaining a deep understanding of HSM process and through research related literature, it was possible to plan and establish the methodologies and techniques used to assess each defect. As discussed in Section 3.2, there are many techniques for processing and analysing time series data based on the nature and properties of the time series data. By gaining the expert knowledge required to identify instances of width-related defects, it also became possible to identify an appropriate method for identifying defects in ambiguous time series data, that being the classification of times series as images with limited pre-processing [22,131,132,161], due to the time series data's random nature, using a Pre-Trained CNN architecture. Future work may include further exploration into other methods of time series analysis that are capable of processing random time series reliably [166].

Following the methodology for Data Collection from HSM data sources and creating a local data structure in MATLAB enables simple data processing and modelling in the MATLAB environment. It also provides a foundation on which to process new, unseen data for testing at a later date, including the data used to produce new testing examples to demonstrate models' functionality within the web tool in Chapter 6. The same methodology could also be adjusted in future works should more data fields be required for purposes such as training new models or visualising further data within the web tool.

As shown in Sections 4.3.2, 4.4.2, and 4.5.2, each model performs sufficiently for the intended purpose, especially considering the limited size of the available datasets. It is still, however, recommended to continue collecting data and retraining each model to reassess and monitor their reliability. Typically, ML models are trained on hundreds of thousands of images [8, 136], but this depends on the data available for the intended application. This is still a prominent challenge in industry and domain-specific applications [22, 131, 132] which has been addressed through the use of Transfer Learning, hyperparamenter optimisation, and K-FCV.

Despite their generally good performance, there were some misclassifications made by the Roughing Mill Necking and Flare and the Finishing Mill Width Pull classification models. The potential causes of these were investigated further with a team of analysts and mostly derived from the main challenges associated with manufacturing and domain-specific ML applications: limited data availability, data ambiguity, and possible human error. Potential solutions to these issues were discussed and, in conclusion, highlighted the benefit of further training and testing these models once more data becomes available to both boost and confirm their performance and reliability.

The models created in this chapter are intended as a step towards further utilisation of ML in steel manufacturing processes in which this technology is seldom used. This chapter showcases the ability to automate manual and time-consuming analyses of ambiguous data using ML technologies, and aims to show that there are a number of methods available for this based on the nature and availability of data. By combing these models with those in Chapter 5, and with the web tool described in Chapter 6, the chapter aims to show that it is possible to create end-to-end tools which enable the detection of defects and the identification of their root causes throughout an entire process, particularly in steel manufacturing processes. Future work would also aim to take this a step further and integrate system feedback directly into the process while also including a broader range of defects, and thus analyses, in the HSM process.

As mentioned above, although the chosen time series classification methodology was determined to be the most suitable for the intended purpose, other methods which allow efficient processing, modelling, and analysis of time series data of a random nature [161] could be used to train and test models for the same, or similar, purposes.

In the following chapter, the same methodology is applied to the root causes of the width-related defects described in the first section of this chapter to derive another set of rules from expert knowledge and to create further ML models from the local data structure created in the second section of this chapter.

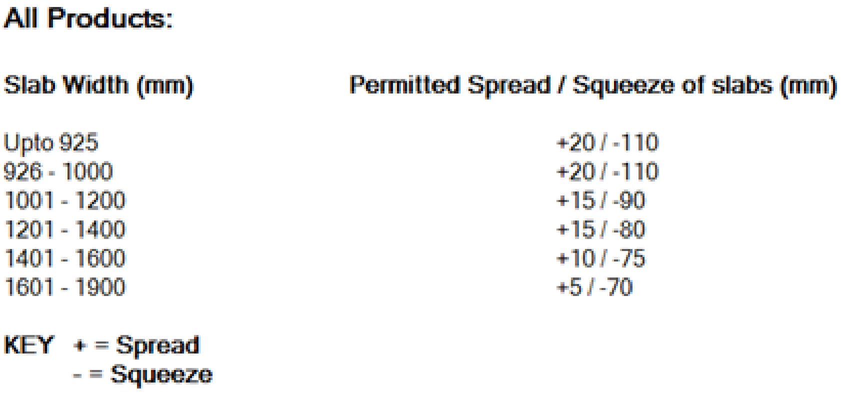
5 Determining Root Causes of Width-Related Defects in the Hot Strip Mill using Machine Learning and Expert Knowledge

This chapter focuses on the potential root causes of those width-related defects described in the previous chapter and, as also outlined in Chapter 4, adheres to the BDA framework described in Section 2.1.2, and the specific methodology to follow this framework described in Chapter 3. The purpose of this chapter is to understand the behaviour of these root causes, and to create and evaluate ML models for those which cannot be identified using simple tolerances and thus conditional checks. The procedure used to collect the data used in these experiments has also been described in detail in the previous chapter, resulting in a table structure within the MATLAB environment containing data ready to be retrieved and processed for training and testing samples. In Section 5.1, an overview is provided for the potential root causes throughout the HSM process which can lead to the discussed width-related defects, the factors that may lead to such root causes occurring in the first place, and how they are currently identified and addressed by operators and analysts. In Section 5.2, the steps used to follow the BDA framework and chosen methodology are reviewed. Justifications are made for the chosen technologies with regards to the ML models created within this chapter. Specifically, this section describes how data is chosen and extracted from its sources, how datasets for the chosen ML models are trained and validated. Sections 5.3.1 and 5.4.1 constitute the Data Management stages of each experiment, following from the initial Data Collection and Management performed in Section 4.2.1. They describe how data is processed and transformed in MATLAB in order to produce the required datasets. In Sections 5.3.2 and 5.4.2, the Modelling and Results and Analysis stages are carried out using selected hyperparameter optimisation and Cross Validation techniques. The findings and results of this chapter form the published conference papers titled ‘Pre-Trained CNN for Classification of Time Series Images of Anti-Necking Control in a Hot Strip Mill’ [22] and ‘Root Cause Classification of Temperature-related Failure Modes in a Hot Strip Mill’ [23], mentioned in Section 1.4.1.

5.1 Root Causes of Width-Related Defects in the Hot Strip Mill Process

5.1.1 Scheduling and Furnaces

There are many factors which can contribute towards width-related defects and they can originate both prior to and during the HSM process. When a bar is scheduled for the HSM process, preliminary checks can be made to determine whether it is suitable for rolling to the customers specification. Firstly, and most importantly, it is determined whether the correct bar width has been used by comparing it to a set of absolute Spread/Squeeze rules, shown in Figure 5.1. These rules outline a set of standard slab widths that can be used to roll a given set of strip widths. Spread is used when only slabs that are narrower than the customer specification width are in stock. Alternatively, squeeze is used when the slab is wider than the required specification. It is preferable and much more common to squeeze since it is easier to control width in the Roughing Mill due to the bar being in contact with edgers from the beginning of the subprocess, and also carries less risk of unnecessarily losing width. If a bar is thinner than expected and needs to be spread, there is no guarantee that it will make contact with the edgers in the roughing mill or that it will spread to the required width during the HSM process. While a rare occurrence, if the final width specification does not belong to the set given by the Spread/Squeeze rules, this is usually seen as the main cause of a width-related defect. It should be noted that the terms bar and slab are interchangeable.



All Products:

Slab Width (mm)	Permitted Spread / Squeeze of slabs (mm)
Upto 925	+20 / -110
926 - 1000	+20 / -110
1001 - 1200	+15 / -90
1201 - 1400	+15 / -80
1401 - 1600	+10 / -75
1601 - 1900	+5 / -70

KEY + = Spread
- = Squeeze

Figure 5.1: The standard set of Spread/Squeeze rules.

While not a direct cause of width of width-related defects, the chemical composition, or grade, of the bar may contribute to poor width performance in the HSM [22,158]. A number of steel grades have high silicon content to increase material strength. This, however, can make a bar less malleable, and thus more difficult to roll, increasing the risk of Full Coil Over Width. Alternatively, but much more rarely, this can also increase the risk of Under Width from excessive roller force and high temperatures used in an attempt to account for decreased malleability.

Malleability issues due to temperature-related causes can also arise from the furnace reheating stage of the HSM process. The HSM uses two furnaces to heat multiple bars at once such that there is no delay between a bar exiting the Coiler and another entering the Roughing Mill. Due to maintenance, it is sometimes necessary to shut down one furnace and continue operation with only one. This is referred to as Single Furnace Operation, and it can result in bars being processed too quickly in order to maintain production speed. This can result in bars not being thoroughly heated, referred to as Undersoaking, and therefore being less malleable. It is also possible for bars to be delayed between the furnace and Roughing Mill whether or not the HSM uses Single Furnace Operation. Delay occurs when furnace reheating is complete but another bar or strip is still present in subsequent HSM subprocesses. During this period, the delayed bar begins to cool down and become less malleable. Temperature is monitored throughout the HSM process and, depending upon how hot bar is, or how quickly and evenly it cools, may be the cause of those width-related defects described in Chapter 4.

5.1.2 Roughing Mill Subprocess

As discussed in Section 4.1.1, the Roughing Mill is the only subprocess in the HSM which directly adjusts width. The bar is passed back and forth through the Roughing Mill between five and eleven times and its sides are rolled using vertical rollers until it reaches a required width. A Roughing Mill model is used to predict the final width of the steel strip at the end of the HSM process by taking further rolling and cooling shrinkage into account, and make automatic adjustments to roller positions and thus force within the Roughing Mill. An error for this model is calculated once the strip has finished the HSM process. An error greater than 5mm or less than -5mm is considered too great as, although the models aims to compensate for width and temperature issues, it can contribute towards Necking and Flare and may result in Under Width or Over Width in subsequent subprocess depending upon the offset direction.

The movements of Roughing Mill edger rolls are controlled by two mechanical functions; screws and capsules. Screws move the edgers within approximately 1000mm of the bar while capsules hydraulically move the edgers to a more precise distance. Simply, screws are moved over larger distances when the bar is not being passed through the Roughing Mill while capsules can be constantly adjusted while a bar is being rolled. Each edger uses its own screw and capsule. Note that on each pass, the bar is only rolled by the first edger it meets on each pass while the other becomes passive; the bar is rolled by the first edger on the first pass, the second edger on the second pass, the first edger on the third pass, and so forth. With regards to this, the most probable causes of Necking in the Roughing Mill are related to ANC, an automated edger movement that aims to reduce or eliminate Necking by applying extra force to the head, tail, or both ends of the bar [22]. The issues associated with ANC which can cause Necking are the movement, or stroke, being too early or too late. If the ANC stroke is too quick, it will encourage further Necking by edging the sides of the bar too early. If the ANC stroke is too late, it will miss the end of the bar and encourage Flare. Early or late strokes may be caused by sensor irises being dirty or by steel being prematurely detected in the mill, meaning a false presence is detected and ANC begins too early, or the bar is difficult to see and is detected too late. If the magnitude of an ANC stroke is too high, the bar will be edged narrower than expected when the ANC is applied. Alternatively, ANC issues can also occur when the capsules are deselected entirely when an operator decides that there is less risk of Necking and Flare occurring in the Roughing Mill. For example, silicon-rich grades are less likely to Neck or Flare since they are generally less malleable than other grades. Examples of signals showing normal and early ANC stroke timings are shown in figures 5.2 and 5.3, respectively. Within these figures, the x-axis represents time while the two y-axis represent width capsule movement and average edger force.

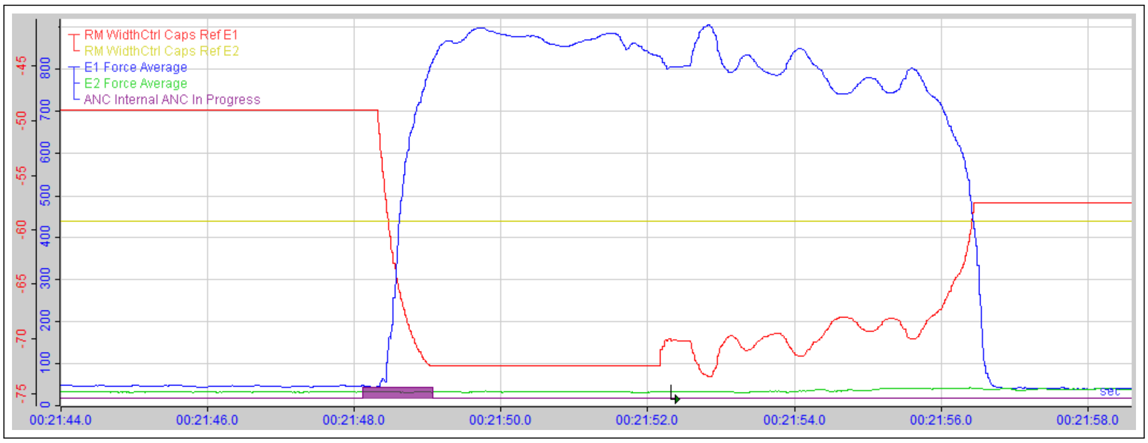


Figure 5.2: Signals showing capsule movement and edger force when an ANC stroke is well-timed.

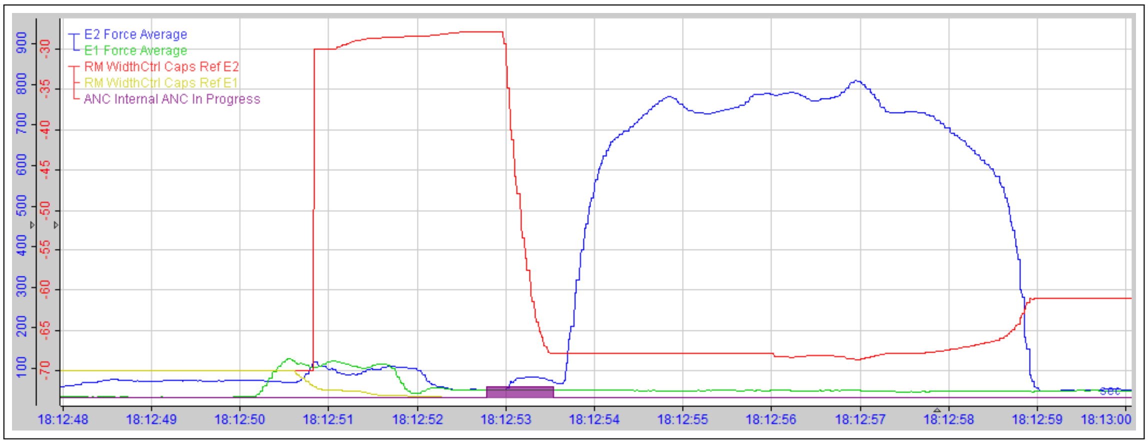


Figure 5.3: Signals showing capsule movement and edger force when an ANC stroke is early.

5.1.3 Finishing Mill Subprocess

Necking and Flare in the Roughing Mill can result in Width Pull and Flare in the Finishing Mill, respectively. Width Pull, however, may originate in the Finishing Mill depending upon the temperature of the strip within this subprocess and the Finishing Mill width model error. Both high and low strip temperatures can result in Width Pull. Strips with low temperatures are less malleable, and therefore more brittle, they are more likely to pull and tear under tension while rolling. Analogously, strips with high temperatures are more malleable, and are more likely to elongate under tension while rolling. While the purpose of the Finishing Mill to elongate the strip while rolling it to a required thickness, too much elongation can result in Width Pull or Full Coil Under Width. The Finishing Mill width model performs similarly to the Roughing Mill width model, although adjustments are made to rollers within the Finishing Mill. A model error of more than 5mm or less than -5mm is also considered too great for this model and can lead to Under Width or Over Width depending upon the offset direction. Examples of temperature signals showing behaviour of low and high temperatures are shown in figures 5.4 and 5.5, respectively. Within these figures, the x-axis represents time while the y-axis represents temperature.

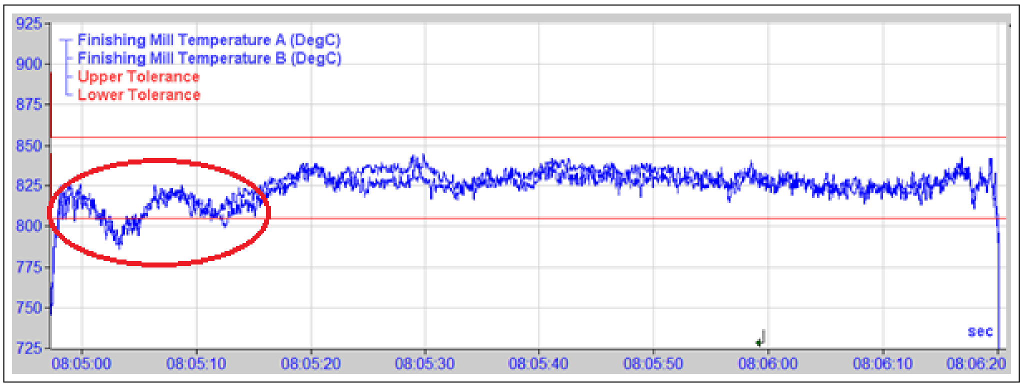


Figure 5.4: Temperature signal of a strip with low temperature at its Head End in the Finishing Mill.

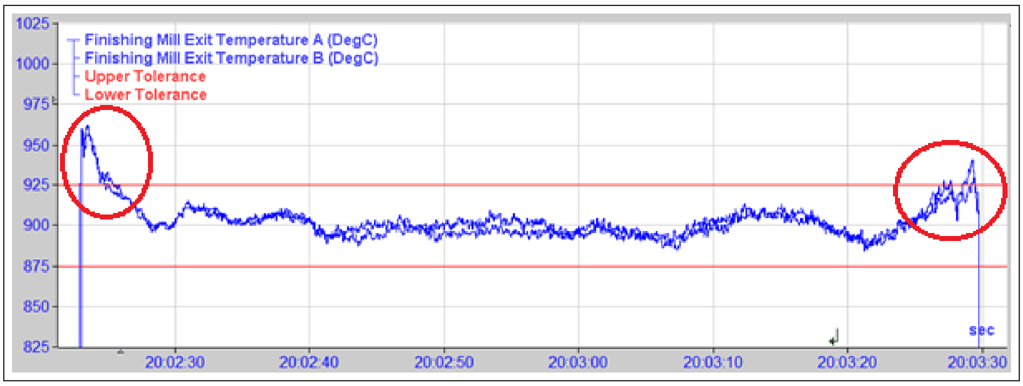


Figure 5.5: Temperature signal of a strip with high temperature at its Head and Tail Ends in the Finishing Mill.

5.1.4 Coiler Subprocess

Defects and root causes found in the Coiler subprocess may also present themselves as a result of defects and root causes originating in previous subprocesses. This includes Flare, Under Width, and Over Width. Coiler Snatch presents similarly to Width Pull and may therefore originate in the Finishing Mill. However, true Coiler Snatch originates in the Coiler and is also usually a result of either high or low strip temperatures within this subprocess. As previously mentioned, higher temperatures can make the strip more malleable, making it more vulnerable to further elongation and, thus, thinning. The most common cause of Coiler Snatch, however, is low strip temperature following cooling the runout table subprocess. As the strip becomes colder, the harder and less brittle it becomes, increasing the possibility of sudden pulls, or snatches, under tension in the Coiler. The possible root causes of width defects in the HSM and in which subprocess they may occur, are shown below in Figure 5.6 and summarised in Table 13. The defects that these root causes may lead to are also shown in Table 13 and may be reviewed in Section 4.1, specifically in Table 3.

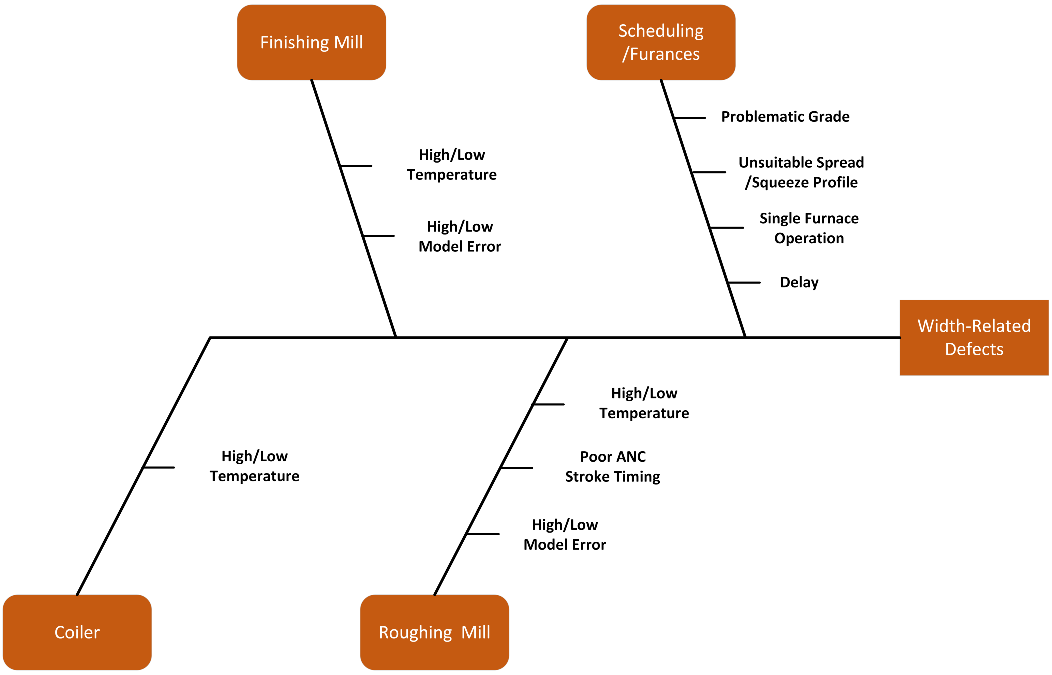


Figure 5.6: Fish bone diagram showing possible root causes of width-related defects in the HSM [1].

Table 13: Summary of possible root causes of width-related defects that can be found in the HSM process.

Root Cause	**Description**	**How it's currently found and addressed**	**Potential defects that could result from this**
		Scheduling and Furnaces	
Problematic Grade	Chemical composition of the bar makes it difficult to roll, namely those with high silicon content.	Check bar grade and consider whether to adjust offset based on this. If engineer offset is made, this may affect successive bars depending on their chemical	Full Coil Under Width, Width Pull, Coiler Snatch

		composition.	
Width Alarm Triggered (Spread-/Squeeze rules not met)	Bar used for rolling does not follow Spread-/Squeeze rules for ordered dimensions thus triggering a width alarm.	Check if width alarm is triggered and decide whether to apply offset to counter model updates which could negatively impact successive bars. Check specification of next bar against ordered bar to make sure it is suitable for final product.	Necking, Flare, Full Coil Under Width, Full Coil Over Width, Width Pull, Coiler Snatch
Furnace Delay	Bar on standby, or delayed, after furnace reheating due to bottlenecks in subsequent subprocesses, allowing for time to cool down.	Check if delay alarm is triggered and potentially delay bars before entering furnace. Also investigate why bottleneck occurs in the first place. Possibly move to Single Furnace Operation if bottleneck persists.	Flare, Full Coil Under Width, Width Pull, Coiler Snatch
Single Furnace Operation	Only one furnace in operation, sometimes meaning that if the process needs to be sped up to meet demand, bars may not be heated enough.	Check if Single Furnace Operation alarm is triggered and check if other furnace can be put back in use. Slow production down if bars are undersoaked.	Flare, Full Coil Under Width, Width Pull, Coiler Snatch

	Roughing Mill		
Roughing Mill Model Error	Roughing Mill width model error is >5mm or <-5mm indicating a fundamental issue with the bar or another width-related failure mode has occurred, hence the large model error.	Check Roughing Mill model error and other failure modes such as delay and width alarm. Adjust engineer offset based on findings.	Necking, Flare, Full Coil Under Width, Full Coil Over Width
ANC Stroke Early/Disabled	ANC stroke capsule movement either triggered early or disabled.	Visual inspection of capsule movement, roller force, and ANC activation time series. Reactivate if possible.	Necking, Flare, Full Coil Over Width
	Finishing Mill		
Finishing Mill Model Error	Finishing Mill width model error is >5mm or <-5mm indicating a fundamental issue with the bar or another width-related failure mode has occurred, hence the large model error.	Check Finishing Mill model error and other failure modes such as delay and width alarm. Adjust engineer offset based on findings.	Full Coil Under Width, Full Coil Over Width, Width Pull, Coiler Snatch
High Temperature	Values or deviation of temperature in the Finishing Mill too high, resulting in increased malleability.	Visual inspection of Finishing Mill temperature time series.	Full Coil Over Width, Width Pull, Coiler Snatch
Low Temperature	Values or deviation of temperature in the Finishing Mill too low,	Visual inspection of Finishing Mill temperature time series.	Full Coil Under Width, Width Pull, Coiler Snatch

	resulting in reduced malleability.		
	Coiler		
Low Temperature	Values or deviation of temperature in the Finishing Mill too high, resulting in increased malleability.	Visual inspection of Coiler temperature time series.	Coiler Snatch
Low Temperature	Values or deviation of temperature in the Finishing Mill too low, resulting in reduced malleability.	Visual inspection of Coiler temperature time series.	Coiler Snatch

In this chapter, a DL model for the classification of multivariate time series images of ANC stroke timing [22] has been created. Only Early and Okay classes are included in this task since late strokes are a very rare occurrence. This means that a limited data is available for this failure mode to such a degree that balancing classes for training and testing would mean a reduction in the amount of Early samples used in the experiment, which are already limited in number. This is discussed in further detail in the concluding section of this chapter. A traditional ML model to binarily classify between high or low temperatures using numeric features [23] has also been created. These models, in addition to those created in Chapter 4, are used in conjunction with expert knowledge to determine the presence of width-related defects and their potential root causes in the web tool presented in following chapter [1].

5.2 Methodology

5.2.1 Algorithms and Hyperparameter Optimisation

For the image classification tasks in this chapter, the pre-trained network GoogLeNet [8] has, again, been selected and its final two layers, fully connected and output, retrained with the image dataset of the ANC classification task. A Transfer Learning approach was chosen due to its ability to utilise knowledge from previous image training data and to apply it to a new classification task [89, 162, 163] and GoogLeNet was selected due to its depth and its use of inception modules which allow it to use a larger number of convolutional layers with less data loss [8, 90, 164].

To determine the optimal hyperparameters for the CNN created in this chapter, GSO [143] has been used. This is again due to its exhaustive method of providing the broadest possible evaluation bu testing every possible combination. In experiments carried out prior to those in this chapter and Chapter 4, a range of values were considered for Initial Learn Rate, Learn Rate Drop Factor and L2 Regularisation [22]. From this, it was concluded that these hyperparameters can had little impact on the training and testing performance of a neural network. It should be noted that this is not the case for all neural networks and datasets [142]. A small, but broad, range of Max Epochs values were also used to reduce total experiment time. The Cartesian product of Table 14, which contains a range of values for Max Epochs and Mini Batch Size is therefore used as the set of hyperparameter values used to train each CNN in this chapter.

Table 14: Hyperparameter values used for GSO in the experiments shown in the ANC stroke timing classification task.

Max Epochs	**Mini Batch Size**
5	2
10	4
15	8
20	16
25	32

5.2.2 Cross Validation

K-FCV [150], with a K value of 5, is used again for both ML experiments in this chapter to determine how well and how reliably each model classifies new, unseen data. As discussed in Section 4.2.2.2, K-FCV can help to identify and thus avoid overfitting in ML models due to its method of splitting data into multiple folds for use in multiple training runs. 5-FCV is performed in conjunction with GSO such that 5-FCV is performed for each hyperparameter value combination.

5.3 Model for Anti-Necking Control Stroke Timing in the Roughing Mill Subprocess

In this section, an image classification model is created to distinguish between Early and Okay ANC timing in the Roughing Mill subprocess. To achieve this, the same process described in Section 4.3 and Figure 4.17 has been applied. A dataset consisting of samples having Early ANC timing, and a corresponding number of samples with normal ANC timing are extracted from the dataset acquired in Section 4.2.1. These samples are pre-processed and used to create images which display the important visual elements of the processed time series data. Three images are created for each sample, representing individual passes in the RM subprocess, and each has its own corresponding label for the classification task. These images form the final dataset used to train CNN image classification models, which are then evaluated to determine the optimal configuration and, therefore, which model is to be used in the decision-making tool proposed in Chapter 6. This process is illustrated in Figure 5.7 and each step is described in detail in Sections 5.3.1 and 5.3.2.

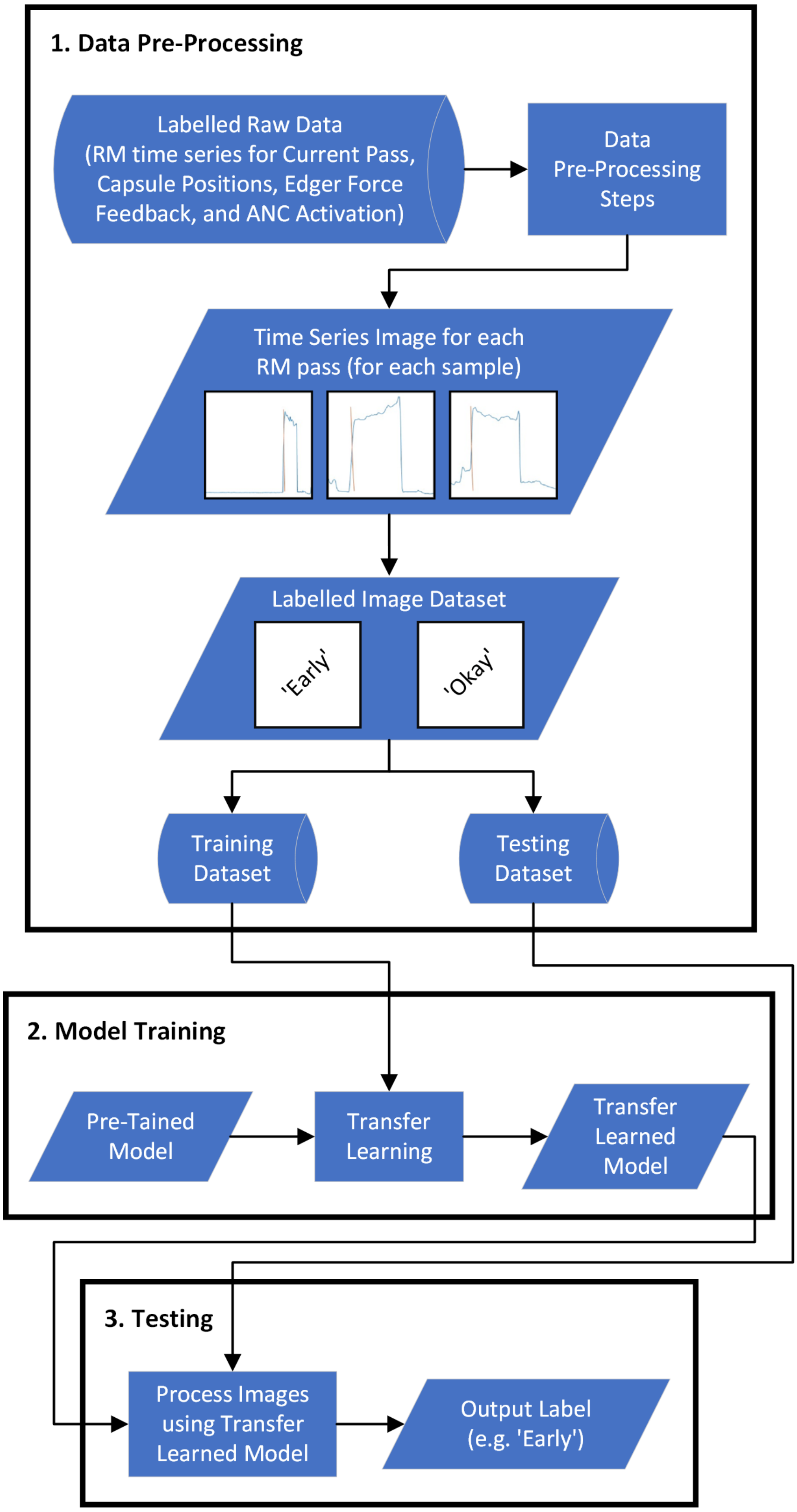

1. Data Pre-Processing
Labelled Raw Data (RM time series for Current Pass, Capsule Positions, Edger Force Feedback, and ANC Activation)
Data Pre-Processing Steps
Time Series Image for each RM pass (for each sample)
Labelled Image Dataset
'Early'
'Okay'
Training Dataset
Testing Dataset
2. Model Training
Pre-Tained Model
Transfer Learning
Transfer Learned Model
3. Testing
Process Images using Transfer Learned Model
Output Label (e.g. 'Early')

Figure 5.7: High-Level flowchart describing the steps taken to conduct the Roughing ANC stroke timing classification task.

5.3.1 Data and Pre-Processing

In this section, an image classification model is created for the purpose of distinguishing between Early and Okay ANC stroke timing in the Roughing Mill subprocess. The dataset used to train and test this model consists of samples which are labelled as having Early or Okay behaviour, and were compiled from real historical data from the Port Talbot HSM, as discussed in Section 3.1. The dataset used in this particular experiment contains 262 samples which display the behaviour of Early ANC stroke timing and 262 samples which show the behaviour of Okay ANC stroke timing. Table 15 illustrates the distribution of samples between labels as well as how samples for each label are split in order to perform 5-FCV.

Table 15: Number of samples belonging to each label for each fold in the Roughing Mill ANC stroke timing classification task.

Cross Validation Fold	**Early**	**Okay**
Fold 1	52	52
Fold 2	52	52
Fold 3	52	52
Fold 4	53	53
Fold 5	53	53
Total (Label)	262	262

For this classification task, time series data consisting of Roughing Mill edger force and capsule movement signals are used to represent each sample. While a binary metal detection signal is available to detect whether a bar is present in the Roughing Mill, this signal is not completely reliable for determining exact entry and exit points of the bar. This is because the signal can be early or late due to debris, or faulty or dirty sensors. This signal is also used to determine the activation time of the ANC itself, meaning that the classification of ANC timing via alternative signals can also help to determine whether there are further issues with metal detection in the Roughing Mill.

Since ANC occurs in the first three passes in the Roughing Mill, three individual classification samples can be created from a single bar. Roughing Mill edger force signals show a repeating pattern in which they sharply rise when the bar enters, and fall when the bar exits. While it is still not possible to determine the exact entry and exit points of the bar using this pattern, it still provides a descriptive image of ANC stroke timing when combined with the capsule movement signal. A binary ANC activation signal can also be used for cross-referencing to eliminate redundant data in the capsule movement signal. The final image for an individual pass is the full edger force signal in combination with the capsule position signal after cross-referencing with the binary ANC activation signal. An example of this process being carried out on the time series data for a bar labelled as having Early ANC stroke timing is shown in Figure 5.8.

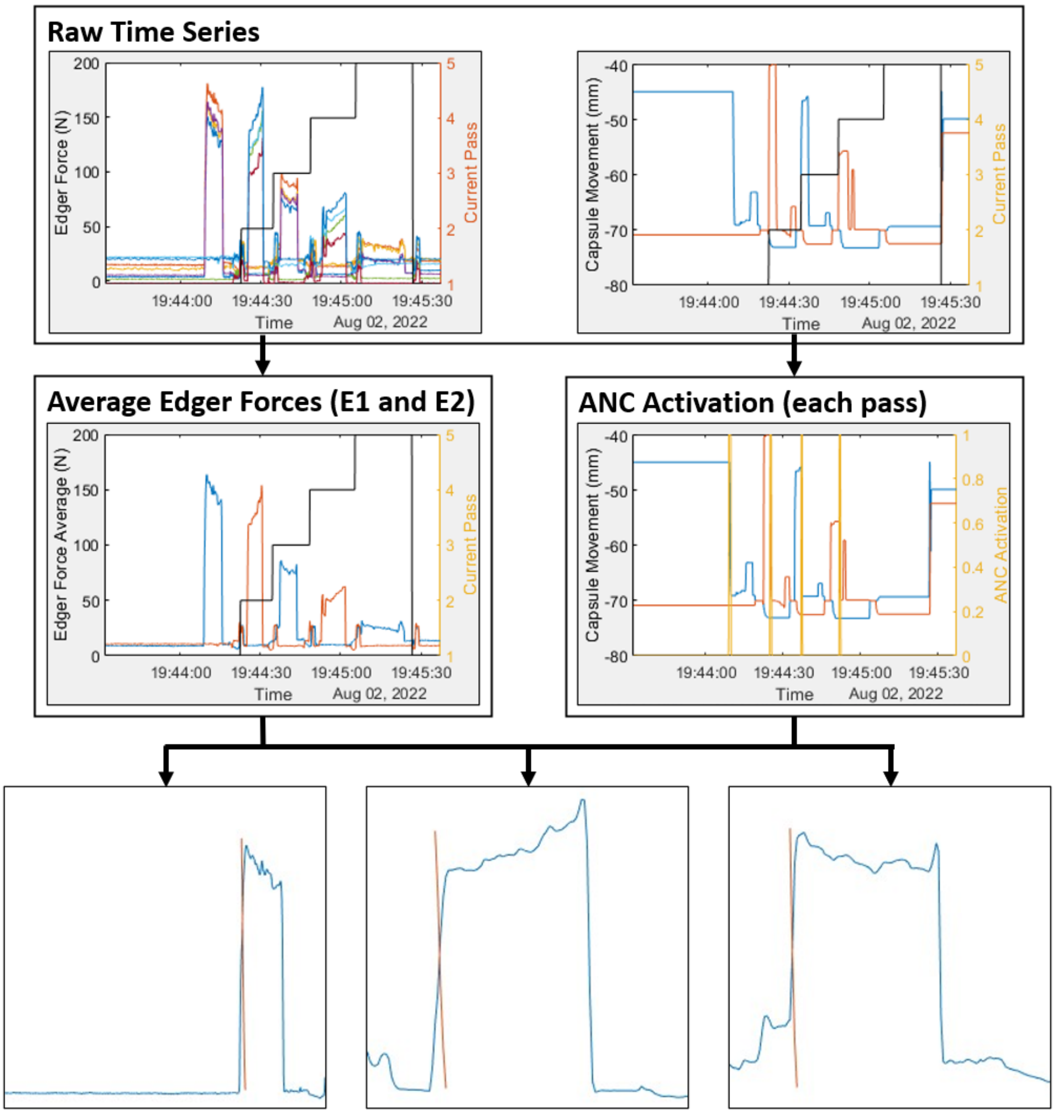
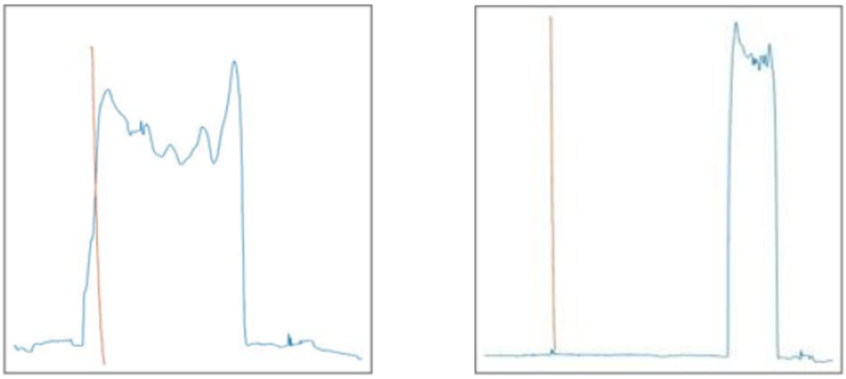


Figure 5.8: Steps for pre-processing Roughing Mill time series data into images for the ANC stroke timing classification task.

The resulting figures are 224 pixel x 224 pixel x 3 colour channel image representations of the remaining, and relevant, time series data, which constitutes the final dataset used in the image classification task in this experiment. An example image from each of the Okay and Early classes is shown in Figure 5.9 (a) and (b), respectively. As discussed in the opening section of this chapter, Early ANC strokes can be identified by capsule movement during ANC activation taking place before the large rise in edger force. Normal behaviour shows capsule movement during ANC activation taking place during the large rise in edger force. In the examples shown in Figure 5.9, the blue line represents the edger force signal while the orange line represents the capsule movement signal.



(a) A sample with acceptable ANC stroke timing.

(b) A sample with early ANC stroke timing.

Figure 5.9: Examples of images contained in the dataset used for the Roughing Mill ANC classification task.

Upon reviewing the finalised image dataset, an image not representative of conventional Early or Okay behaviour was found, shown in Figure 5.10. After retrieving the corresponding ID and date for this sample, the bar's data was reviewed with the aid of a HSM analyst to determine the true nature of its behaviour. After further analysis, it was determined that the ANC stroke was neither Early or Okay. Operators had noted several bars later that the sensors used to determine when to activate ANC were dirty and due for inspection. As described in Section 5.1.2, this caused the ANC to activate too late, shown in Figure 5.10 halfway through the bar's length by the expected increase in force length when the ANC activates. This bar, however, was not recorded as defective as its width profile was rolled within specification by the end of the Roughing Mill subprocess. The image sample was thus removed from the dataset and another instance of an Okay example was used to fairly represent the correct characteristics of the class and prevent the model from incorrectly learning from and generalising the Okay class. This instance highlights that it is useful to collect data as it occurs in individual contexts, particularly where domain-specific data is limited.

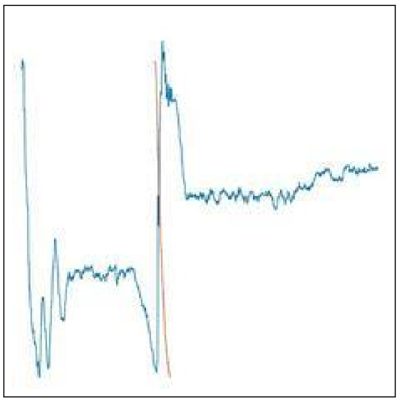
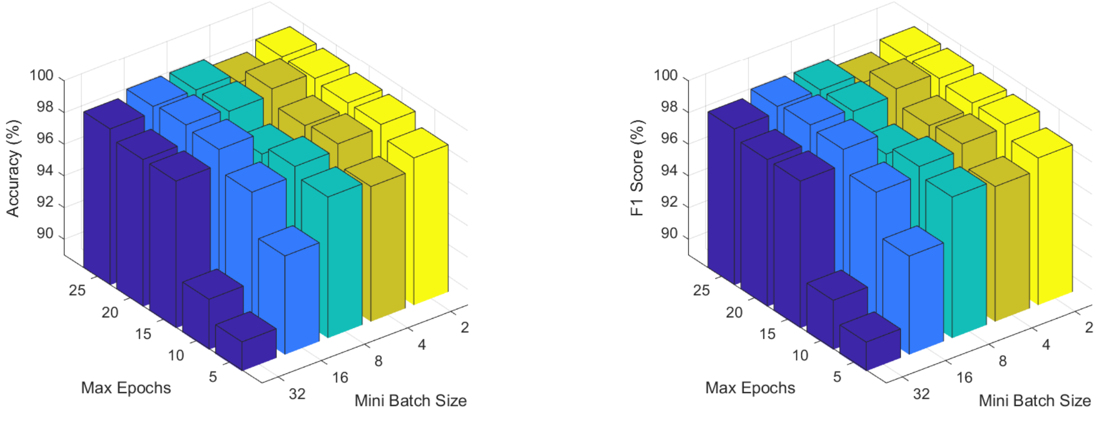


Figure 5.10: Signals showing an ANC stroke attempted when RM sensors are dirty.

5.3.2 Results and Conclusions

GSO is performed simultaneously with 5-FCV using the folds described in Table 15. Figure 5.11 shows the average testing performance of all folds for each hyperparameter value for the Roughing Mill ANC stroke timing classification task. Within Figure 5.11, (a) shows accuracy while (b) shows F1 Score which provides an insight into the precision and recall of the model.



(a) Average accuracy.

(b) Average F1 Score.

Figure 5.11: Average 5-FCV performance of each hyperparameter value combination for GSO in the Roughing Mill ANC stroke timing classification task.

Table 16: Averaged 5-FCV results for GSO in the Roughing Mill ANC stroke timing classification task.

		Max Epochs									
		5		10		15		20		25	
		Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
Mini Batch Size	2	98.86	98.86	99.24	99.24	99.23	99.23	98.66	98.66	99.23	99.23
	4	98.28	98.28	99.43	99.43	99.24	99.24	99.62	99.62	99.23	99.23
	8	98.28	98.28	99.24	99.24	97.9	97.9	98.66	98.66	99.05	99.05
	16	92.17	92.11	97.9	97.89	98.48	98.47	98.86	98.86	99.04	99.04
	32	90.84	90.83	95.23	95.22	97.91	97.91	97.53	97.53	98.29	98.29

Figure 5.11 and Table 16 show a general trend of model performance increasing along with Max Epochs. This may be because the model is allowed to learn for a longer period of time due to a higher number of iterations. It should, however, be noted that there is a slight decrease in performance when a Max Epochs value of 25 is used as opposed to a value of 20, which could be the result of overfitting during training. This can be confirmed by the results of using a Max Epochs value of 5, which shows the lowest performance across almost all Mini Batch Size values. The performance of each Mini Batch Size does not show a completely linear pattern across all Max Epoch values. However, a lower Mini Batch Size value becomes more favourable for this classification task as Max Epochs increases. Based on these findings, a Max Epochs value of 20 and a Mini Batch Size of 4 have therefore been selected as the final hyperparameter values for the final Roughing Mill ANC stroke timing image classification model. Table 16 also shows this to be the best overall performing hyperparameter value combination with both the average accuracy and F1 Score for the 5-FCV splits being 99.62%. Figure 5.12 shows a confusion matrix which shows the classification results of a randomly selected fold from this experiment using the selected hyperparameter values.

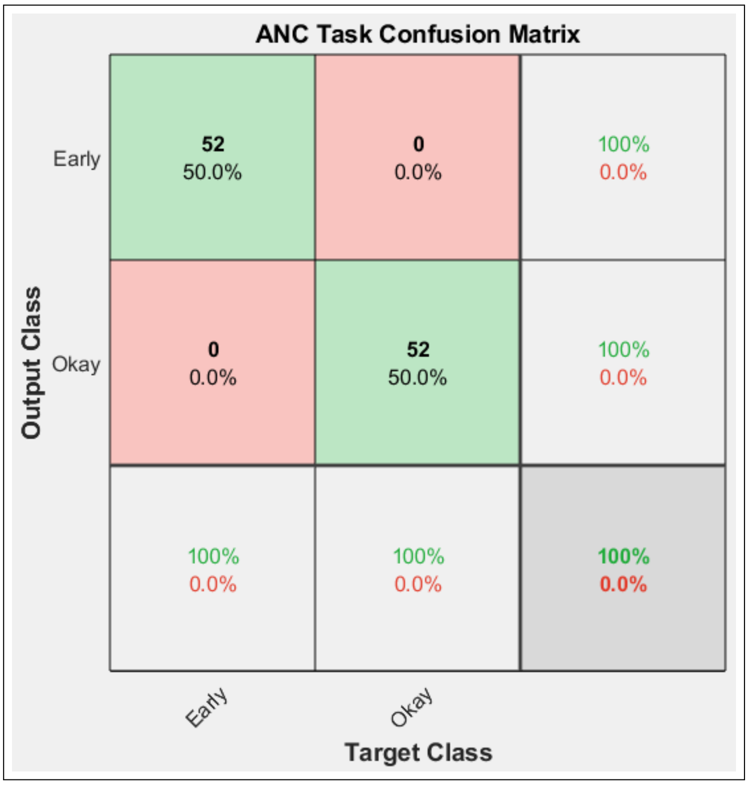


Figure 5.12: Classification results of a randomly selected fold using the selected hyperparameter values for the ANC stroke timing classification task.

Initial experiments for training and testing image classification models for ANC stroke timing were performed in the early stages of the thesis project for purpose of learning and establishing the workflow used throughout [22]. From these experiments, it was found that other hyperparameters, including Learn Rate Drop Factor and L2 Regularization had little impact on classification performance. Default values of 0.0001 and 0.3 are therefore used for L2 Regularization and Learn Rate Drop Factor, respectively, in all of the image classification models created in this chapter and Chapter 4.

While this model produced no incorrect classifications, the relatively limited availability of data suggests that future work should aim to collect further images of Early instances once more data becomes available. This would allow for further training, testing, and evaluation of the model, and most importantly, further confirmation of its reliability in this classification task. If enough samples become available, a Late ANC stroke timing label could also be added to the dataset before training and evaluating the model again.

5.4 Model for Temperature-Related Failure Modes in the Finishing Mill and Coiler Subprocesses

In this experiment, 11 traditional classification models are created to distinguish between high and low temperatures within the Finishing Mill and Coiler subprocess. As described in Section 5.1, temperature behaviour is inspected visually and manually by analysts by looking at the temperature's distribution both as a whole and on parts of the strip where defects have occurred, as opposed to identifying specific shape features as is the case in the previous experiments. Therefore, numerical features describing the distribution of the temperature values in time series data are used instead of images to train these traditional classification models.

To achieve this, a dataset consisting of samples labelled as having high and low temperatures are extracted from the dataset acquired in Section 4.2.1. These samples are preprocessed and a set of numerical features are derived from their temperature data. This set of features forms the final dataset used to train the various classification models, which are then evaluated to determine the optimal model for use in the decision-making tool proposed in Chapter 6. This process is illustrated in Figure 5.13 and each step is described in detail in Sections 5.4.1 and 5.4.2.

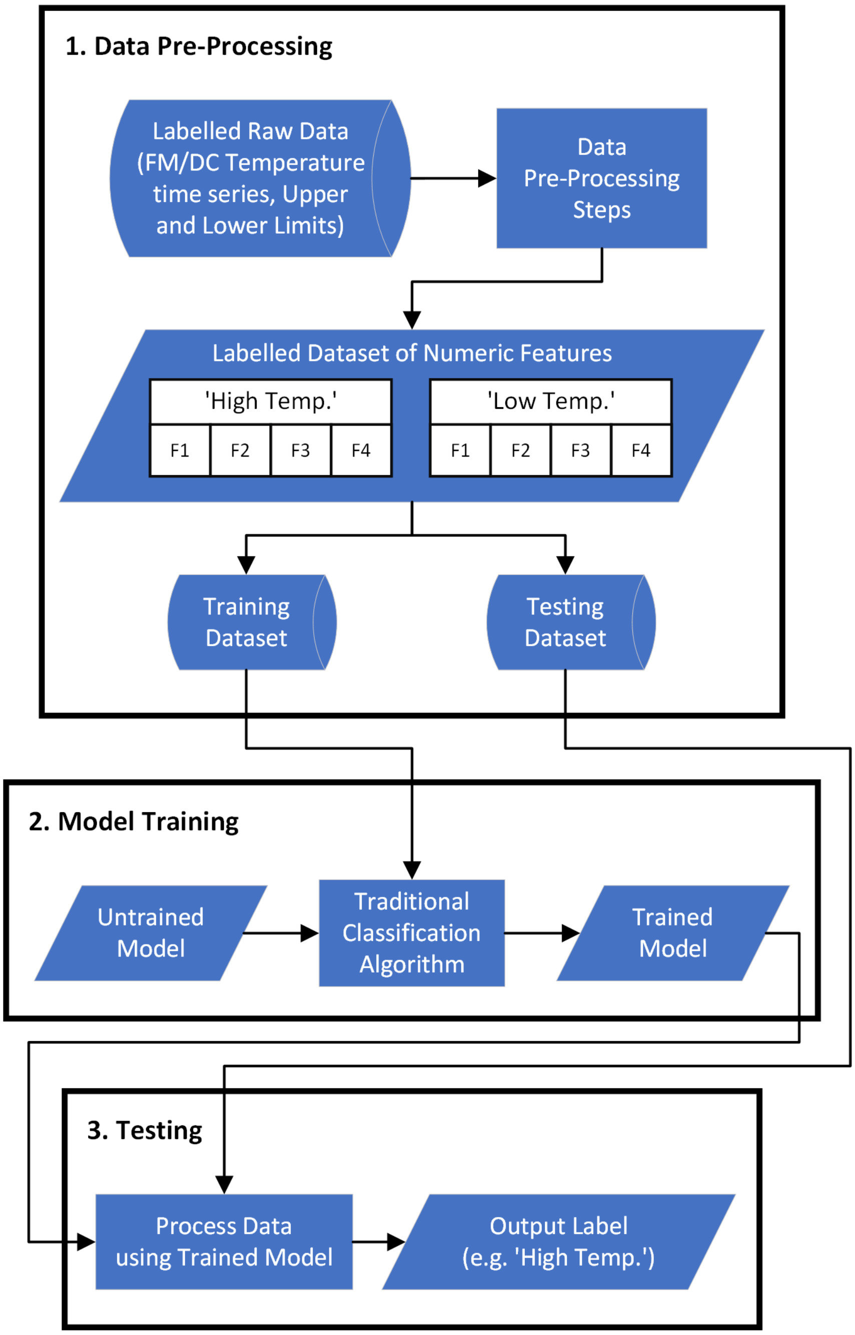

1. Data Pre-Processing
Labelled Raw Data (FM/DC Temperature time series, Upper and Lower Limits)
Data Pre-Processing Steps
Labelled Dataset of Numeric Features
'High Temp.'
F1
F2
F3
F4
'Low Temp.'
F1
F2
F3
F4
Training Dataset
Testing Dataset
2. Model Training
Untrained Model
Traditional Classification Algorithm
Trained Model
3. Testing
Process Data using Trained Model
Output Label (e.g. 'High Temp.')

Figure 5.13: High-Level flowchart describing the steps taken to conduct the Finishing Mill and Coiler Temperature classification task.

5.4.1 Data and Pre-Processing

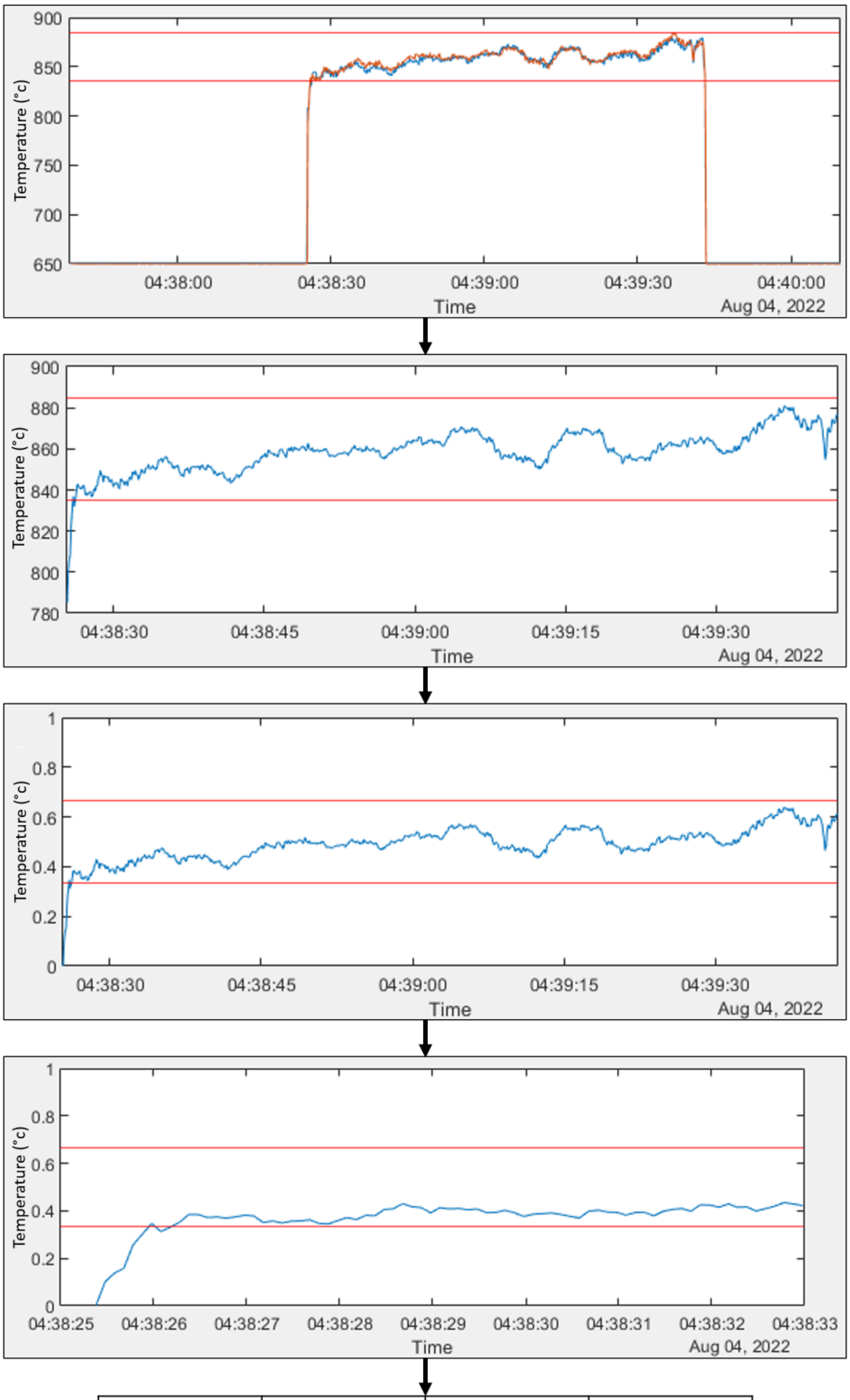
The dataset used in this particular experiment contains 55 samples which consist of features representative of high temperature, and 111 samples which consist of features representative of low temperature. Table 17 illustrates the distribution of samples between labels as well as how samples for each label are split in order to perform 5-FCV.

Table 17: Number of samples belonging to each label for each fold in the Finishing Mill and Coiler Temperature classification task.

Cross Validation Fold	**Low Temp.**	**High Temp.**
Fold 1	22	11
Fold 2	22	11
Fold 3	22	11
Fold 4	22	11
Fold 5	23	11
Total (Label)	111	55

For this classification task, time series data consisting of a Finishing Mill temperature signal and two constant y-axis values representing upper and lower tolerances are used to represent each sample. While metal detection signals cannot be used to accurately determine the entry and exit points of a strip in the HSM subprocesses, as mentioned in the previous section, it can be used to eliminate a majority of redundant measurements which occur before and after the relevant data. In this case, a metal detection signal for the final stand, or exit, of the Finishing Mill is applied to the temperature signals to retrieve the most relevant measurements.

If the temperature exceeds the upper tolerance, or falls below lower tolerance, for 50% of the strip, then a conditional check can be used to determine whether there is a fundamental temperature issue. These instances can be determined prior to the classification task and can save computation power and time. However, if the temperature of a given sample deviates by greater than 20°c, it is considered for the classification task. In this occurrence, a set of basic numerical features is extracted from the remaining time series data. A single sample is therefore comprised of a label, which is collected from historical label data, and the set of numerical features. An example of this process being carried out on the time series data for a strip labelled as having Low Temperature in its head end is shown in Figure 5.14.

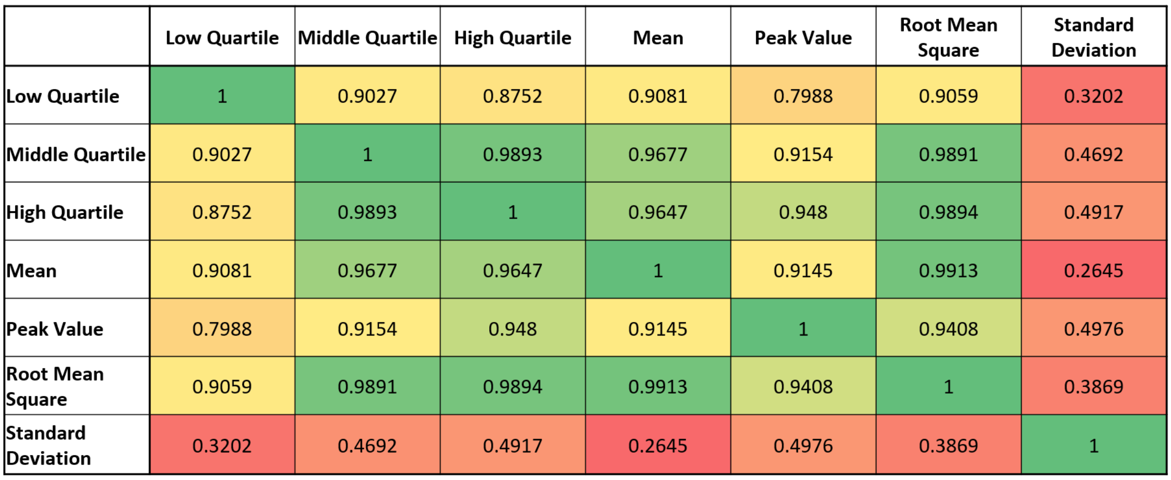


LQ	Mean	Peak Value	SD
0.3631	0.3708	0.4348	0.0723

Figure 5.14: Steps for pre-processing Finishing Mill and Coiler time series data into numeric features for Temperature classification task.

Pearson's Correlation Coefficient 6 is used to determine which features are most suitable and whether some features can be excluded due to overly-similar correlation with other features which would yield little to no benefit during training. The heatmap shown in Figure 5.15 described the correlation between each feature for all samples in the dataset. The resulting set of features consists of the mean, peak value, root mean square, and standard deviation. This is due to the high correlation between the middle quartile, high qaurtile, root mean square, and mean features, and the high correlation between the high quartile and peak value features. By eliminating the middle quartile, high quartile, and root mean square features, and retaining the mean and peak value features, information is retained and not duplicated during training, reducing the possibility of overfitting.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{6}$$



	Low Quartile	Middle Quartile	High Quartile	Mean	Peak Value	Root Mean Square	Standard Deviation
Low Quartile	1	0.9027	0.8752	0.9081	0.7988	0.9059	0.3202
Middle Quartile	0.9027	1	0.9893	0.9677	0.9154	0.9891	0.4692
High Quartile	0.8752	0.9893	1	0.9647	0.948	0.9894	0.4917
Mean	0.9081	0.9677	0.9647	1	0.9145	0.9913	0.2645
Peak Value	0.7988	0.9154	0.948	0.9145	1	0.9408	0.4976
Root Mean Square	0.9059	0.9891	0.9894	0.9913	0.9408	1	0.3869
Standard Deviation	0.3202	0.4692	0.4917	0.2645	0.4976	0.3869	1

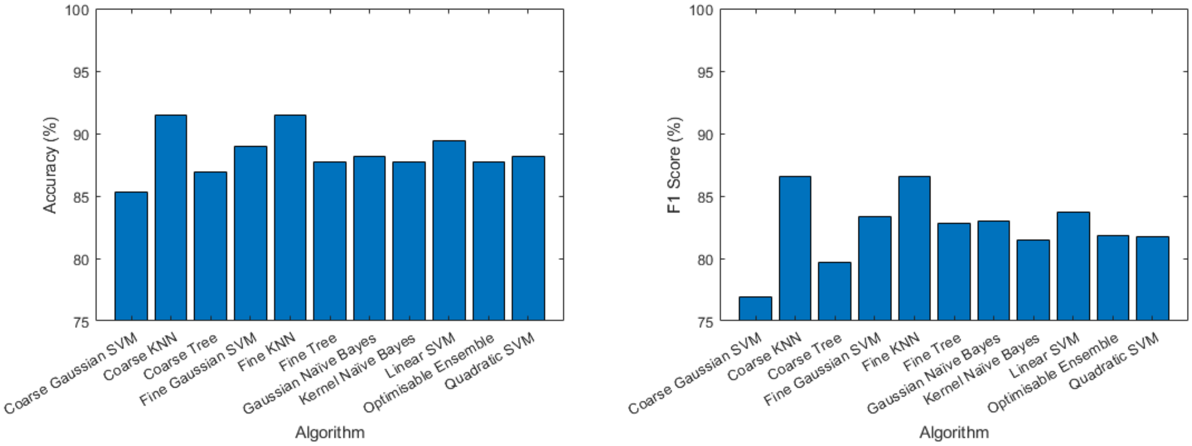
Figure 5.15: Heatmap showing the correlation between initially extracted features using Pearson's Correlation Coefficient.

5.4.2 Results and Conclusions

Figure 5.16 and Table 18 show the average 5-FCV results of the 11 classical ML algorithms used for training and testing in the Temperature classification task. Within Figure 5.16, (a) shows the average accuracy while (b) shows the average F1 Score. A majority of the models show a relatively high classification performance. Two models, however, consistently have the highest scoring accuracy and F1 Score. The Coarse and Fine KNN models show the best performance with respect to their distribution across each metric, both also having an average accuracy of 91.43% and F1 Score of 86.59%. KNN models are one of the least complex classification algorithms and are therefore very quick to train, which can be useful when further training data becomes available. Other models, including the Guassian SVM, Tree, and Naive Bayes models, also performed well but achieved neither a higher accuracy or F1 Score than either of the KNN models, as shown in Table 18. The Fine KNN model has therefore been selected as it uses fewer neighbours than the Coarse KNN algorithm to determine classification labels. This is a lower number of neighbours is less likely to overlap in KNN's mapping of features for samples with different labels, meaning that samples labelled as having low temperature are less likely to be confused with those labelled as having high temperature during this classification. This, however, can only be confirmed with further testing using new, unseen samples.

Table 18: Averaged 5-FCV results for the Finishing Mill and Coiler Temperature classification task.

Algorithm	**Accuracy (%)**	**F1 Score (%)**
Coarse Gaussian SVM	85.31	76.88
Coarse KNN	91.43	86.59
Coarse Tree	86.94	79.72
Fine Gaussian SVM	88.98	83.32
Fine KNN	91.43	86.59
Fine Tree	87.76	82.77
Gaussian Naïve Bayes	88.16	83.03
Kernel Naïve Bayes	87.76	81.47
Linear SVM	89.39	83.71
Opt. Ensemble	87.76	81.85
Quadratic SVM	88.16	81.72



(a) Accuracy.

(b) F1 Score.

Figure 5.16: Average 5-FCV performance of each classification algorithm in the Temperature classification task.

While the model performed well, misclassifications and their circumstances should still be considered. Figure 5.17 shows the processed temperature signal for one such example. Originally, this samples was labelled by analysts as having a low Finishing Mill temperature. Upon visual inspection of the signal, it can be seen that this is the true label, especially since, in this particular case, the temperature drops below the lower tolerance. The model, however, classified this sample as having high temperature, likely due to consistently high values in the later half of the signal. It is possible that this has been caused by both the limited amount of data available for training and testing the model and due to the imbalance between the classes in this particular classification task. While it is best to use a balanced dataset when creating ML models, the maximum number of samples for each defective behaviour was used in this case as samples were already limited. Future work would aim to improve both the performance and reliability of this model by balancing classes once a sufficient amount of new data becomes available.

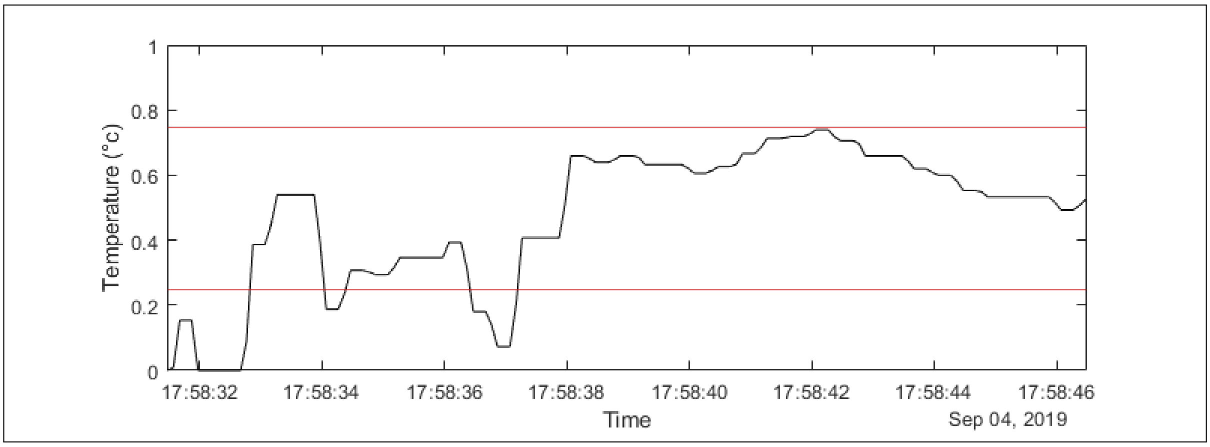


Figure 5.17: Example of a signal labelled by analysts as having low temperature, misclassified by the model as having high temperature.

5.5 Summary and Conclusions

In this chapter, the possible root causes of those width-related defects described in Chapter 4 were discussed in detail, including those that can be found during Scheduling and Furnace reheating prior to the Roughing Mill subprocess. Specifically, the nature and origin of these root causes and how they are identified by operators and analysts in the HSM. From this, further rules are derived for inclusion in the decision-making process described in Chapter 6. DL and multiple ML approaches have been used for those root causes which require the analysis of ambiguous time series data to which simple conditional rules could not be applied. Specifically, a pre-trained CNN architecture and Transfer Learning were used to create a DL image classification model for the purpose of identifying early ANC stroke timing in the Roughing Mill subprocess in Section 5.3. A traditional Fine KNN ML model was also created for the purpose of distinguishing between characteristics of high and low temperatures in the Finsihing Mill and Coiler subprocesses in Section 5.4. These models are to be used in conjunction with rules derived from the expert knowledge gained in Section 5.1 such that both aspects of the final decision-making process contribute to a well-informed decision in the tool created in Chapter 6 [1]. Each model was trained and tested using a combination of 5-FCV and GSO using a range of hyperparameter values. Each model showed optimal performance and will be combined with rules derived from expert knowledge, to create the data-driven web tool proposed in Chapter 6 which will ultimately have the capability of detecting width-related defects, described in Chapter 4, and their root causes.

The experiments carried out in this chapter has resulted in two ML models which perform sufficiently for the intended application, as shown in Sections 5.3.2 and 5.4.2. The experience gained from the experiments carried out in Chapter 4 also helped towards achieving this. However, although the Finishing Mill and Coiler Temperature classification model performed well, it was difficult to address this challenge using the same approach taken in the image classification tasks. This is because Transfer Learning for image classification models aids in the generalisation of features found using edge detection. To use Transfer Learning or a pre-training methodology for numeric features, the nature of the data originally used for training must be similar to that of the intended application. However, since the application is both domain and process-specific, there is no publicly available model trained on the relevant data and features to be able to apply this methodology to the Temperature classification task. For this reason, it is still recommended to continue collecting data and retraining this model in an effort to improve its performance.

Further discussions with analysts and, thus, further gathering of expert knowledge has also been shown to be invaluable in constructing both the models in this chapter and the decision-making process in Chapter 6. Specifically, it has helped to identify and analyse instances of strips with abnormal behaviour which either does not fall into the scope of the classification models produced, thus needing further conditional checks, or causes the models to misclassify test samples. One example of this is shown in Section 5.3.1, specifically Figure 5.10. Upon investigation, a bar was found to have fundamental width issues, making the timing of the ANC stroke irrelevant and ineffective. After further discussion with an analyst, it was determined that this bar should not be used as a training sample in the ANC stroke timing classification task, regardless of its true timing, so that it does not negatively impact the network's perception of what a true early or well-timed sample is. This case further highlights the need to discuss and include expert knowledge necessary for deducing such unique or overlooked cases, and future work would aim to include checks for this particular underlying cause and also to factor in the feedback of such an analysis into other models so that more informative decisions can be made even for ambiguous cases with the aim of understanding the causes and impacts of these behevaiours as they occur.

Another example showcasing the need to include expert knowledge in the decision-making process is the misclassification shown in Section 5.4.2. In this example, a sample originally labelled by an analyst as having low Finishing Mill temperature was wrongly classified by the model as having high temperature. Upon further manual inspection with an analyst, it was found that, although the temperature signal of this sample clearly exhibited behaviours of low and deviating temperatures, a large portion of the signal remained at a relatively high, although not abnormal, temperature. It was determined that these high values were likely the cause of the misclassification, and that limited data availability and unbalanced classes were likely a significant contributing factor to this, despite the models sufficient performance. While the model generally performed well, it is clear that the inclusion of expert knowledge in addition to ML is vital to creating a successful data-driven decision-making process such that the process is not completely reliant on one technology or the other but rather that certain analyses are conducted on data where appropriate to construct an informed decision.

The models created in this chapter showcase the potential of ML in steel-making applications [22, 23] and aim to encourage other researchers to expand the scope of the currently limited range of ML applications in this industry, particularly within the HSM process. As discussed, the findings of this chapter also showcase the importance of including expert knowledge when analysing defects and their root causes. This is further demonstrated in Chapter 6 by combining the created ML models created in Chapters 4 and 5 and by integrating the expert knowledge gained over the course of these chapters into a single tool with a process-wide data-driven decision-making process [1].

Future work would aim to further the results of this chapter by balancing the classes used in the Temperature classification task, and by collecting a larger number of samples for training and testing, once further data because available. ML and DL models are typically trained on hundreds of thousands of samples, but this is subject to the amount of data available which can be particularly limited in industry. Future work would aim both to explore other methods of processing, modelling, and analysing time series data of a random nature efficiently [161, 166], and to include further rules and models for the detection unforeseen cases, which may also be ambiguous, such as instances of dirty sensors causing mistiming of the ANC stroke as shown in Section 5.3.1.

In the following chapter, the architecture and development of the proposed web tool, which integrates the rules and models derived in this and the previous chapter, is described in detail. Its functionality is also showcased and the models are tested again within this context using newly extracted data.

6 Combining and Integrating Machine Learning and Expert Knowledge into a Data-Driven Decision-Making Process

The purpose of this chapter is to combine and utilise the ML models created and the expert knowledge gained from Chapters 4 and 5. This is done by creating a local web page environment which enables the loading and visualisation of data such that it simulates data loading sequentially in HSM subprocesses, and the execution of a decision-making process derived from the previously mentioned ML models and expert knowledge and the visualisation of the results it produces. In Section 6.1, the Data Collection and Management methods used throughout the thesis and the current procedures used in the Port Talbot HSM, including access permissions and limitations, are reviewed. In Section 6.2, the design of the proposed web tool is described in great detail. This includes how the local web environment and a new test dataset are created, how data is loaded into the web tool from this dataset, and the decision-making process constructed from the models created and expert knowledge gathered in previous chapters. In Section 6.3, the web tool is demonstrated by visualising the results of the decision-making process on the new test data. Finally, in Section 6.4, the web tool and its success with regards to the aims of the thesis and its performance when processing the new test data. The findings and results of this chapter form part of the published journal article titled ‘A Tool to Combine Expert Knowledge and Machine Learning for Defect Detection and Root Cause Analysis in a Hot Strip Mill’ [1], mentioned in Section 1.4.1.

The output of this is a web tool that is designed to enable operators to track width-related defects as they occur during the HSM process, providing a unified view of the process. The tool uses the same data pre-processing steps and trained models described in the previous chapters. While a decision has not yet been made by Port Talbot Steelworks on whether to deploy the web tool, this section provides a demo and test examples through the tool’s graphical user interface.

6.1 Data Infrastructure and Review of Current Procedures in the Port Talbot Hot Strip Mill

6.1.1 Review of Data Collection

The data used to produce the images and features used to train the ML models created in Chapters 4 and 5 is collected from the existing Port Talbot HSM data infrastructure using the process described in Section 4.2.1, and the data sources discussed in Section 3.1. Time series data produced during the PT HSM process is stored in an independent server which can be accessed using IBA Analyzer, a data analysis tool for time series data. The time series themselves are stored in application-encoded .iba files, but can be extracted into a .csv format by an analyst with the correct access permissions. Access permissions are discussed in further detail in Section 6.1.3. The time series data can be accessed is typically accessed on a coil-by-coil basis, meaning that the data relating to each coil can be viewed or extracted individually.

Process data consisting of simple data types is stored onsite in a database run on a local server. There are many tables contained within this database and table names are based on the origin of its data. For example, scheduling data is stored in a table named ‘scheduling_pdi’, while some Roughing Mill process data is stored in a table named ‘hfr-madlog’. While other databases are run on the same server, the required database, named ‘EngServer2’, specifically contains HSM process data and thus the table data necessary for the experiments in this thesis.

6.1.2 Current Procedures

While a number of sensors, tolerances, and conditional checks are used to detect faults or abnormal behaviour in the Port Talbot HSM, the use of ML for defect detection and RCA altogether in this process is limited. On top of this, it is also rare for AI or ML to be used in the RCA and review processes of these defects, and these are therefore largely conducted manually. Currently, within the Port Talbot HSM, live strip data is monitored by operators in engineering logs and on simple dashboard interfaces which includes graphs with tolerances.

If abnormal behaviour is detected by an operator, it is recorded in a table titled ‘QC_NRFT’ which is contained within the ‘EngServer2’ database. ‘QC_NRFT’ is an acronym for ‘Quality Control Not Right First Time’. Such instances are queued for review by analysts. Analysts, however, also undertake responsibilities as research engineers and are therefore not always readily available. This is usually due to preoccupation with other workloads and responsibilities, and the time consumed by reviewing other defects in a backlog of queued instances. Unless the defect is severe, there is no queue priority in this backlog. This can result in the analysis of defects and root causes being delayed for varying amounts of time, from hours to days. During review, the analyst determines whether the defect was assessed correctly by the operator or traffic light system as well as its origins and root causes. This is carried out using a combination of the signal data viewed in ibaAnalyzer and query data from the database tables, as well as the analyst's own expert knowledge. The updated information is then recorded in another table in the EngServer2 database titled 'TBL_QC_NRFT' and is used to determine corrective actions for both successive strips of the same or similar profile, and follow up actions for post-processing or sale of the affected strip. It may also be revisited by analysts for use in long term projects when reviewing statistical data related to defects and root causes. This process is illustrated in Figure 6.1.

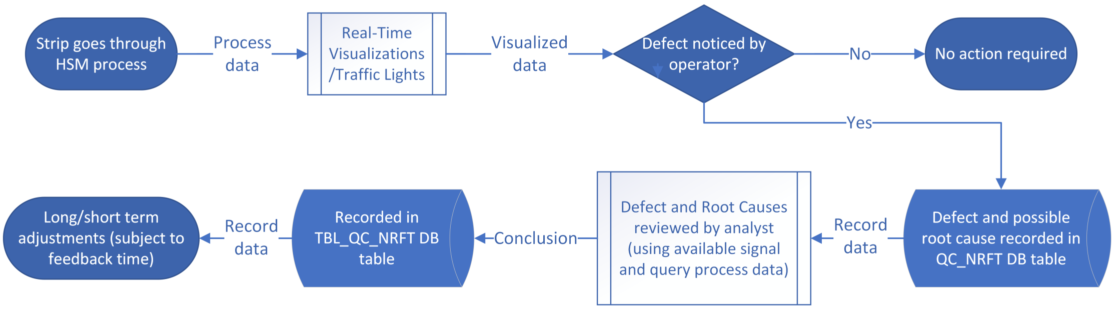

Strip goes through HSM process
Process data
Real-Time Visualizations /Traffic Lights
Visualized data
Defect noticed by operator?
No
No action required
Yes
Defect and possible root cause recorded in QC_NRFT DB table
Record data
Defect and Root Causes reviewed by analyst (using available signal and query process data)
Conclusion
Recorded in TBL_QC_NRFT DB table
Record data
Long/short term adjustments (subject to feedback time)

Figure 6.1: Current process for collecting and analysis data for defective strips.

Manual interaction from analysts using the current procedure can yield some benefits such as thoroughly reviewing abnormal behaviour, ruling out ambiguous cases, and contradicting operators when an incorrect observation has been made or insufficient details of defects and root causes have been provided. However, the current procedure still largely relies on interaction from operators which can lead to a number of issues when detecting and recording abnormal behaviour in the HSM process, particularly human error which can result in false positives or missed instances [167]. Longer review and feedback times can also have an impact on both the current strip and successive strips if mitigation actions are not performed quickly enough in response to abnormal behaviour. These longer review times can naturally result in an increased workload for analysts, diverting their attention from other projects as well as delaying the review of other strips with detected abnormal behaviour even further [167]. This can even have knock-on effects on post-processing and sales if final decisions are not made quickly enough.

In this chapter, an automated web tool is created with the aim of improving upon the current, largely manual procedure for detecting and analysing defects and their root causes in the Port Talbot HSM process. This tool focuses on width-related defects and incorporates the use of AI and ML by integrating the models created in experiments in Chapters 4 and 5 and combining them with conditional rules derived from the expert knowledge gained in these chapters. This combination enables the tool to feedback to users whether a defect has occurred and, if so, its potential root cause and origin. Through this, the tool provides a foundation upon which further AI and ML tools can be integrated for use with operators such that defects can be caught, reviewed, and fed back as soon as possible, leading to quicker decision-making with regards to both corrective and follow up actions compared to the current processes outlined in Figures 3.2 and 6.1. The tool is run in a local web environment and loads data dynamically for each strip such that feedback is provided after they have completed each subprocess.

6.1.3 Access, Permissions, and Limitations

As discussed in Section 6.1.1, ibaAnalyzer is used by analysts to access and review HSM time series data. Normal .iba files, which are the of the file type associated with the ibaAnalyzer software, can be downloaded at anytime. However, as dicussed in Section 3.1.1, only a limited number of employees with the correct access permissions can extract time series into readable file formats, such as .csv, using a script.

Any operator or analyst in the HSM is able to access .iba files for general viewing in ibaAnalyzer. However, higher permissions are only given to a select few analysts as increased live access rates would slow down access speeds. Any operator or analyst in the HSM can also view data stored in database tables using SQL software and the correct permissions. From this, data can be queried and extracted into the required format.

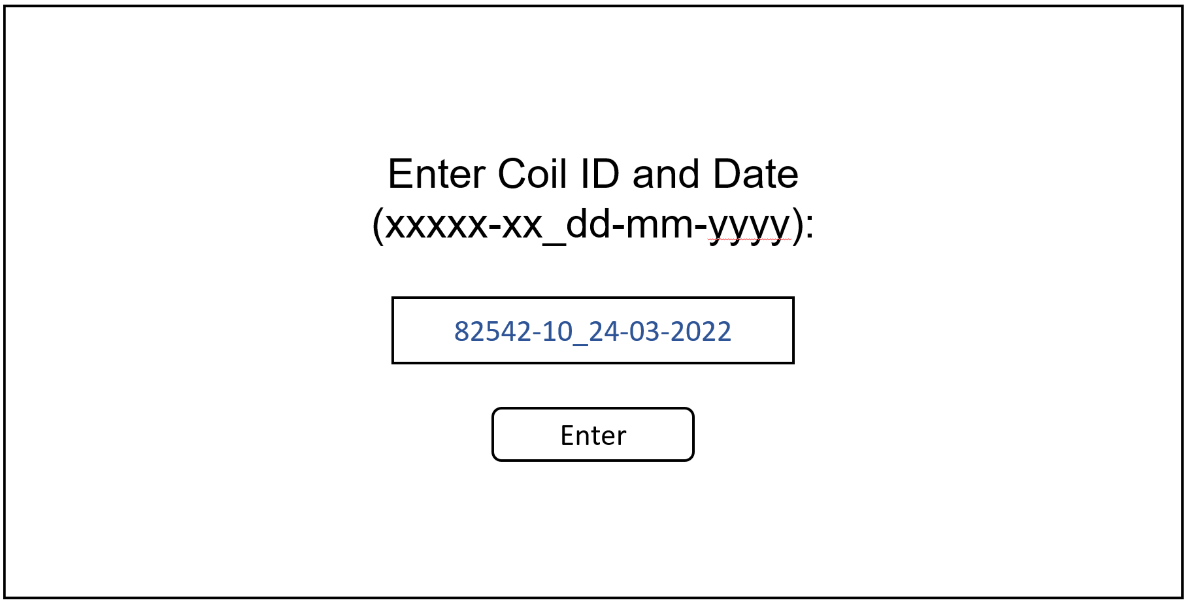
Only a select few analysts can access IBA data using a connection to the IBA server itself. Too many connections would slow down access speeds and interfere with the performance of live interfaces accessing data via these connections. Within this project, limitations on access permissions has resulted in approaching an analyst with these permissions and extracting the data needed by proxy to create the ML models in Chapters 4 and 5, and the new test data in this chapter.

6.2 Proposed Decision-Making Process and Web Tool Combining Machine Learning and Expert Knowledge

6.2.1 Creating a Local Web Environment

The web tool has been created using a combination of PHP, HTML, and JavaScript, the same web infrastructure used in the PT HSM. The local environment in which the tool is run is created using XAMPP, a tool which enables the user to host web servers locally and to run web pages without uploading any files or data to an external server. As the models created in Chapters 4 and 5 were created using MATLAB, the MATLAB compiler application is used to create executable programs which can be run programmatically by the web tool when the decision-making process, described in Section 6.2.4, must be performed.

The design of the web tool's user interface is made both progammatically and visually similar to existing tools used to monitor data in the PT HSM. This will make it more familiar and thus simpler to use for both operators and analysts, and, if deployed, will make it simpler to integrate into the PT HSM's web service. The tool features a search page from which users are able to enter a coil ID and date, and an interface to visualise the data for the corresponding coil, initial designs for which are illustrated in Figures 6.2 and 6.3, respectively. The tools functionality, including loading, processing, and visualising data, is described in detail in the following sections.



Enter Coil ID and Date
(xxxxx-xx_dd-mm-yyyy):

82542-10_24-03-2022

Figure 6.2: Design and layout of the search page to be included in the proposed web tool.

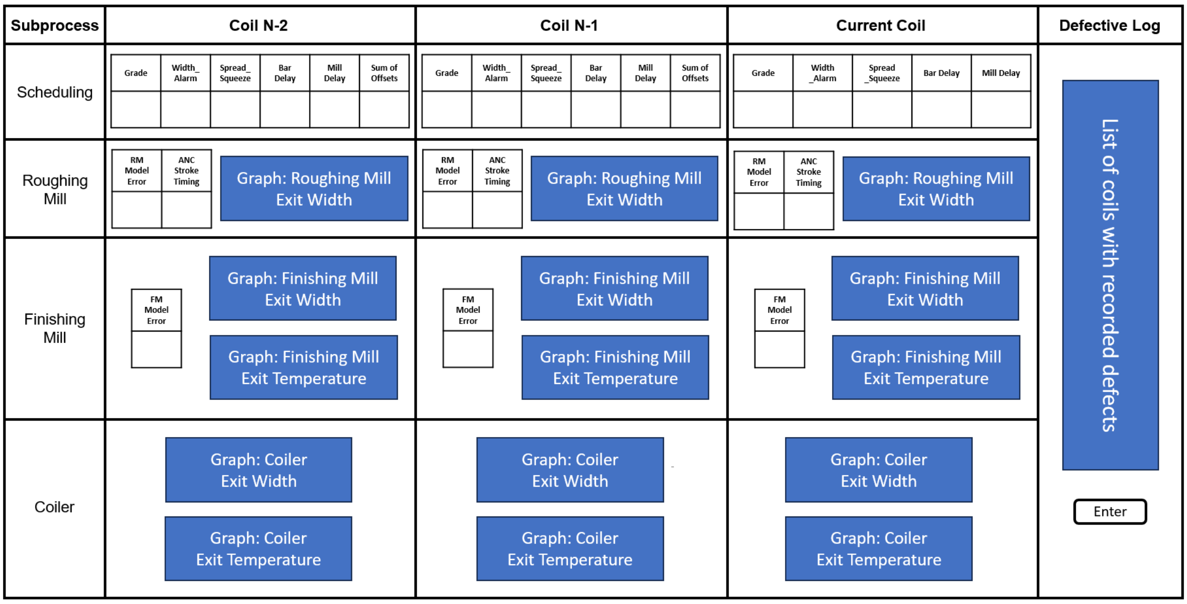


Figure 6.3: Design and layout of the main page to be included in the proposed web tool.

6.2.2 Data Review and Further Data Collection

To resolve the issues related to access speeds and permissions, and to be able to test in a local environment before deployment can be considered, the same Data Collection process used in Section 4.2.1 is used to create another local dataset by combining time series data from the IBA server and simple data types from the 'EngServer2' database. This process is also reviewed in the illustration shown in 6.4. This dataset is loaded and processed in MATLAB so that it can be utilised by executable, or '.exe', programs created using the MATLAB Compiler application. These '.exe' programs are discussed in further detail in Section 6.2.3.

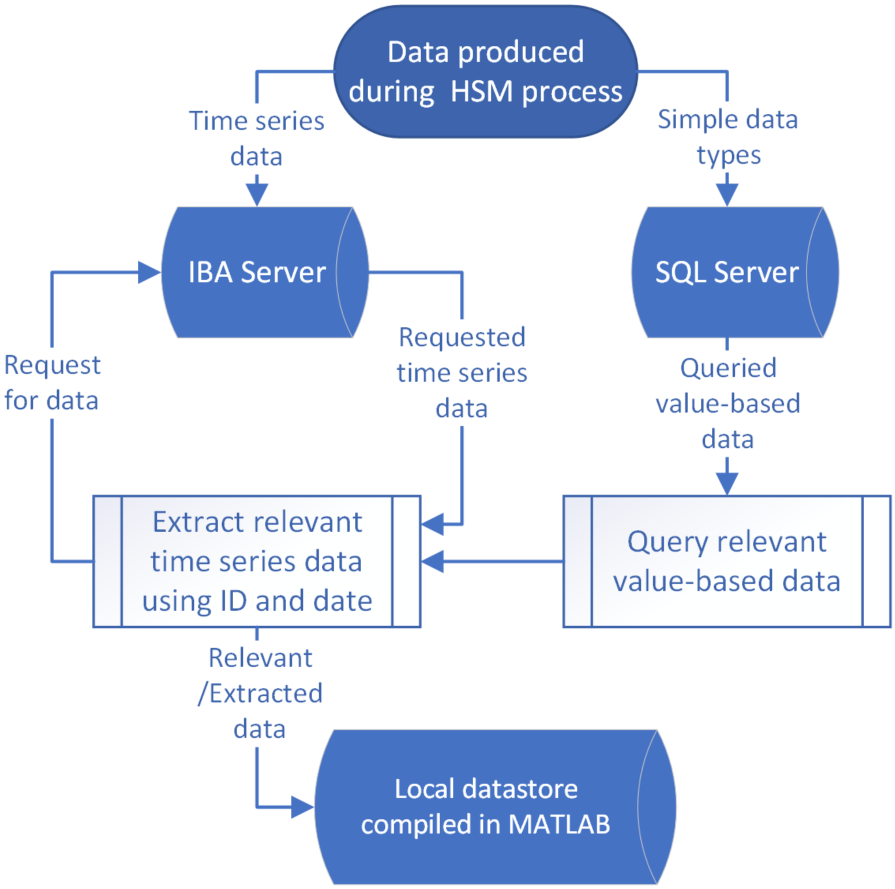

Data produced during HSM process
Time series data
Simple data types
IBA Server
SQL Server
Request for data
Requested time series data
Queried value-based data
Extract relevant time series data using ID and date
Query relevant value-based data
Relevant /Extracted data
Local datastore compiled in MATLAB

Figure 6.4: Flowchart showing the flow of data through the HSM data infrastructure.

Unfortunately, the amount of newly available samples labelled in the TBL_QC_NRFT table as having the defective behaviours discussed in this thesis were very limited in the time between the models in Chapters 4 and 5 being created and the web tool being tested. It was decided, however, that these samples would still be collected and tested within the web tool along with a number of samples which were not labelled as having any abnormal behaviours to confirm that false positives would not be produced by the decision-making process described in Section 6.2.4. The new testing data is described in Table 19. As discussed in Section 4.2.1, root causes can sometimes go unrecorded or unidentified by operators and analysts, which is the case for some of these samples.

6.2.3 Loading Data

As shown in Section 6.2.1, the web tool provides the user with a search page in which they are able to enter the ID and date of a specific coil. After entering this information, the user is

Table 19: Number of new samples collected for testing the web tool.

Defect	**No. of Samples**
Necking	5
Flare	3
Width Pull	5
Coiler Snatch	2
Okay	50

presented with the tool's main interface. Data for the searched coil is loaded in an allocated column and, after completing each subprocess, a data-driven decision-making process is used to determine whether width-related defects are present, as well as their potential root causes. Once all subprocesses have been completed, the data for this coil moves into the next column and the process repeats for data for the following coil. If the user enters no information, the search will result in loading data for the most recent coil. This process is illustrated in Figure 6.5 but is described in thorough detail in the remained of this section, and the data-driven decision-making process is also described in detail in the following section, Section 6.2.4.

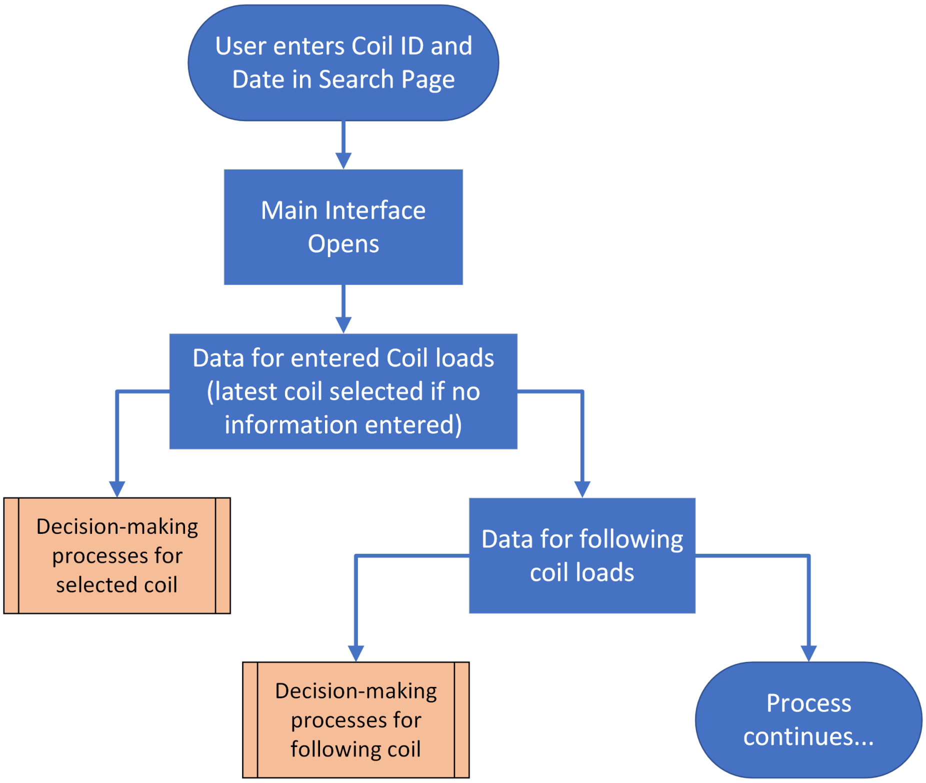

User enters Coil ID and Date in Search Page
Main Interface Opens
Data for entered Coil loads (latest coil selected if no information entered)
Decision-making processes for selected coil
Data for following coil loads
Decision-making processes for following coil
Process continues...

Figure 6.5: High-level overview of the web tool's functionality.

6.2.3.1 Scheduling and Furnace Data

To use the web tool, a search page is first used to prompt the user to input a coil ID and date from which the main web page is loaded. Each subprocess for each coil is displayed as an individual frame and time series data for the Roughing Mill, Finishing Mill, and Coiler subprocesses is loaded dynamically. When the main web page is loaded, the 'Load_PDI_Data.exe' program is run, passing Scheduling and Furnace data from the local MATLAB data structure into the web tool environment. The data passed into the web tool represents both the bar of which the coil ID was selected. The data passed in this operation is then checked for abnormal behaviour using conditional statements in the web tool's PHP code, and displayed in a table in the Scheduling and Furnaces section in the 'Coil N' column, following the decision-making process described in Section 6.2.4.1.

6.2.3.2 Roughing Mill Data

Once operations for Scheduling and Furnace data have been completed, width data for the Roughing Mill subprocess is passed into the web tool environment by running 'Load_RM_Data.exe'. The received data is loaded dynamically into a graph in the Roughing Mill subprocess section in the 'Coil N' column. Once data has finished loading into the graph, 'RM_Classify.exe' is run, carrying out the decision-making process described in Section 6.2.4.2. This process carries out conditional checks and, if necessary, image classification to determine the presence and potential causes of defects in the Roughing Mill subprocess, including ANC stroke timing and Roughing Mill model error. Feedback from the process is passed back from 'RM_Classify.exe' to the web environment and is displayed in the Roughing Mill section of the 'Coil N' column. Data is highlighted red or green, and is accompanied by a label, based on the feedback received from 'RM_Classify.exe'.

6.2.3.3 Finishing Mill Data

Once operations for Roughing Mill data have been completed, width and temperature data for the Finishing Mill subprocess is passed into the web tool environment by running 'Load_FM_Data.exe'. The received data is loaded dynamically into a graph in the Finishing Mill subprocess section in the 'Coil N' column. Once data has finished loading into the graph, 'FM_Classify.exe' is run, carrying out the decision-making process described in Section 6.2.4.3. This process carries out conditional checks and, if necessary, classifications to determine the presence and potential causes of defects in the Finishing Mill subprocess, including high or low temperatures and Finishing Mill model error. Feedback from the process is passed back from 'FM_Classify.exe' to the web environment and is displayed in the Finishing Mill section of the 'Coil N' column. Data is highlighted red or green, and is accompanied by a label, based on the feedback received from 'FM_Classify.exe'.

6.2.3.4 Coiler Data

Once operations for Finishing Mill data have been completed, width and temperature data for the Coiler subprocess is passed into the web tool environment by running 'Load_DC_Data.exe'. The received data is loaded dynamically into a graph in the Coiler subprocess section in the 'Coil N' column. Once data has finished loading into the graph, 'DC_Classify.exe' is run, carrying out the decision-making process described in Section 6.2.4.4. This process carries out conditional checks and, if necessary, classifications to determine the presence and potential causes of defects in the Coiler subprocess, including high or low temperatures. Feedback from the process is passed back from 'DC_Classify.exe' to the web environment and is displayed in the Coiler section of the 'Coil N' column. Data is highlighted red or green, and is accompanied by a label, based on the feedback received from 'DC_Classify.exe'.

Once the Coiler data for the current coil has finished loading, its data is transferred to the 'Coil N-1' column and the Roughing Mill data for the subsequent strip will begin loading in the 'Coil N' column. The sum of offsets made to the originally selected coil will also be displayed in the 'Coil N-1' column along with feedback produced by conditional checks made within the web tool. The entire process is illustrated in Figure 6.6.

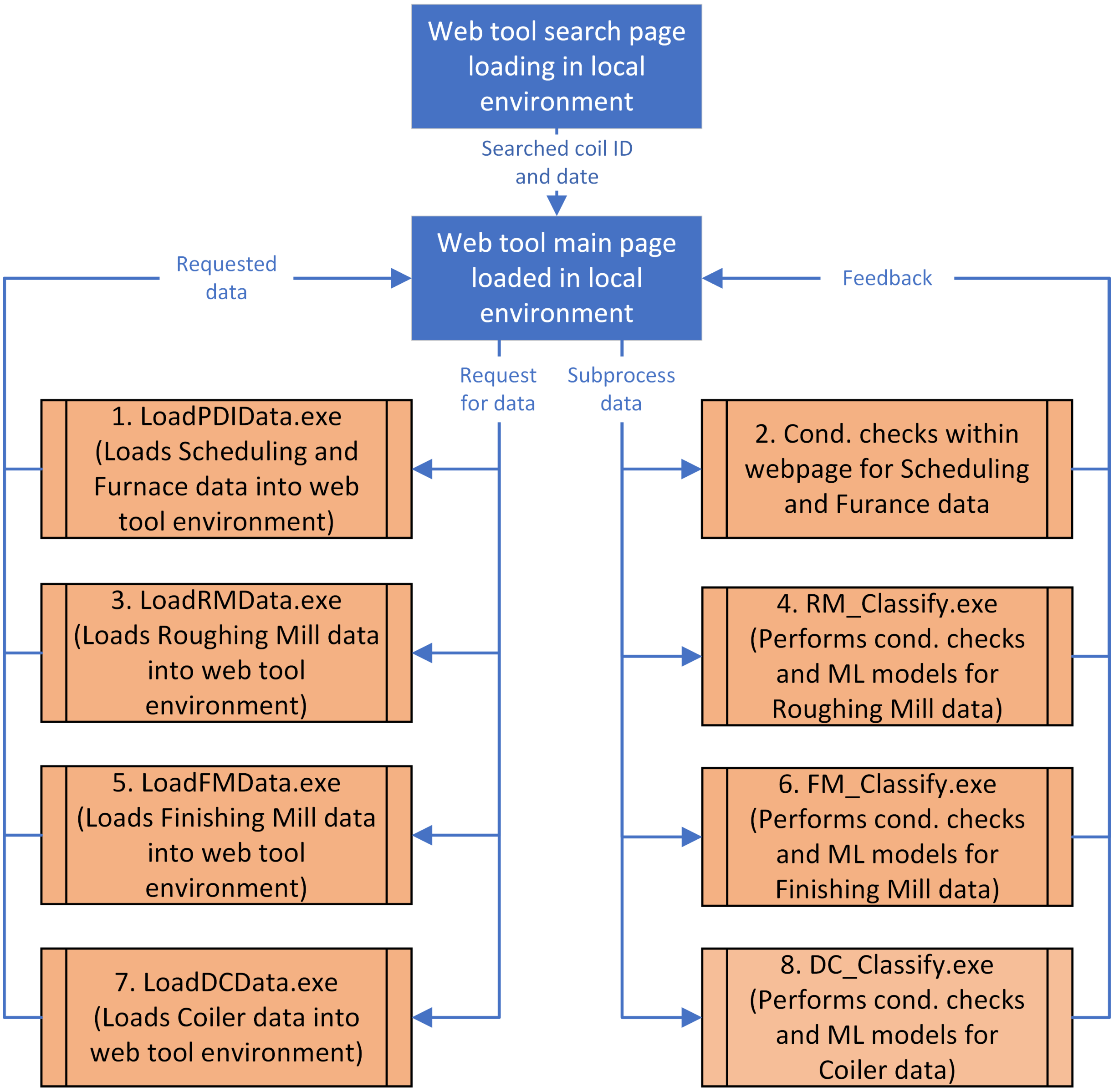

Web tool search page loading in local environment
Searched coil ID and date
Web tool main page loaded in local environment
Requested data
Feedback
Request for data
Subprocess data
1. LoadPDIData.exe (Loads Scheduling and Furnace data into web tool environment)
2. Cond. checks within webpage for Scheduling and Furance data
3. LoadRMData.exe (Loads Roughing Mill data into web tool environment)
4. RM_Classify.exe (Performs cond. checks and ML models for Roughing Mill data)
5. LoadFMData.exe (Loads Finishing Mill data into web tool environment)
6. FM_Classify.exe (Performs cond. checks and ML models for Finishing Mill data)
7. LoadDCData.exe (Loads Coiler data into web tool environment)
8. DC_Classify.exe (Performs cond. checks and ML models for Coiler data)

Figure 6.6: Overview of the web tool's functionality.

6.2.4 Decision-Making and Classification

6.2.4.1 Scheduling and Furnaces

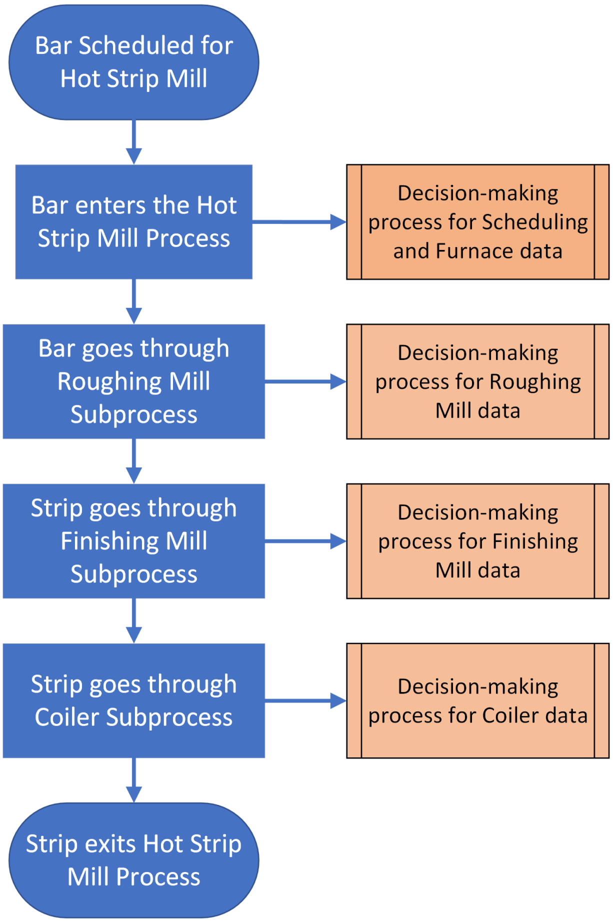

Bar Scheduled for Hot Strip Mill
Bar enters the Hot Strip Mill Process
Decision-making process for Scheduling and Furnace data
Bar goes through Roughing Mill Subprocess
Decision-making process for Roughing Mill data
Strip goes through Finishing Mill Subprocess
Decision-making process for Finishing Mill data
Strip goes through Coiler Subprocess
Decision-making process for Coiler data
Strip exits Hot Strip Mill Process

Figure 6.7: Overview of the web tool's decision-making process.

The decision-making process of the proposed web tool, outlined in Figure 6.7, emulates the logic used by operators to identify width-related defects and their root causes. Simple conditional checks, such as those for defects originating from Scheduling or the Furnaces are carried out within the PHP code used to construct the page. These include checking whether the current strip is of one of a known number of problematic grades, whether the width alarm for Spread/Squeeze rules has been triggered, whether the HSM is currently using single furnace operation, and whether delay has occurred in the furnaces. If true, these are flagged within the web tool by highlighting their table values in red and can be indicators or root causes of potential width-related defects later in the HSM process. This part of the decision-making process is illustrated in Figure 6.8.

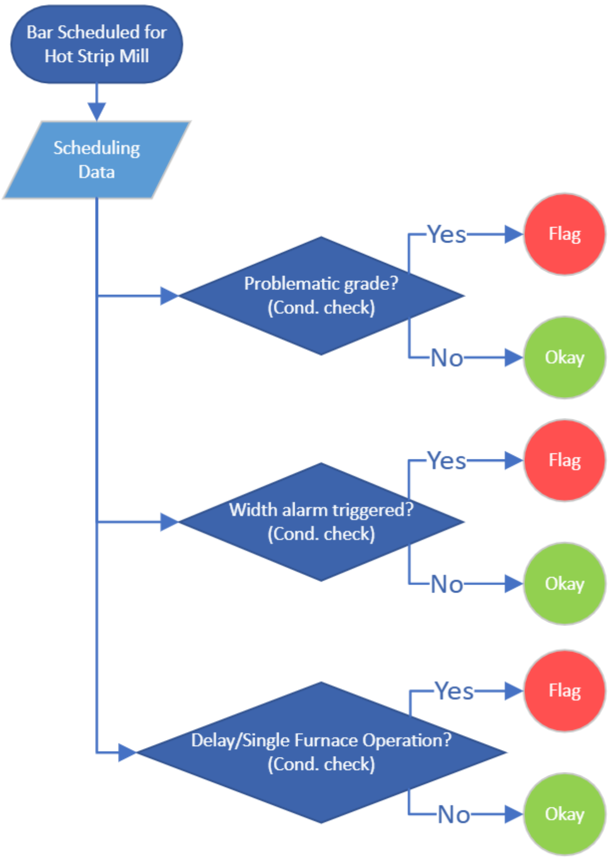

Bar Scheduled for Hot Strip Mill
Scheduling Data
Problematic grade? (Cond. check)
Width alarm triggered? (Cond. check)
Delay/Single Furnace Operation? (Cond. check)
Yes
No
Flag
Okay

Figure 6.8: Decision-making process for Scheduling and Furnace data.

6.2.4.2 Roughing Mill Subprocess

As described in previous subsections, decision-making and classification for HSM subprocesses occur after data has been loaded dynamically into the web page. Once data for the Roughing Mill subprocess has finished loading, data for the Finishing Mill subprocess will begin to load and the 'RM_Classifiy.exe' program is run. All decisions for Roughing Mill data, including the classification of Necking, Flare, and ANC stroke timing classifications, are made within this program following the decision-making process shown in Figure 6.9.

The 'RM_Classifiy.exe' program first calculates whether the width deviation of the bar is -50mm for its entire length to determine whether there is a faulty sensor is present. In the HSM, this is the cut off point for low values inputted by this sensor and is the fault value if the sensor is on but not receiving any data. The program then checks if valid width data is being read by calculating whether over 10% of the bar's width deviation is less than -50mm, or if over 50% is greater than 50mm, which could indicate fundamental width issues. The Roughing Mill Necking and Flare image classification model created in Chapter 4 is then used to determine whether the bar contains Necking or Flare at its head and tails ends, or whether it shows no abnormal behaviour. To do this, images are created using the width deviation signal data before loading the previously created classification model and using it to classify the new image.

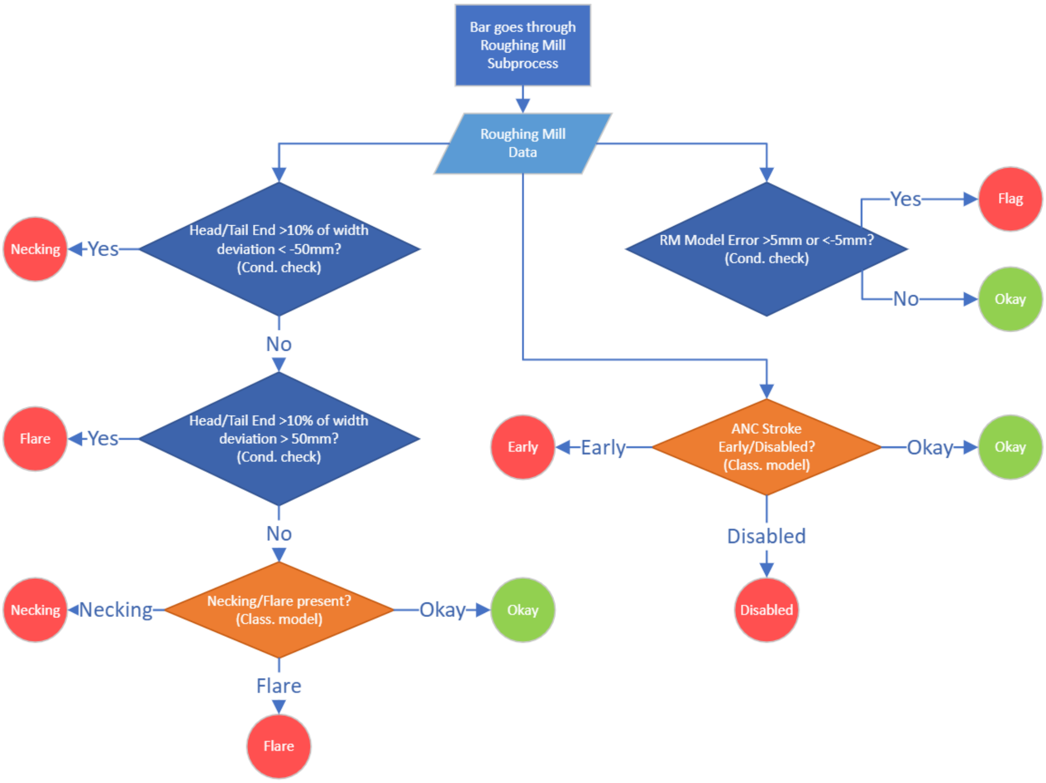

Bar goes through Roughing Mill Subprocess
Roughing Mill Data
Head/Tail End >10% of width deviation < -50mm? (Cond. check)
Yes
Necking
No
Head/Tail End >10% of width deviation > 50mm? (Cond. check)
Yes
Flare
No
Necking/Flare present? (Class. model)
Necking
Necking
Okay
Okay
Flare
Flare
RM Model Error >5mm or <-5mm? (Cond. check)
Yes
Flag
No
Okay
ANC Stroke Early/Disabled? (Class. model)
Early
Early
Okay
Okay
Disabled
Disabled

Figure 6.9: Decision-making process for Roughing Mill data.

Following this, a conditional check is made to determine whether the Roughing Mil model error is less than -5mm or greater than 5mm, which would indicate contribution to potential width-related defects originating in or after the Roughing Mill subprocess. The program then determines if ANC stroke timing was disabled by checking whether the the ANC activation signal shows any true values. If ANC was not disabled, classification of ANC stroke timing is then carried out by loading the image classification model created in Chapter 5. Images for each Roughing Mill pass are created using the edger force, capsule position, and ANC activation signals, and are classified using the corresponding model. This will determine whether ANC stroke timing is Early or Okay. Before completing execution, 'RM_Classifiy.exe' feeds the results of these decisions to the web tool environment and any width-related defects and potential root causes that are present are flagged accordingly with a corresponding label and highlighted red in the Roughing Mill data row. This part of the decision-making process is illustrated in Figure 6.9.

6.2.4.3 Finishing Mill Subprocess

Once data for the Finishing Mill subprocess has finished loading, data for the Coiler subprocess will begin to load and the 'FM_Classify.exe' program is run. All decisions for Finishing Mill data, including the presence of Width Pull and Temperature-Related root causes, are made within this program following the decision-making process shown in Figure 6.10.

The program first checks for fundamental outlier issues as indicators of Full Coil Under Width and Over Width by calculating whether more than 10% of the total width deviation is less than the lower width tolerance, or greater than 50% of the total width deviation is greater than the upper width tolerance. If no fundamental width issue is found here, the head end of the strip is then checked for Width Pull using the image classification model created in Chapter 4. An image is created using the width deviation signal and the upper and lower tolerances before loading the previously created classification model and using it to classify the new image. This will finally determine whether Width Pull is present in the head or tails ends of the strip, or if there is no abnormal behaviour present.

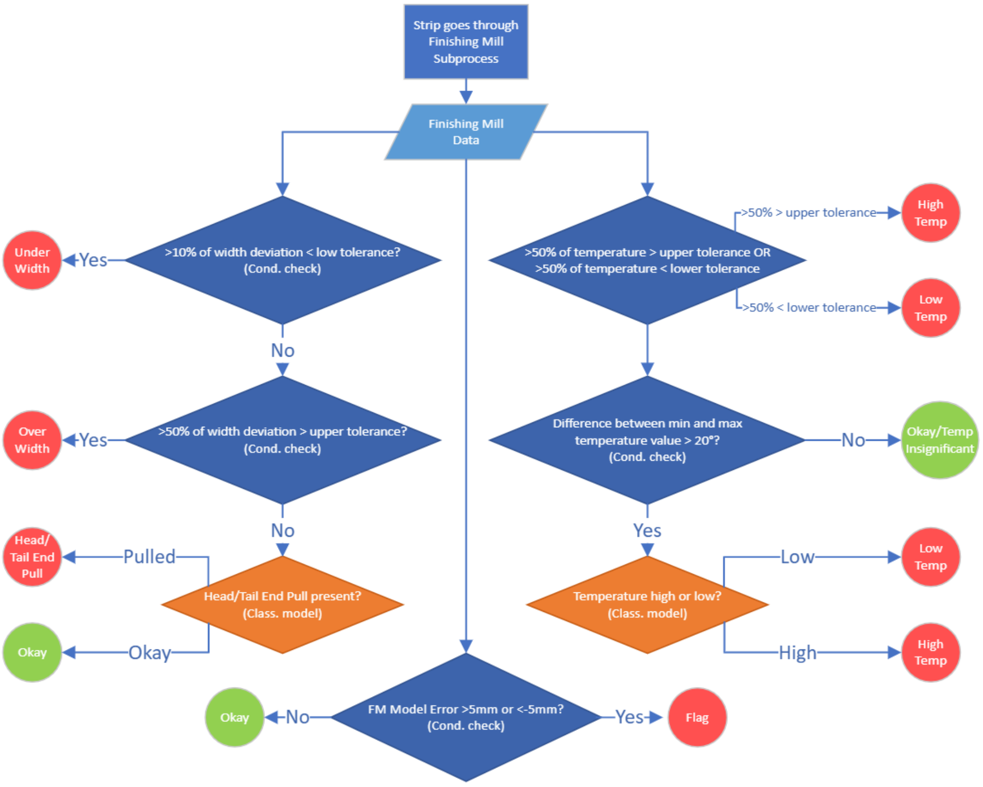

Strip goes through Finishing Mill Subprocess
Finishing Mill Data
>10% of width deviation < low tolerance? (Cond. check)
Yes
Under Width
No
>50% of width deviation > upper tolerance? (Cond. check)
Yes
Over Width
No
Head/Tail End Pull present? (Class. model)
Pulled
Head/ Tail End Pull
Okay
Okay
>50% of temperature > upper tolerance OR >50% of temperature < lower tolerance
>50% > upper tolerance
High Temp
>50% < lower tolerance
Low Temp
Difference between min and max temperature value > 20°? (Cond. check)
No
Okay/Temp Insignificant
Yes
Temperature high or low? (Class. model)
Low
Low Temp
High
High Temp
FM Model Error >5mm or <-5mm? (Cond. check)
No
Okay
Yes
Flag

Figure 6.10: Decision-making process for Finishing Mill data.

Following this, a conditional check is made to determine whether the Finishing Mill model error is less than -5mm or greater than 5mm, which would indicate contribution to potential width-related defects originating in or after the Finishing Mill subprocess. Temperature is then checked to determine whether it is either okay or if it bears no significant impact on width behaviour. This is carried out firstly by checking whether a majority of over 50% of the strip is above or below the upper and lower tolerances, respectively, to determine whether there are fundamental temperature issues. If not, a check is performed to calculate whether the difference between the minimum and maximum temperature values is greater than 20°. If temperature values are found to deviate in a greater range than this, the temperature classification model created in Chapter 5 is used to determine whether temperatures deviate to a high or low state, thus being a contributing factor to any abnormal width behaviour. This may also reflect on previous indicators such as single furnace operation or delay. The correct numeric features are extracted from the Finishing Mill temperature signal before loading the model and using it to classify them. Before 'FM_Classify.exe' completes execution, the results of these decisions are fed back to the web tool environment and any width-related defects and potential root causes are flagged accordingly with a corresponding label and highlighted red in the Finishing Mill data row. This part of the decision-making process is illustrated in Figure 6.10.

6.2.4.4 Coiler Subprocess

Once data from the Coiler subprocess has finished loading, the 'DC_Classify.exe' program is run. All decisions for Coiler data, including the presence of Coiler Snatch and Temperature-Related root causes are made within this program following the decision-making process shown in Figure 6.11.

The program first checks for fundamental outlier indicators of Full Coiler Under Width and Over Width by calculating whether less than 10% of the total width deviation is less than the lower width tolerance, or greater than 50% of the total width deviation is greater than the upper width tolerance. If no fundamental width issue is found here, the entire length of the strip is checked for Coiler Snatch using the image classification model created in Chapter 4. Images are created using the width deviation signal data and the upper and lower tolerances before loading the previously created classification model and using it to classify the new images. This will finally determine whether Coiler Snatch is present in the strip, or if there is no abnormal width behaviour preset.

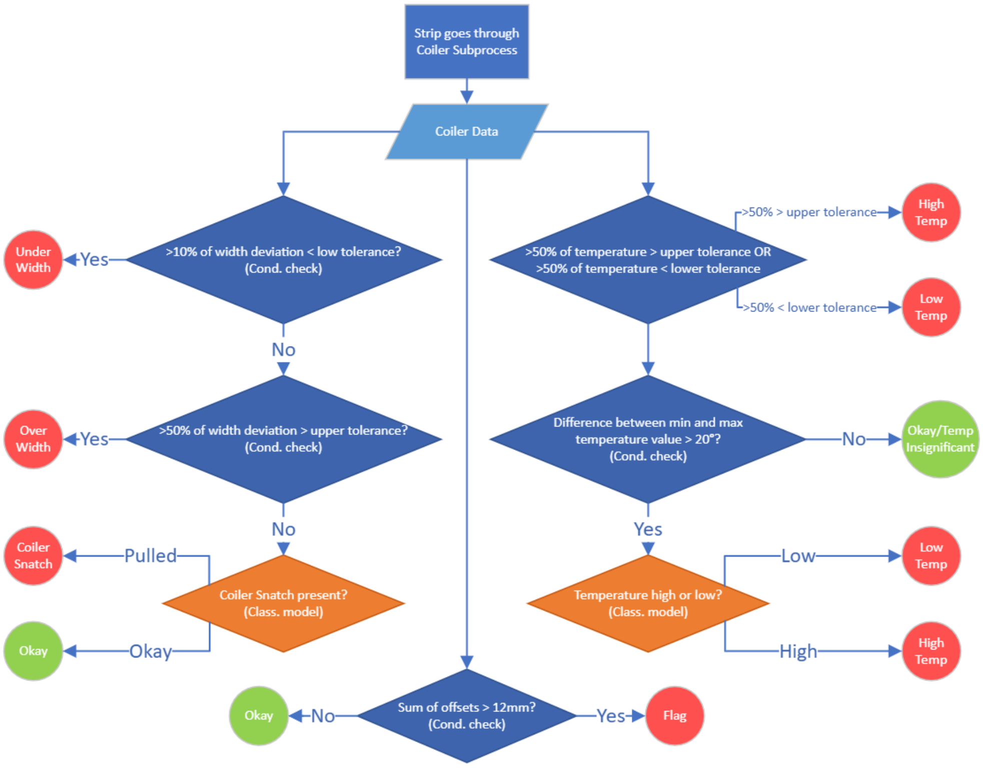

Strip goes through Coiler Subprocess
Coiler Data
>10% of width deviation < low tolerance? (Cond. check)
Yes
Under Width
No
>50% of width deviation > upper tolerance? (Cond. check)
Yes
Over Width
No
Coiler Snatch present? (Class. model)
Pulled
Coiler Snatch
Okay
Okay
>50% of temperature > upper tolerance OR >50% of temperature < lower tolerance
>50% > upper tolerance
High Temp
>50% < lower tolerance
Low Temp
Difference between min and max temperature value > 20°? (Cond. check)
No
Okay/Temp Insignificant
Yes
Temperature high or low? (Class. model)
Low
Low Temp
High
High Temp
Sum of offsets > 12mm? (Cond. check)
No
Okay
Yes
Flag

Figure 6.11: Decision-making process for Coiler data.

Similarly to the Finishing Mill subprocess, temperature is then checked to determine whether it is either okay or if it bears no significant impact on width behaviour. This is carried out by checking whether the difference between the minimum and maximum values of the strip's temperature is greater than 20°. If temperature values are found to deviate in a greater range than this, the temperature classification model created in Chapter 5 is used to determine whether temperatures deviate to a high or low state, thus being a contributing factor to any abnormal width behaviour. This may also reflect on previous indicators such as single furnace operation, delay, or high or low Finishing Mill temperatures. The correct numeric features are extracted from the Finishing Mill temperature signal before loading the model and using it to classify them. Before 'DC_Classify.exe' completes execution, the results of these decisions are fed back to the web tool environment and any width-related defects and potential root causes are flagged accordingly with a corresponding label and highlighted red in the Coiler data row. This part of the decision-making process is illustrated in Figure 6.11.

Once all of the data for a coil has been loaded, a calculation is made within the webpage to determine whether the sum of the offsets made by both operators and adaption processes is greater than 12mm. High offsets can lead directly to Over Width or Under Width throughout the HSM process and can have an impact on the adaption models of successive strips. If the sum of these offsets is found to be greater than 12mm, the number is highlighted red, and is displayed once all data for the coil has finished loading, in the scheduling row and in the 'Coil N-1' and 'Coil N-2' columns of the web tool for easy visual reference.

Table 20: Summary of checks made and outputs produced by the tool's decision-making process.

Behaviour /Defect	**Type of check**	**What is checked?**	**What does the tool displayed?**
		Scheduling and Furnaces	
Problematic Grade	Conditional check	Checks if bar grade is in list of problematic grades	Grade highlighted red in table
Width Alarm Triggered	Conditional check	Checks if width alarm has been triggered due to bar not following Spread/Squeeze rules	Width alarm/spread-/squeeze limits highlighted red in table
Delay	Conditional check	Checks if bar has been delayed after Furnaces due to bottlenecks in subsequent subprocesses	Check highlighted red in table
Single Furnace Operation	Conditional check	Checks if only one furnace is in operation	Check highlighted red in table
		Roughing Mill	
Necking	Conditional checks followed by image classification	Checks for fundamental problems with sensors and width	RM Width data highlighted red in table and 'Necking' label

		before checking if Necking is present	shown in graph subtitle
Flare	Conditional check for outliers/abnormal behaviour followed by image classification	Checks for fundamental problems with width before checking if Flare is present	RM Width data highlighted red in table and 'Flare' label shown in graph subtitle
Roughing Mill Model Error	Conditional check	Checks if RM Model Error is >5mm or <-5mm	RM Model Error highlighted red in table
ANC Stroke Early/Disabled	Conditional check followed by image classification	ANC stroke timing is disabled before classifying between Early and Okay timing	Label displayed in table and highlighted red
Finishing Mill			
Under Width	Conditional check	Checks whether 10% or more of the strip's width is below lower tolerance	FM Width data highlighted red in table and 'Under Width' label shown in graph subtitle
Over Width	Conditional check	Checks whether 50% or more of the strip's width is above upper tolerance	FM Width data highlighted red in table and 'Over Width' label shown in graph subtitle
Width Pull	Image classification	Checks whether Width Pull is present	FM Width data highlighted red in table and 'Width Pull'

			label shown in graph subtitle
Finishing Mill Model Error	Conditional check (within web page)	Checks if FM Model Error is >5mm or <-5mm	FM Model Error highlighted red in table
High Temperature	Conditional check followed by classical ML	Checks for fundamental problems with temperature before checking if temperature is high	FM Temperature data highlighted red in table and ‘High Temperature’ label shown in graph subtitle
Low Temperature	Conditional check followed by classical ML	Checks for fundamental problems with temperature before checking if temperature is low	FM Temperature data highlighted red in table and ‘Low Temperature’ label shown in graph subtitle
	Coiler		
Under Width	Conditional check	Checks whether 10% or more of the strip’s width is below lower tolerance	DC Width data highlighted red in table and ‘Under Width’ label shown in graph subtitle
Over Width	Conditional check	Checks whether 50% or more of the strip’s width is above upper tolerance	DC Width data highlighted red in table and ‘Over Width’ label shown in graph subtitle

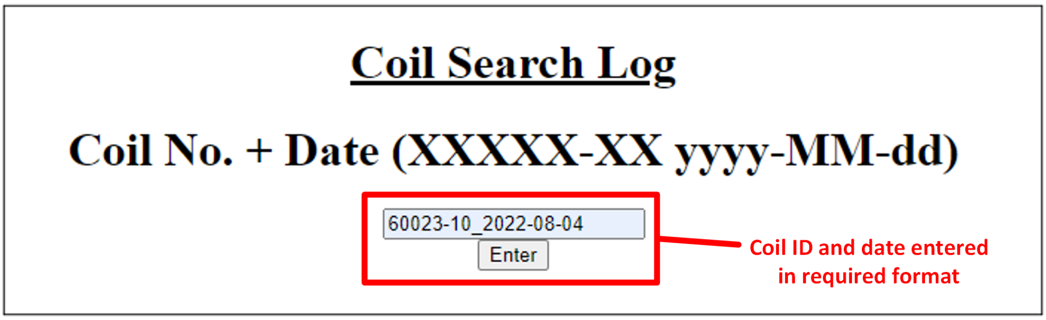
Coiler Snatch	Image classification	Checks whether Coiler Snatch is present	DC Width data highlighted red in table and ‘Coiler Snatch’ label shown in graph subtitle
High Temperature	Conditional check followed by classical ML	Checks for fundamental problems with temperature before checking if temperature is high	DC Temperature data highlighted red in table and ‘High Temperature’ label shown in graph subtitle
Low Temperature	Conditional check followed by classical ML	Checks for fundamental problems with temperature before checking if temperature is low	DC Temperature data highlighted red in table and ‘Low Temperature’ label shown in graph subtitle
Total Offset			
Sum of offsets too high/low	Conditional check	Checks whether the total sum of offsets made by HSM models and engineers is >12mm	Sum of offsets highlighted red in table

6.3 Web Tool Demo

6.3.1 Web Tool Overview

As mentioned in Section 6.2.1, the tool is created using a combination of PHP, HTML, and JavaScript, and is loaded in a local web environment using XAMPP. Loading times in a local environment are subject to the available computational power. In this particular experiment, the local web environment, and thus the tool, are run on a computer with 16GB RAM, 6GB VRAM, and a 24-core, 3GHz CPU. Initially, a Search page, based on the design in Figure 6.2, is loaded. Here, the user can input a coil ID and date in the requested format. If no information is entered, the program will default to the latest coil's information. Once the 'Enter' button is pressed, the program will load the Main page, based on the design in Figure 6.3.

The intended coil ID and date is first entered into the Coil Search page, shown in Figure 6.12. In the Main page data for this coil is loaded into the tool for each subprocess as described in Section 6.2.3 by each corresponding .exe program. Figure 6.13 shows an example of the full web tool interface having had data loaded and classified for several coils, while in the process of loading the most recent coil's Coiler subprocess data.



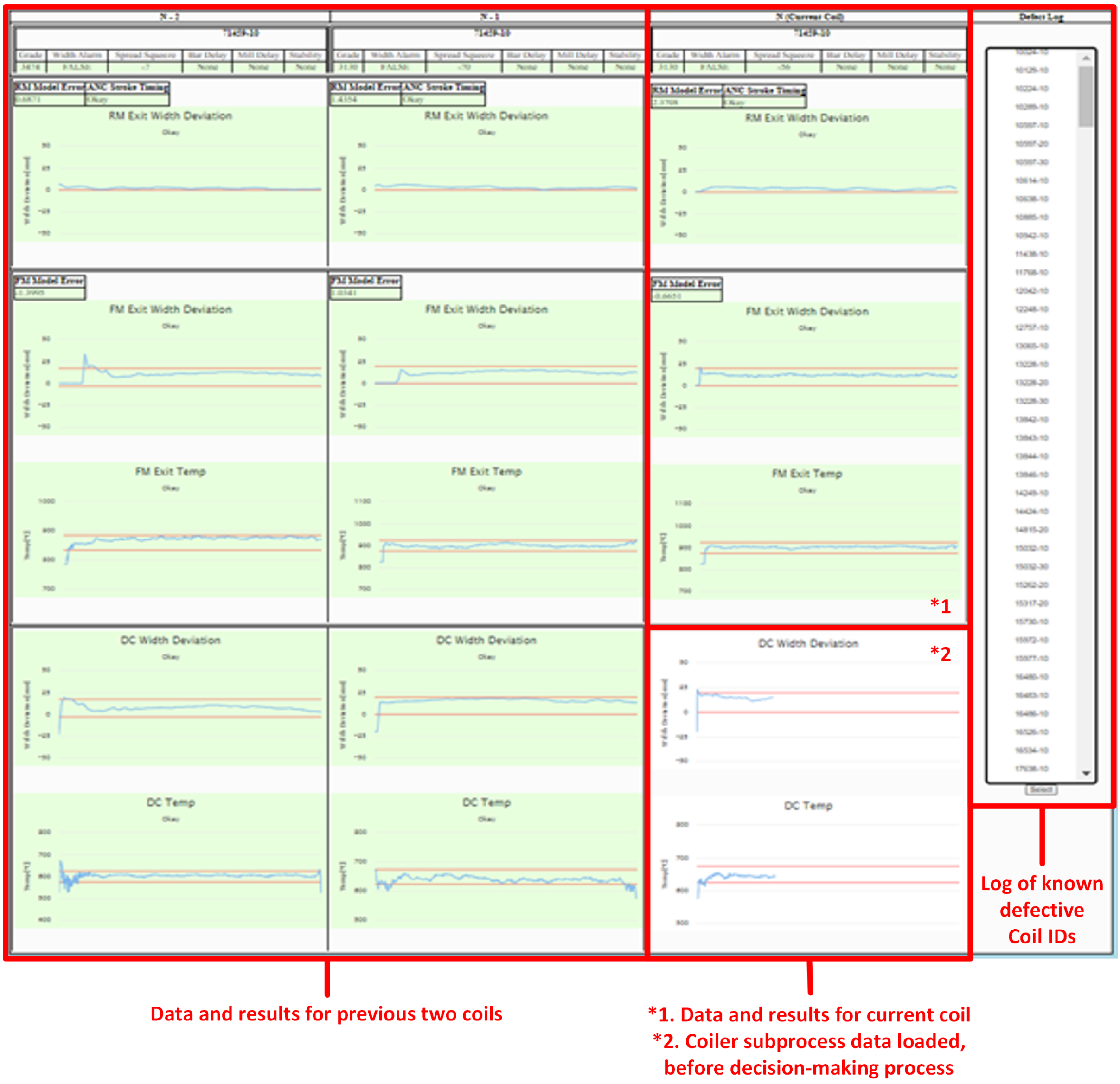
Coil Search Log

Coil No. + Date (XXXXX-XX yyyy-MM-dd)

60023-10_2022-08-04

Coil ID and date entered in required format

Figure 6.12: Interface of the Search Page of the proposed tool.



Log of known defective Coil IDs

Data and results for previous two coils

*1. Data and results for current coil
*2. Coiler subprocess data loaded, before decision-making process

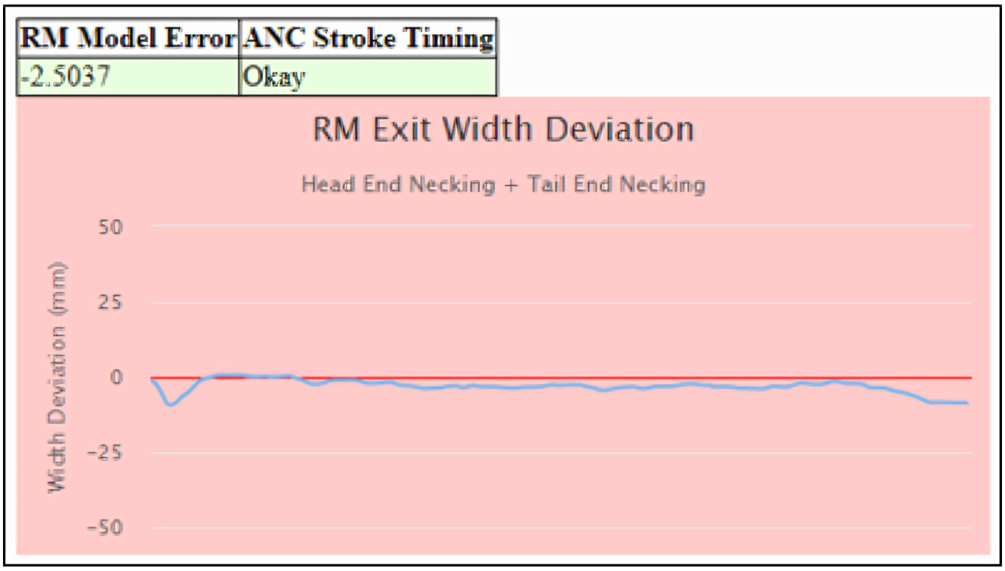
Figure 6.13: Interface of the proposed tool displaying visualized data from the HSM process. In this instance, Coiler subprocess data is being dynamically loaded, while data for previous coils and subprocesses has finished loading and thus been classified [1].

6.3.2 Examples and Results

After processing the newly collected samples, described in Table 19, through the tool, it was found that all defects were identified correctly by the new decision-making process such that defect labels matched those in the table. Some of the possible root causes, however, differed from those inputted by analysts into the root cause and comment fields. These fields can be reviewed in Table 4. Some examples of samples identified as having defects are shown and described in detail below.

6.3.2.1 Necking and Flare in the Roughing Mill Subprocess

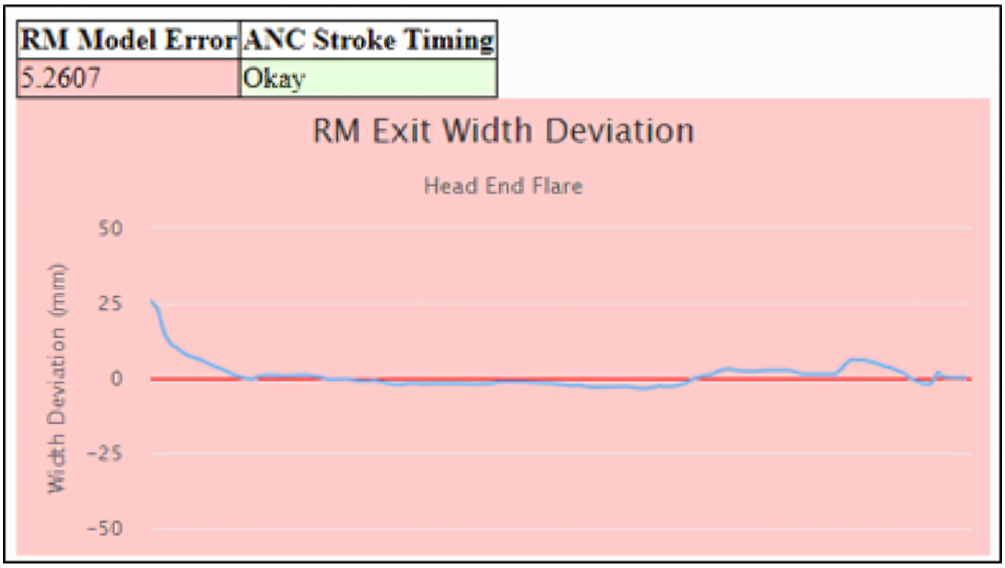
Figure 6.13 shows that once data from a previous coil has finished loading, it is moved into the column on its left to create room for the new coil's data. Figure 6.13 showcases full web tool with no width-related issues having being detected. The following figures show individual components of the web tool to focus on instances of various width-related defects and their possible root causes. Figure 6.15 shows an example, within the tool, of the output of the 'RM_Classify.exe' program, which utilises the new, data-driven decision-making process, when tested with a newly collected sample with Necking, described in Table 19. The graph subtitle 'Head End Necking + Tail End Necking' and the word 'Okay' under the 'ANC Stroke Timing' table column are both labels outputted by the program, while the highlighted colours are displayed by the tool itself according to these labels and the value shown under the 'RM Model Error' column. Equally, subsequent figures also display labels based on the outputs of their corresponding .exe programs and are highlighted by the program itself.



RM Model Error	ANC Stroke Timing
-2.5037	Okay

Figure 6.14: Example of a strip that has passed through the Roughing Mill subprocess that has been classified as having Necking [1].

The defect label outputted by the decision-making process correctly matches the sample's true label. Upon visual inspection of Figure 6.15, it can be concluded that the classification result for this instance is correct due to the width deviation at the head and tail ends of the bar. Despite this, no root causes appear to have been identified during either the Roughing Mill subprocess, Furnaces, or Scheduling. No root causes were recorded by analysts for this sample, however, meaning that the labels outputted by the tool's decision-making process remain correct. While it can also be confirmed that the result is correct from both visual inspecting the data used in the decision-making process and matching the original defect label, this instance shows that there may be other non-width-related factors in the Roughing Mill subprocess, Furnaces, or Scheduling, that may cause such an occurrence of Necking, and that more factors may be considered during future development of the web tool. Non-width-related factors should be considered in future work to ensure that such cases are taken into account and are included in an updated decision-making process.



RM Model Error	ANC Stroke Timing
5.2607	Okay

Figure 6.15: Example of a strip that has passed through the Roughing Mill subprocess that has been classified as having Flare due to model error [1].

It can be concluded that the classification result in the instance shown in Figure 6.15 is correct due to the steel increase in width at the head end of the bar. The root cause of this is a high Roughing Mill model error, as identified by the conditional check which highlights the Roughing Mill model error value red within the web tool.

6.3.2.2 Width Pull in the Finishing Mill Subprocess

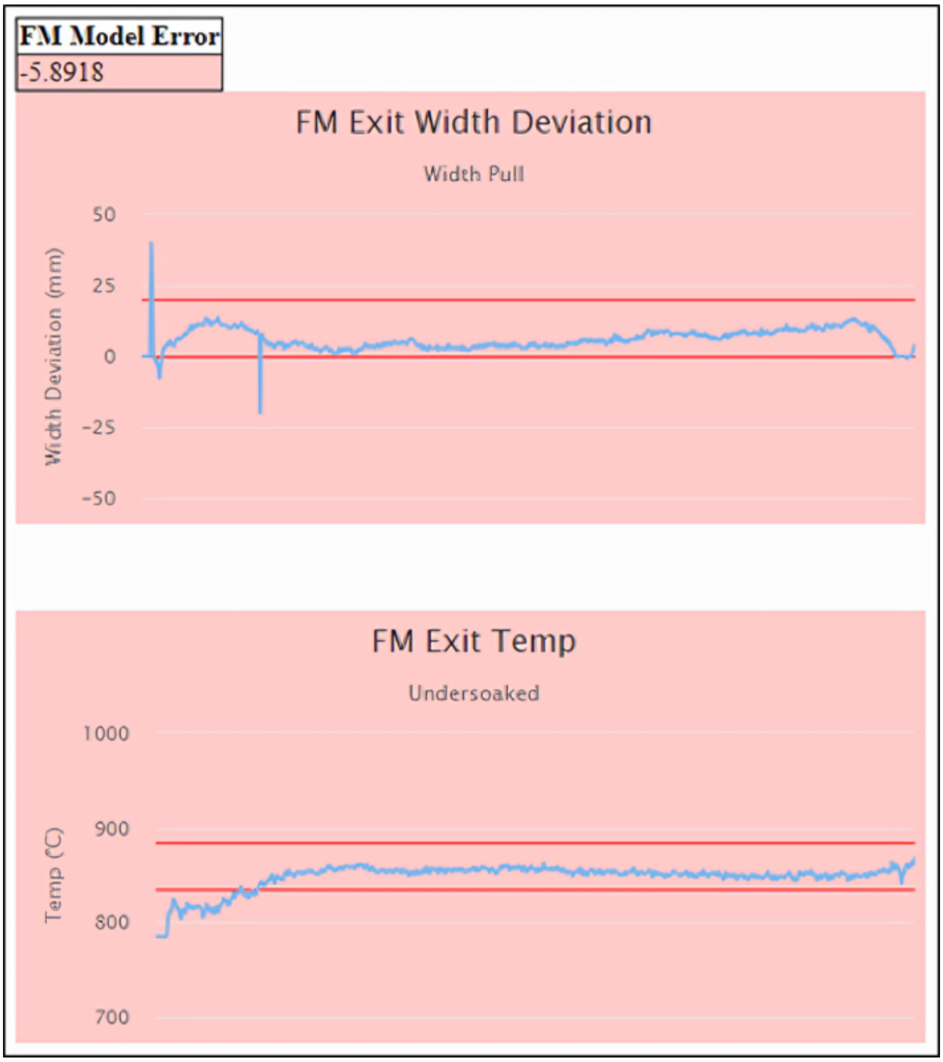


Figure 6.16: Example of a strip that has passed through the Finishing Mill subprocess that has been classified as having Width Pull and due to a combination of low temperatures and model error [1].

While the instance of Width Pull shown in Figure 6.16 appears to be more ambiguous than the width-related defects identified in the previous two examples, the sharp decrease in width at the head end of the strip clarified that Width Pull is present. The ambiguity in this instance arises from the generally low, though acceptable, width throughout the majority of the strip, making it possible that Width Pull is also present at the tail end of the strip where width drops slightly below the lower tolerance, although not sharply. This may be due to the fact that two root causes are identified within the Finishing Mill subprocess. Firstly, the model error has been found to be low, which suggests that not enough roller force was introduced to the strip by the model. This would explain the relatively low width throughout the body of the strip. The temperature of the strip was also found to be low at the head end of the strip. This would make the strip less malleable and thus more prone to sudden tension during rolling, explaining the sharp loss and recovery of width at the head end of the strip in the width deviation graph.

6.3.2.3 Snatch in the Coiler Subprocess

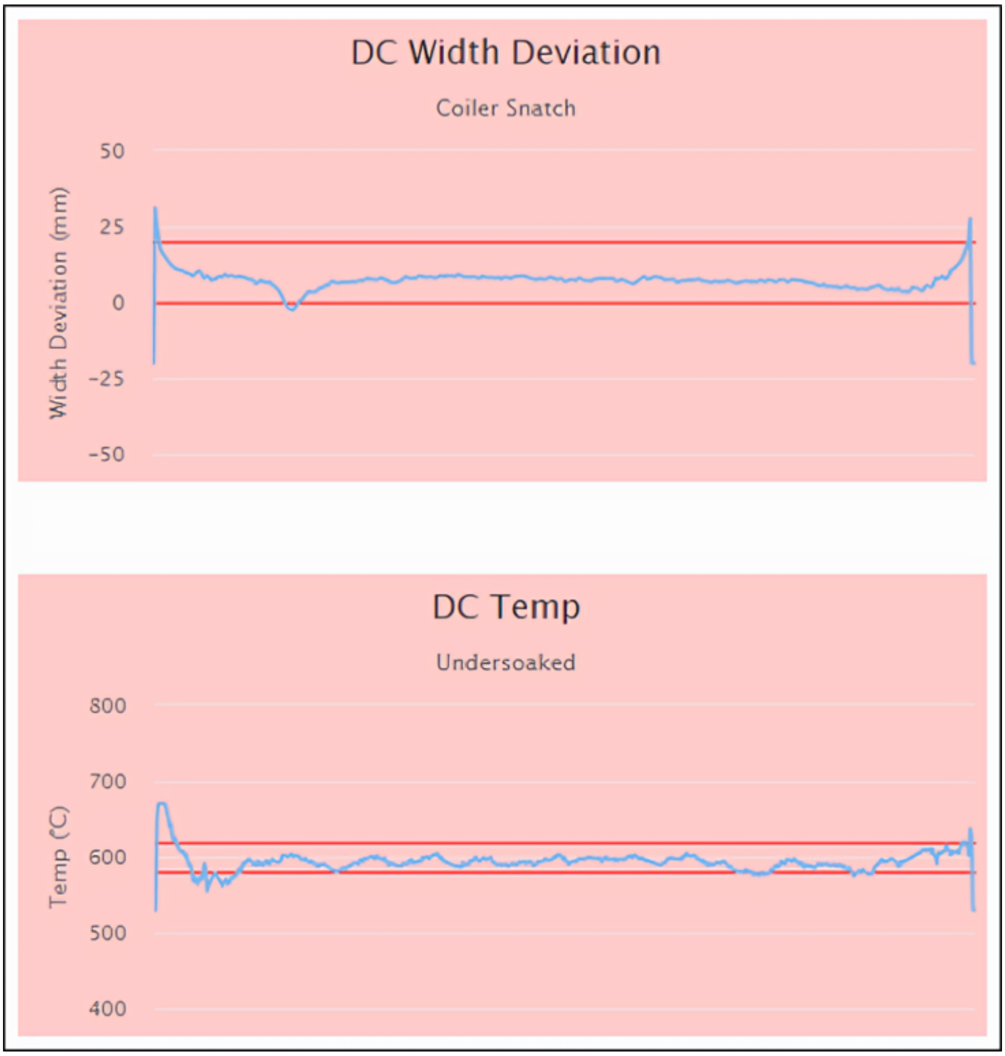


Figure 6.17: Example of a strip that has passed through the Coiler subprocess that has been classified as having Snatch due to low temperatures [1].

It can be concluded that the classification result for the instance shown in Figure 6.17 is correct due to the short, isolated drop in width in the body of the strip. It is possible that the cause of this is due to the temperature of the strip's body being generally low and inconsistent. Although temperature has been flagged as and may be a true contributing factor to the identified defect, it is possible that other non-width-related causes are present.

6.4 Summary and Conclusions

In this chapter, a fully-functional web tool was created for the purpose of simulating and visualising live HSM data and using a combination of expert knowledge and ML to determine the existence, cause, and origin of width-related defects. This is achieved by creating a local dataset derived from the HSM infrastructure, creating a local web page and environment using a combination of PHP, HTML, JavaScript, and XAMPP, and conditional checks and classifications included both within the web page and executable programs created using MATLAB's compiler application.

In Chapters 4 and 5, it was concluded that models performed optimally for their intended tasks [22, 23]. When comparing this to the model's performance on unseen data and within the web tool, these conclusions are further confirmed. It can therefore also be argued that it is possible to create and utilise an end-to-end methodology for defect detection and RCA within the HSM by combining expert knowledge and ML modes [1]. There is also potential to do this in a real-time manner such that process data can be loaded dynamically and classified simultaneously with minimal delay between a strip exiting a subprocess and feedback to operators.

It should be noted that only a handful of new defective instances could be collected from the HSM infrastructure as these were the only new samples available since creating the datasets using in Chapters 4 and 5. For clarification, the classification examples shown in Section 6.3.2 display these new instances when classified using the selected models from Chapters 4 and 5 with no new training data. As previously discussed, further training data would only increase model reliability. However, unless a Cross Validation method is used, a significant number of new instances should be collected before retraining the model and testing in the web tool to improve the reliability of classification results. This would be a priority in future work due to the challenge of limited data availability in this project.

The creation of the web tool in this chapter, along with the models created in Chapters 4 and 5, aim to showcase the potential process-wide tools and ML for defect detection and root cause analysis in steel manufacturing processes. They also aim to highlight the current restrictions and challenges, including limited data availability and human error, which may explain the currently limited state of applications in this setting. Future work would aim to integrate models and expert knowledge to detect further defective behaviours for other majorly impacting factors in the HSM process such as thickness and surface defects.

In the following chapter, conclusions are drawn on the work carried out in this thesis and an evaluation is made of its usefulness and contribution to the research field.

7 Conclusions

New research and technologies are being developed by steel companies world wide in an effort to be world-leading in outputting quality steel products. A large and important component of the innovation that stems from this effort is digitisation, particularly through the use of ML. This is because many large and long-established steel-making companies often have vast, ever-growing data stores which are often underutilised. Data is now seen as an asset by most companies in most industries and one of the most common uses of historical data is to utilise it within ML applications to make decisions based on patterns learned from previous behaviours. ML applications within steel-making, particularly the HSM process, are limited to surface quality inspection and roller force models, and these are typically conducted on a subprocesses-specific basis.

Recent research and development funded by Tata Steel and the UK Government aims to target this gap by addressing the challenges associated with the collection, storage, and processing of historical data and how it can be integrated and utilised to improve quality in an already well-established manufacturing process [1, 22, 23]. This research and development also aims to create tools which can be combined to improve decision-making and product quality across the entire HSM process, as opposed to only specific subprocesses [1].

In this thesis, the current state of steel-making, both globally and in the UK, was discussed and the HSM process was described as the basis for the experiments carried out in this thesis. This defined the problems associated with the move for innovation and quality, and with the limited application of ML within the steel-making industry, which are addressed by the experiments in this thesis. The potential and challenges of BDA were discussed along with ML technologies which were seen to have the potential to identify and resolve issues identified by HSM employees. In particular, Computer Vision and CNNs were considered as an approach for converting ML problems in image classification problems which attempt to emulate the visual decision-making process currently employed by HSM operators and analysts. Following this, a BDA approach was outlined and the methods and techniques chosen for use in the experiments carried out in this thesis were described in greater detail. This included chosen data sources, ML algorithms, optimisation and validation techniques, and technologies used for creating local web environments.

Practical experiments were then carried out in which ML models were created for the purpose of identifying width-related defects throughout the HSM process as well as their origins and their potential root causes. Specifically, three pre-trained CNN DL models were created to identify Necking and Flare in the Roughing Mill subprocess, Width Pull in the Finishing Mill subprocess, and Snatch in the Coiler subprocess through image classification. Another pre-trained CNN DL model for image classification, and a classical ML model, were created to identify Early ANC stroke timing in the Roughing Mill subprocess, and hot or cold temperatures in the Finishing Mill and Coiler subprocesses, respectively. Further to this, a set of rules was developed based on the knowledge gained from HSM operators and analysts. These rules were used to aid the defect detection and root cause analysis process to introduce further certainty and data utilisation into the newly-created decision-making process.

The collection of these models and expert knowledge were combined into a single, process-wide decision-making process and integrated into a web tool created using the Port Talbot HSM's web infrastructure. The final result is an end-to-end tool for the HSM process which can be used by operators and analysts as an aid for quickly identifying width-related defects and their root causes. This thesis showcases the potential of ML in the steel-making industry with regards to process-wide applications as well as its potential outside of surface defect detection and roller model optimisation. It also aims to show how the challenges of collecting, processing, and analysing big and historical data should be considered and how they can be approached or overcome.

7.1 Research Contributions

7.1.1 Review of Objectives

As discussed above, this thesis seeks to address a variety of challenges and gaps in research including those associated with the collection and processing of historical data, the limited use of ML in steel-making applications, and refining decision-making processes by combining ML and expert knowledge. The results of this have largely been communicated through journal and conference publications [1, 22, 23]. The research objectives of this thesis, outlined in Chapter 1, were originally created with the aim of address the challenges and gaps described above, and are evaluated below.

Objective 1: *Conduct a review into the potential and challenges of collecting and utilising data in a real-world manufacturing environment. Further to this, investigate and analyse the development and current state of AI and ML in the manufacturing and steel-making industry in order to identify technologies which could be used to address challenges and improve* *processes related to quality control in the Port Talbot HSM.*

In Chapter 2, a review was conducted for the purpose of understanding the potential, and challenges of big data with regards to its storage, collection, processing, and analysis. I4.0 was discussed and this provided an insight into current automation and digitisation standards in manufacturing. With BDA being a component of I4.0, the characteristics of big data, defined by the 5Vs were explored. By understanding the 5Vs, an understanding was gained of the challenges of working with data under the conditions that they have been historical stored in and are currently collected in today. This, in combination with the standard framework for BDA described in Section 2.1.2, provided an approach which was followed to conduct the practical experiments later in the thesis.

The knowledge gained from this encouraged thoroughness in the selection of methods and techniques identified in the following research, and awareness of potential issues when collecting and processing data in the practical experiments based on the characteristics of big data. Understanding the potential of collecting and utilising data in real-world manufacturing environments also enabled a better understanding of the goals and outcomes of the work produced in this thesis with regards to the application and the industry itself.

Following this research, the history and development of AI and ML in manufacturing and the steel industry was discussed. This aided in understanding how technologies developed from manual and Lean 6 Sigma methodologies through to the integration of AI for expert and prescriptive systems, and how various ML approaches have been used to identify and analyse various types of issues based on the type of data available. This also aided in identifying the fundamental gaps in research and development with regards to AI and ML in the steel-making industry. In particular, that applications of ML are limited in the HSM process, a process which Tata Steel has identified to be under-utilising data for quality improvement, and that applications are typically focused on a singular aspect of the mill with little feedback throughout the entire process, leading to potential gaps and losses in information downstream when analysing process data.

A detailed overview of the fundamentals of ML, ANNs, and Computer Vision was also provided in Chapter 2. In further subsections, specific technologies such as DL, CNNs, and Image Classification were also described in detail. These discussions served as a foundation on which technologies could be identified for identifying width defects in the HSM process based on the nature of the data available in the Port Talbot HSM. These specific technologies, and the reasoning for their selection for use in the practical experiments carried out in this thesis, were discussed in Chapter 3. This chapter also outlined the methodology used to carry out these experiments.

The current state of applications of AI and ML in steel-making, particularly in the HSM process, is limited. While this makes the findings and practical output of this thesis novel and a push for innovation in this field, it also makes evaluation against existing tools difficult. This is not only due to the limited areas of application, namely surface defect detection and roller model optimisation, but also because these applications are limited to single subprocesses within the HSM process. The models and tool developed in this thesis identify width defects and root causes which existing applications have not addressed in multiple HSM subprocesses and combine their output to produce an informed analysis of width performance over the entire HSM process.

Objective 2: *Gather expert knowledge and develop novel tools for the detection and analysis of width defects and their root causes using those technologies reviewing in the previous objective.*

As previously mentioned, Tata Steel has identified the HSM as a process in which data has much potential but is underutilised for quality control purposes. Width-related defects form a significant portion of the total defects that occur in this process and thus these types of defect were selected as the focus of experiments and analysis in this thesis.

In Chapter 4, a detailed understanding was gained of all possible width-related defects which can occur in the various subprocesses throughout the HSM process. This was achieved through a combination of research and regular communication with operators and analysts in the Port Talbot HSM. This process was also repeated to gain an understanding of the potential root causes of thesis defects in Chapter 5. Attaining this expert knowledge resulted in three significant benefits. Firstly, a thorough understanding of how data is collected, processed, and analysed was gained. This enabled us to determine which data was to be used in the ML models created at the end of these chapters and how it was to be processed. Secondly, it enabled the creation of a set of rules used for simple checks and analysis which do not require the use of ML. This was combined with the ML models created with these chapters and included in the web tool created in Chapter 6, forming a fundamental part of the decision-making process for identifying width defects and their root causes. Lastly, operators and analysts in the HSM were able to provide additional information on abnormal or complex occurrences of width defects such as the effects of external factors, including weather and dirt, on the bar and equipment which can unexpectedly impact their performance throughout the HSM process. These occurrences are difficult to identify without expert knowledge as they are rare occurrences and are not currently included as formal analyses in the HSM. This information also became useful when analysing the results of the created ML models when assessing the potential reasons for misclassifications.

ML models were created for those width defects which required the analysis of ambiguous data. Computer Vision and Image Classification was used for most of these models to emulate the current process of visually inspecting time series data. Specifically, the pre-trained CNN architecture, GoogLeNet, was used in combination with Transfer Learning to emulate analysts' visual inspection of time series data. A traditional ML learning algorithm, a Fine KNN, was also used to train a temperature classification model. This was selected in this case as opposed to an image classification due to the simplicity of the features needed to make the classification. While the test results of all models showed that they are fit for the intended scenario, reliability can and should be continuously confirmed with further data collection and evaluation. The misclassifications shown in some of these results should, however, also be taken into account. As mentioned, expert knowledge proved useful here as further investigation into the behaviours leading to defects was required to understand why these misclassification occurred. These tools have expanded the existing range of ML applications in steel-making and, more specifically, the HSM process. Along with the expert knowledge gained in these chapters, they have also shown the potential of combining ML tools and expert knowledge for process-wide quality inspection.

Limited data availability was the most prominent challenge when creating the tools required to complete this objective, making it difficult to validate and evaluate models effectively. It did, however, also justify the use of Transfer Learning which was used to combat this issue by utilising the context and performance of a readily trained CNN. As discussed in Section 3.3.1.2, GoogLeNet was chosen as the pre-trained CNN architecture for the experiments in this thesis over other available and simpler architectures due to its depth and its use of inception modules which allow it to use a larger number of convolutional layers with less data loss [8, 90, 164], reducing the possibility of underfitting and overfitting. As concluded in Sections 4.6 and 5.5, further experiments should be carried out to further confirm the reliability of the created models once more quality data becomes available in the Port Talbot HSM.

Although available data for ML experiments was limited, future work would aim to improve and confirm their reliability of the models with more data. Further expert knowledge would also be gathered in order to further formalise the current process of analysing ambiguous data when analysing defects. It should be noted that while the created models may not directly be used to address other defects, it may be possible to utilise the same methodology and technologies to create new models to detect other defects in the HSM along with their root causes, further to width-related defects focused on in this thesis, adding breadth to the created tool.

Objective 3: *Develop a tool, using the Port Talbot HSM's web infrastructure, by combining the knowledge gained and models created from the previous objective to identify width defects and their potential root causes throughout the entire HSM process.*

As evaluated in the previous objective, several fit-for-purpose machine learning models were created as a result of the experiments carried out in chapters 4 and 5, and expert knowledge was gained which enabled the identification of width defects and their root causes throughout the HSM process. In Chapter 6, the current decision-making process was formalised such that the knowledge gained and the models created could be combined and integrated into a single tool to streamline and speed up the detection, analysis, and feedback for these defects.

New, unseen samples were gathered using the same Data Collection methodology used in the previous chapters. This allowed for further testing of model reliability within the setting of the created tool. The tool was also created using a combination of HTML, PHP, and JavaScript which are the same technologies used to construct tools and pages in the Port Talbot HSM web infrastructure. This enables simpler integration of the created tool into the current Port Talbot HSM web infrastructure for future deployment. Due to limitations on access permissions, the tool was run locally such that data was not streamed live from the Port Talbot HSM, but it was loaded such that data can be accessed and visualised within the tool to simulate it being streamed as strips progress through the HSM process. Once the data for each subprocess has finished loading, an analysis is run to determine whether defects are present and, if so, their potential root causes, and feedback of these analyses are displayed immediately in the corresponding section in the web tool. The tool thus provides immediate feedback to operators such that mitigating actions can be taken before subsequent strips are negatively affected. This information can also become useful to analysts should they need to examine the samples identified as defective at a later date.

While integrated decision-making process and the web tool generally performed well and showed no misclassifications in the results of the final experiment, it is important to keep in mind the misclassifications that occurred in earlier chapters. While results in these chapters showed that the models are sufficient for the intended scenario, those mislcassifications caused by rare, ambiguous factors may still occur since the same models are being used. While expert knowledge can be used to address this when manually inspecting the relevant data, future work would attempt to address with two approaches. Firstly, further rules could be added to the formalised decision-making process to account for these particular scenarios. This approach, however, may become perpetual as rules may need to be continuously added based on specific scenarios. Therefore, the second approach would be to continuously update and test the models with newly available data overtime in order to confirm its reliability and thus improve its performance.

7.1.2 Novel Field Contributions

This project has made the following novel contributions:

1. Novel DL models were created to identify width-related issues, expanding the scope of defect detection and root cause analysis applications in the HSM process.

2. A data-driven framework to improve decision-making was proposed. The novelty of this approach lies in the combination of expert knowledge and multiple DL models for defect detection and root cause analysis of width-related defects across the entire HSM process.

3. To demonstrate the application of this approach, a bespoke web tool was developed by integrating HSM data and the developed data-driven framework. The tool was developed for the purpose of providing HSM operators and analysts with quick and effective feedback.

7.2 Future Work

This thesis provides an insight into both the challenges and potential of utilising industrial data and ML for defect analysis in steel-making processes, and provides practical examples which serve as a step towards further innovation and process-wide applications, particularly within the HSM process. The main challenges associated with such a task were derived from data availability, access permissions, and utilisation of expert knowledge.

Data availability was the prominent challenge tackled throughout this thesis. While overall there were a large number of labelled defective occurrences available for which to retrieve corresponding data, the number of samples is generally considered small for ML problems. As discussed, while experiment results may show that models perform suitably for the intended purpose, misclassifications and reliability testing with further available data. This could even be carried out in conjunction with a full or partial deployment of the created web tool as a semi-supervised ML method [1]. In this case, the tool can be used by operators and analysts to detect and label new instances of defects and their root causes before using these instances as further samples to retrain the model. This process would be repeated to continuously improve the model and assess its reliability.

The final chosen set of data used in the practical aspects of this thesis were also derived from data which operators and analysts showed that they use to make their current manual analyses of the width defects and root causes described in Chapters 3 and 4. Future work may also aim to conduct research into whether other data is available which can complement the data currently used in these experiments and analyses, potentially making their results more robust and reliable.

It should also be noted that a majority of this thesis was carried out during and after the COVID-19 pandemic, which further highlighted the challenges of collecting data and expert knowledge quickly, efficiently, and at scale with limited access to both the physical HSM process and limited interaction with HSM personnel due to site restrictions and furlough schemes. If this thesis were to be repeated, or if further work were to be carried out, under circumstances in which these restrictions do not apply, further time would be spent physically at the Port Talbot site itself to resolve the consequences of these issues.

Further challenges and aims that future work would aim to address and accomplish are:

- Further collection and utilisation of expert knowledge, which has been shown in this thesis to be invaluable when analysing ambiguous defect occurrences. Such knowledge again be formalised in such a way that further ambiguous cases, such as those shown in the experiment results in Chapters 4 and 5, would be addressed in a new iteration of the web tool. This would include investigations into defects whose root causes were not identified or recorded by operators or analysts.

- Inclusion of Natural Language Processing technologies to aid in obtaining domain knowledge

- Further collection of defective instances for continued improvement and assessment of the performance and reliability of the created ML models and web tool's decision-making process.

- Further collection of expert knowledge and data for the purpose of creating new models for other types of defects in the HSM process including thickness and surface defects. These would be integrated into an updated decision-making process such that all models and gathered knowledge continue to contribute to a process-wide analysis.

- A further case study on another process in Port Talbot Steelworks independent of the HSM, further showcasing the potential of ML applications for process-wide defect analysis in the steel-making industry and progressing Tata Steel's stance in the global race of innovation and quality perfection.

Bibliography

[1] S. Latham and C. Giannetti, “A tool to combine expert knowledge and machine learning for defect detection and root cause analysis in a hot strip mill,” *SN Computer Science*, vol. 4, p. 628, 2023.

[2] M. D. de Assunção, R. N. Calheiros, S. C. S. Bianchi, M. A. S. Netto, and R. Buyya, “Big data computing and clouds: Trends and future directions,” *J. Parallel Distributed Comput.*, vol. 79-80, pp. 3–15, 2013.

[3] Acatech, G. Schuh, R. Anderl, J. Gausemeier, M. ten Hompel, and W. Wahlster, “Industrie 4.0 Maturity Index (Managing the Digital Transformation of Companies).” 2017.

[4] S. Agatonovic-Kustrin and R. Beresford, “Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research,” *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, no. 5, pp. 717–727, 2000.

[5] L. Alzubaidi, M. Fadhel, O. Al-Shamma, J. Zhang, J. Santamaría, Y. Duan, and S. Oleiwi, “Towards a better understanding of transfer learning for medical imaging: A case study,” *Applied Sciences*, vol. 10, p. 4523, 06 2020.

[6] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” *Neural Information Processing Systems*, vol. 25, 01 2012.

[7] X. Ou, P. Yan, Y. Zhang, B. Tu, G. Zhang, J. Wu, and W. Li, “Moving object detection method via resnet-18 with encoder–decoder structure in complex scenes,” *IEEE Access*, vol. 7, pp. 108152–108160, 2019.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–9, IEEE, 2015.

[9] Q. Ren, M. Li, and S. Han, “Tectonic discrimination of olivine in basalt using data mining techniques based on major elements: a comparative study from multiple perspectives,” *Big Earth Data*, vol. 3, pp. 1–18, 02 2019.

[10] H.-K. Hsu and J.-N. Aoh, “The mechanism of position-mode side guide in correcting camber in roughing process of a hot strip mill,” *Metals*, vol. 9, no. 5, 2019.

[11] H.-K. Hsu and J.-N. Aoh, “Effect of clearances in mill stands on strip end motion during finishing rolling,” *Metals*, vol. 9, no. 7, 2019.

[12] W. S. Association, “World steel in figures 2023,” 2023.

[13] J. M. Cullen, J. M. Allwood, and M. D. Bambach, “Mapping the global flow of steel: From steelmaking to end-use goods,” *Environmental Science Technology*, vol. 46, pp. 13048–13055, 12 2012. doi: 10.1021/es302433p.

[14] F. Li, A. He, Y. Song, Z. Wang, X. Xu, S. Zhang, Y. Qiang, and C. Liu, “Deep learning for predictive mechanical properties of hot-rolled strip in complex manufacturing systems,” *International Journal of Minerals, Metallurgy and Materials*, vol. 30, pp. 1093–1103, 2023.

[15] E. . I. S. D. for Business, “Future capacities and capabilities of the uk steel industry,” 2017.

[16] M. Keep, I. Jozepa, and M. Ward, “Contribution of the steel industry to the uk economy,” 2023.

[17] C. C. Committee, “Progress report: Reducing emissions in wales,” 2023.

[18] W. Government, “Wellbeing of wales, 2023,” 2023.

[19] D. for Energy Security Net Zero, “Uk local authority greenhouse gas emissions estimates 2021,” 2023.

[20] D. Cemernek, S. Cemernek, H. Gursch, A. Pandeshwar, T. Leitner, M. Berger, G. Klösch, and R. Kern, “Machine learning in continuous casting of steel: a state-of-the-art survey,” *Journal of Intelligent Manufacturing*, vol. 33, pp. 1–19, 08 2022.

[21] H. Kildahl, L. Wang, L. Tong, and Y. Ding, “Cost effective decarbonisation of blast furnace – basic oxygen furnace steel production through thermochemical sector coupling,” *Journal of Cleaner Production*, vol. 389, p. 135963, 2023.

[22] S. Latham and C. Giannetti, "Pre-Trained CNN for Classification of Time Series Images of Anti-Necking Control in a Hot Strip Mill," in *Proceedings of 9th IIAE International Conference on Industrial Application Engineering (ICIAE2021)*, pp. 77–84, 2021.

[23] S. Latham and C. Giannetti, "Root Cause Classification of Temperature-Related Failure Modes in a Hot Strip Mill," in *3rd International Conference on Innovative Intelligent Industrial Production and Logistics*, pp. 36–45, 2022.

[24] N. Elgendy and A. Elragal, "Big data analytics: A literature review paper," in *Advances in Data Mining. Applications and Theoretical Aspects* (P. Perner, ed.), (Cham), pp. 214–227, Springer International Publishing, 2014.

[25] "Data Science and Big Data Analytics - Introduction to Big Data Analytics," John Wiley & Sons, 2014.

[26] Y. Demchenko, C. D. Laat, and P. Membrey, "Defining Architecture Components of the Big Data Ecosystem," *2014 International Conference on Collaboration Technologies and Systems (CTS)*, vol. 10, pp. 104–112, 2014.

[27] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "Big Data technologies: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018.

[28] R. Kitchin and G. McArdle, "What makes big data, big data? exploring the ontological characteristics of 26 datasets," *Big Data Society*, vol. 3, pp. 6–8, 3 2016.

[29] G. Vranopoulos, N. Clarke, and S. Atkinson, "Addressing big data variety using an automated approach for data characterization," *Journal of Big Data*, pp. 1–3, 3 2022.

[30] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.

[31] W. A. Günther, M. H. Rezazade Mehrizi, M. Huysman, and F. Feldberg, "Debating big data: A literature review on realizing value from big data," *The Journal of Strategic Information Systems*, vol. 26, no. 3, pp. 191–209, 2017.

[32] X. Jin, B. W. Wah, X. Cheng, and Y. Wang, "Significance and Challenges of Big Data Research," *Big Data Research*, vol. 2, no. 2, pp. 59–64, 2015.

[33] M. Lycett, "'Datafication': Making sense of (big) data in a complex world," 2013.

[34] S. Garcia, S. Ramirez-Gallego, J. Luengo, J. Manuel Benitez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Analytics*, vol. 1, no. 1, pp. 1–22, 2016.

[35] J. J. Berman, *Principles of big data: preparing, sharing, and analyzing complex information.* Newnes, 2013.

[36] M. Obermeyer, Ziad and P. Emanuel, Ezekiel J., M.D., "Predicting the Future - Big Data, Machine Learning, and Clinical Medicine," *New England Journal of Medicine*, vol. 375, no. 13, pp. 1212–1216, 2016.

[37] M. D. Smith and R. Telang, *Streaming, Sharing, Stealing: Big Data and the Future of Entertainment.* MIT Press, 2016.

[38] Z. Tufekci, "Big Questions for Social Media Big Data : Representativeness , Validity and Other Methodological Pitfalls," pp. 505–514.

[39] N. Mehta and A. Pandit, "Concurrence of big data analytics and healthcare: A systematic review," *International journal of medical informatics*, vol. 114, pp. 57–65, 2018.

[40] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.

[41] D. M. West, "Big Data for Education: Data Mining, Data Analytics, and Web Dashboards," *Governance Studies at Brookings*, vol. 4, pp. 1–10, 2012.

[42] J. Zeng and J. Jia, "The Impact of Big Data on School Sports and Competitive Sports," *2017 Chinese Automation Congress (CAC)*, pp. 596–599, 2017.

[43] E. Raguseo, "Big data technologies: An empirical investigation on their adoption, benefits and risks for companies," *International Journal of Information Management*, vol. 38, no. 1, pp. 187–195, 2018.

[44] J. Krumeich, D. Werth, P. Loos, J. Schimmelpfennig, and S. Jacobi, "Advanced planning and control of manufacturing processes in steel industry through big data analytics: Case study and architecture proposal," *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, pp. 16–24, 2014.

[45] F. Provost and T. Fawcett, "Data Science and its Relationship to Big Data and Data-Driven Decision Making," *Big Data*, vol. 1, no. 1, pp. 51–59, 2013.

[46] E. Brynjolfsson and K. McElheran, "The rapid adoption of data-driven decision-making," *American Economic Review*, vol. 106, no. 5, pp. 133–139, 2016.

[47] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient Machine Learning for Big Data: A Review," *Big Data Research*, vol. 2, no. 3, pp. 87–93, 2015.

[48] J. Lee, H. A. Kao, and S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.

[49] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255 – 260, 2015.

[50] D. O. Hebb, *The organization of behavior; a neuropsychological theory.* The organization of behavior; a neuropsychological theory., Oxford, England: Wiley, 1949.

[51] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[52] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[53] P. Cunningham and S. Delany, "k-Nearest neighbour classifiers," *Mult Classif Syst*, 2007.

[54] M. B, *Clustering: A data recovery approach.* second ed., 2016.

[55] S. Linnainmaa, "Taylor expansion of the accumulated rounding error," *BIT Numerical Mathematics*, vol. 16, pp. 146–160, Jun 1976.

[56] E. Commission, J. R. Centre, B. Delipetrev, C. Tsinaraki, and U. Kostić, *AI watch, historical evolution of artificial intelligence – Analysis of the three main paradigm shifts in AI.* Publications Office, 2020.

[57] S. Loussaief and A. Abdelkrim, "Machine learning framework for image classification," in *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pp. 58–61, IEEE, 2016.

[58] J. Padmanabhan and M. J. Johnson Premkumar, "Machine learning in automatic speech recognition: A survey," *IETE Technical Review*, vol. 32, no. 4, pp. 240–251, 2015.

[59] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 67, 2016.

[60] T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," in *The elements of statistical learning*, pp. 9–41, Springer, 2009.

[61] H. S. R. Rajula, V. Giuseppe, M. Manchia, N. Antonucci, and V. Fanos, "Comparison of conventional statistical methods with machine learning in medicine: Diagnosis, drug development, and treatment," *Medicina Journal*, vol. 56, 09 2020.

[62] I. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Computer Science*, vol. 2, no. 3, 2021.

[63] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver, "Review of tool condition monitoring in machining and opportunities for deep learning," *The International Journal of Advanced Manufacturing Technology*, pp. 1–22, 2020.

[64] M. Berry and A. Mohamed, *Supervised and Unsupervised Learning for Data Science.* 2020.

[65] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning.* The MIT Press, 1st ed., 2010.

[66] A. Kusiak, "Smart manufacturing," *International Journal of Production Research*, vol. 56, no. 1-2, pp. 508–517, 2018.

[67] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine learning for industrial applications: A comprehensive literature review," *Expert Systems with Applications*, vol. 175, p. 114820, 2021.

[68] Q. Wu, "Product demand forecasts using wavelet kernel support vector machine and particle swarm optimization in manufacture system," *Journal of Computational and Applied Mathematics*, vol. 233, no. 10, pp. 2481–2491, 2010.

[69] S. Ravikumar, K. Ramachandran, and V. Sugumaran, "Machine learning approach for automated visual inspection of machine components," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3260–3266, 2011.

[70] A. Saxena and A. Saad, "Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems," *Applied Soft Computing*, vol. 7, no. 1, pp. 441–454, 2007.

[71] C. I. Thorsten Wuest, Daniel Weimer and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.

[72] R. I. Mukhamediev, Y. Popova, Y. Kuchin, E. Zaitseva, A. Kalimoldayev, A. Symagulov, V. Levashenko, F. Abdoldina, V. Gopejenko, K. Yakunin, E. Muhamedijeva, and M. Yelis, "Review of artificial intelligence and machine learning technologies: Classification, restrictions, opportunities and challenges," *Mathematics*, vol. 10, no. 15, 2022.

[73] K. Gurney, *An Introduction to Neural Networks.* USA: Taylor & Francis, Inc., 1997.

[74] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1139–1147, PMLR, 2013.

[75] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook.* Springer Publishing Company, Incorporated, 1st ed., 2018.

[76] W. C. Wong, E. Chee, J. Li, and X. Wang, "Recurrent neural network-based model predictive control for continuous pharmaceutical manufacturing," *Mathematics*, vol. 6, no. 11, 2018.

[77] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 2, pp. 135–142, 2017.

[78] K. Lakshmanan, E. Borghini, A. Beckmann, C. Pleydell-Pearce, and C. Giannetti, "Data modelling and remaining useful life estimation of rolls in a steel making cold rolling process," *Procedia Computer Science*, vol. 207, pp. 1057–1066, 2022.

Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 26th International Conference KES2022.

[79] A. Jamwal, R. Agrawal, and M. Sharma, "Deep learning for manufacturing sustainability: Models, applications in industry 4.0 and implications," *International Journal of Information Management Data Insights*, vol. 2, no. 2, p. 100107, 2022.

[80] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018. Special Issue on Smart Manufacturing.

[81] H. Gajjar, S. Sanyal, and M. Shah, "A comprehensive study on lane detecting autonomous car using computer vision," *Expert Systems with Applications*, vol. 233, p. 120929, 2023.

[82] A. Esteva, K. Chou, S. Yeung, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, "Deep learning-enabled medical computer vision," *npj Digital Medicine*, vol. 4, p. 5, 12 2021.

[83] P. Kaur, K. Krishan, S. K. Sharma, and T. Kanchan, "Facial-recognition algorithms: A literature review," *Medicine, Science and the Law*, vol. 60, no. 2, pp. 131–139, 2020. PMID: 31964224.

[84] L. Zhou, L. Zhang, and N. Konz, "Computer vision techniques in manufacturing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 1, pp. 105–117, 2023.

[85] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning.," *IEEE transactions on medical imaging*, vol. 35, pp. 1285–1298, may 2016.

[86] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, pp. 84–90, may 2017.

[87] R. Yamashita, M. Nishio, R. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, 06 2018.

[88] J. Wang, L.-C. Yu, K. Lai, and X. Zhang, “Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 225–230, 2016.

[89] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[90] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021.

[91] H. Sharma, H. Kumar, A. Gupta, and M. A. Shah, “Computer vision in manufacturing: a bibliometric analysis and future research propositions,” *The International Journal of Advanced Manufacturing Technology*, vol. 127, 07 2023.

[92] J. Jiao, M. Zhao, J. Lin, and K. Liang, “A comprehensive review on convolutional neural network in machine fault diagnosis,” *Neurocomputing*, vol. 417, pp. 36–63, 2020.

[93] Y. C. Liang, W. D. Li, X. Lu, and S. Wang, “Fog computing and convolutional neural network enabled prognosis for machining process optimization,” *Journal of Manufacturing Systems*, vol. 52, no. May, pp. 32–42, 2019.

[94] S. Khokhar, A. A. Mohd Zin, A. P. Memon, and A. S. Mokhtar, “A new optimal feature selection algorithm for classification of power quality disturbances using discrete wavelet transform and probabilistic neural network,” *Measurement: Journal of the International Measurement Confederation*, vol. 95, pp. 246–259, 2017.

[95] S. Wang and H. Chen, “A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network,” *Applied Energy*, vol. 235, no. March 2018, pp. 1126–1140, 2019.

[96] Microsoft Corporation, “Azure machine learning.” `https://azure.microsoft.com/en-gb/products/machine-learning`.

[97] Amazon Web Services, “Amazon sagemaker.” `https://aws.amazon.com/sagemaker/`.

[98] Edge Impulse, “Edge impulse.” v1.66.12, 2025-01-09, `https://edgeimpulse.com/`.

[99] V. R. Sreedharan and R. Raju, "A systematic literature review of Lean Six Sigma in different industries," *International Journal of Lean Six Sigma*, vol. 7, no. 4, pp. 430–466, 2016.

[100] O. Serrat, *The Five Whys Technique*, pp. 307–310. Springer, 1st editio ed., 2017.

[101] E. D. Arnheiter and J. E. Greenland, "Looking for root cause: a comparative analysis," *The TQM journal*, vol. 20, no. 1, pp. 18–30, 2008.

[102] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0," *Information Fusion*, vol. 50, pp. 92–111, 2019.

[103] I. Yaqoob, I. A. T. Hashem, A. Gani, S. Mokhtar, E. Ahmed, N. B. Anuar, and A. V. Vasilakos, "Big data: From beginning to future," *International Journal of Information Management*, vol. 36, no. 6, Part B, pp. 1231–1247, 2016.

[104] S. Madden, "From Databases to Big Data," *IEEE Internet Computing*, vol. 16, no. 3, pp. 4–6, 2012.

[105] Z. Cinar, A. Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0," *Sustainability*, vol. 12, no. 19, p. 8211, 2020.

[106] A. Dogan and D. Birant, "Machine learning and data mining in manufacturing," *Expert Systems with Applications*, vol. 166, p. 114060, 2021.

[107] A. Essien and C. Giannetti, "A Deep Learning Framework for Univariate Time Series Prediction Using Convolutional LSTM Stacked Autoencoders," pp. 1–6, 2019.

[108] C. Giannetti and A. Essien, "Towards Scalable and Reusable Predictive Models for Cyber Twins in Manufacturing Systems," *J. Intell. Manuf.*, vol. 33, pp. 441–455, feb 2022.

[109] N. Neogi, D. Mohanta, and P. Dutta, "Review of vision-based steel surface inspection systems," *Journal of Computing in Higher Education*, vol. 2014, 11 2014.

[110] K. B. Chigateri and A. M. Hebbale, "A steel surface defect detection model using machine learning," *Materials Today: Proceedings*, vol. 100, pp. 51–58, 2024. 2nd International Conference on Materials Science and Mathematics for Advanced Technology (MSMAT 2022).

[111] S. Y. Lee, B. A. Tama, S. J. Moon, and S. Lee, "Steel surface defect diagnostics using deep convolutional neural network and class activation map," *Applied Sciences*, vol. 9, no. 24, 2019.

[112] H. Zhang, S. Li, Q. Miao, R. Fang, S. Xue, Q. Hu, J. Hu, and S. Chan, "Surface defect detection of hot rolled steel based on multi-scale feature fusion and attention mechanism residual block," *Scientific Reports*, vol. 14, pp. 1–16, 04 2024.

[113] D. He, K. Xu, and P. Zhou, "Defect detection of hot rolled steels with a new object detection framework called classification priority network," *Computers Industrial Engineering*, vol. 128, pp. 290–297, 2019.

[114] X. Jiang, H. Qi, X. Qiang, B. Zhao, and H. Dong, "A convolutional neural network-based corrosion damage determination method for localized random pitting steel columns," *Applied Sciences*, vol. 13, no. 15, 2023.

[115] S. Ashrafi, S. Teymouri, S. Etaati, J. Khoramdel, Y. Borhani, and E. Najafi, "Steel surface defect detection and segmentation using deep neural networks," *Results in Engineering*, vol. 25, p. 103972, 2025.

[116] A. A. M. S. Ibrahim and J.-R. Tapamo, "A survey of vision-based methods for surface defects' detection and classification in steel products," *Informatics*, vol. 11, no. 2, 2024.

[117] S. Rath, S. Thakur, S. Mohapatra, and D. Karmakar in *6th International Conference on Automation and Information Technology in Steel and Mining industries (AITISM'19)*, pp. 17–19, 10 2019.

[118] J. C. Foster and M. Price, "Chapter 1 - security coding," in *Sockets, Shellcode, Porting, Coding* (J. C. Foster and M. Price, eds.), pp. 1–64, Burlington: Syngress, 2005.

[119] Edge HeidiSQL, "Heidisql." v12.10, 2025-01-21, `https://www.heidisql.com/`.

[120] iba AG, "Iba analyzer." v8.23, 2024-10-25, `https://www.iba-ag.com/en/ibaanalyzer`.

[121] J. D. Cryer and K.-s. Chan, *Time series analysis : with applications in R.* New York: Springer New York, 2nd ed ed., 2008.

[122] A. Nielsen, *Practical Time Series Analysis: Prediction with Statistics and Machine Learning.* Titolo collana, O'Reilly Media, Incorporated, 2019.

[123] J. C. B. Gamboa, "Deep learning for time-series analysis," *ArXiv*, vol. abs/1701.01887, 2017.

[124] A. T. Jebb, L. Tay, W. Wang, and Q. Huang, "Time series analysis for psychological research: examining and forecasting change," *Frontiers in Psychology*, vol. 6, p. 727, 2015.

[125] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," *2014 Science and Information Conference*, pp. 372–378, 2014.

[126] E. O. Brigham and R. E. Morrow, "The Fast Fourier Transform," *IEEE Spectr.*, vol. 4, pp. 63–70, dec 1967.

[127] D. Cao and J. Liu, "Research on dynamic time warping multivariate time series similarity matching based on shape feature and inclination angle," *Journal of Cloud Computing*, vol. 5, no. 1, p. 11, 2016.

[128] M. B. Shrestha and G. R. Bhatta, "Selecting appropriate methodological framework for time series data analysis," *The Journal of Finance and Data Science*, vol. 4, no. 2, pp. 71–89, 2018.

[129] C.-L. Yang, C.-Y. Yang, Z.-X. Chen, and N.-W. Lo, "Multivariate Time Series Data Transformation for Convolutional Neural Network," in *2019 IEEE/SICE International Symposium on System Integration (SII)*, pp. 188–192, 2019.

[130] C. Park and D. Lee, "Classification of Respiratory States Using Spectrogram with Convolutional Neural Network," *Applied Sciences*, vol. 12, no. 4, p. 1895, 2022.

[131] G. Todeschini, K. Kheta, and C. Giannetti, "An image-based deep transfer learning approach to classify power quality disturbances," *Electric Power Systems Research*, vol. 213, no. 4, p. 108795, 2022.

[132] E. Balouji and O. Salor, "Classification of power quality events using deep learning on event images," in *3rd International Conference on Pattern Recognition and Image Analysis (IPRIA 2017)*, pp. 216–221, apr 2017.

[133] M. Iman, H. Arabnia, and K. Rasheed, "A review of deep transfer learning and recent advancements," *Technologies*, vol. 11, p. 40, 03 2023.

[134] S. S. Tirumala, “A novel weights of weights approach for efficient transfer learning in artificial neural networks,” *Procedia Computer Science*, vol. 212, pp. 295–303, 2022. 11th International Young Scientist Conference on Computational Science.

[135] B. Maschler and M. Weyrich, “Deep transfer learning for industrial automation: A review and discussion of new techniques for data-driven machine learning,” *IEEE Industrial Electronics Magazine*, vol. 15, no. 2, pp. 65–75, 2021.

[136] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[137] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, 09 2014.

[138] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” pp. 770–778, 06 2016.

[139] S. Afaq and S. Rao, “Significance Of Epochs On Training A Neural Network,” *International Journal of Scientific & Technology Research*, vol. 9, no. 6, pp. 485–488, 2020.

[140] O. G. Ajayi and J. Ashi, “Effect of varying training epochs of a faster region-based convolutional neural network on the accuracy of an automatic weed classification scheme,” *Smart Agricultural Technology*, vol. 3, p. 100128, 2023.

[141] I. Kandel and M. Castelli, “The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset,” *ICT Express*, vol. 6, no. 4, pp. 312–315, 2020.

[142] Y. Li, C. Wei, and T. Ma, “Towards Explaining the Regularization Effect of Initial Large Learning Rate in Training Neural Networks,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 1 – 12, 2019.

[143] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, feb 2012.

[144] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on bayesian optimizationb," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.

[145] M. Feurer, T. Springenberg, and F. Hutter, "Initializing bayesian hyperparameter optimization via meta-learning," *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*, vol. 29, pp. 1128–1135, 02 2015.

[146] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, (Red Hook, NY, USA), p. 2951–2959, Curran Associates Inc., 2012.

[147] A. Krogh and J. Vedelsby, "Neural Network Ensembles, Cross Validation and Active Learning," in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, (Cambridge, MA, USA), pp. 231–238, MIT Press, 1994.

[148] Q. Nguyen, H.-B. Ly, H. Lanh, N. Al-Ansari, H. Le, T. Van Quan, I. Prakash, and B. Pham, "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Mathematical Problems in Engineering*, vol. 2021, 02 2021.

[149] E. Kee, J. J. Chong, Z. J. Choong, and M. Lau, "A comparative analysis of cross-validation techniques for a smart and lean pick-and-place solution with deep learning," *Electronics*, vol. 12, no. 11, 2023.

[150] D. Berrar, "Cross-Validation," *Encyclopedia of Bioinformatics and Computational Biology*, vol. 1, pp. 542–545, jan 2018.

[151] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pp. 78–83, 2016.

[152] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *Advances in Information Retrieval* (D. E. Losada and J. M. Fernández-Luna, eds.), (Berlin, Heidelberg), pp. 345–359, Springer Berlin Heidelberg, 2005.

[153] MathWorks, “Matlab.” v2024b, 2024-09-12, `https://ch.mathworks.com/products/matlab.html`.

[154] BitRock, “Xampp.” v8.2.12, 2023-11-21, `https://www.apachefriends.org/`.

[155] X. Li, F. Luan, and Y. Wu, “A Comparative Assessment of Six Machine Learning Models for Prediction of Bending Force in Hot Strip Rolling Process,” *Metals*, vol. 10, no. 5, p. 685, 2020.

[156] R. Jiao, K. Peng, and J. Dong, “Remaining Useful Life Prediction for a Roller in a Hot Strip Mill Based on Deep Recurrent Neural Networks,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, pp. 1345–1354, jul 2021.

[157] D.-C. Wang, Y. Xu, B. Duan, Y. Wang, M. Song, H. Yu, and H. Liu, “Intelligent Recognition Model of Hot Rolling Strip Edge Defects Based on Deep Learning,” *Metals*, vol. 11, no. 2, p. 223, 2021.

[158] L. Tan, L. Wang, X. Zhang, and F. Wang, “Study of Short Stroke Control Model on Hot Rolling Mill,” in *2018 3rd International Conference on Electrical, Automation and Mechanical Engineering (EAME 2018)*, pp. 108–110, 2018.

[159] V. R. Khramshin, S. A. Evdokimov, A. I. Yu, A. G. Shubin, and A. S. Karandaev, “Algorithm of no-pull control in the continuous mill train,” in *2015 International Siberian Conference on Control and Communications (SIBCON)*, pp. 1–5, 2015.

[160] A. A. Radionov, V. R. Gasiyarov, A. S. Karandaev, D. Y. Usatiy, and V. R. Khramshin, “Dynamic Load Limitation in Electromechanical Systems of the Rolling Mill Stand during Biting,” in *2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, pp. 149–154, 2020.

[161] C. Kosemen and D. Birant, “Multi-label classification of line chart images using convolutional neural networks,” *SN Applied Sciences*, vol. 2, pp. 1–20, 2020.

[162] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big Data*, vol. 3, pp. 1–40, 2016.

[163] L. Shao, F. Zhu, and X. Li, “Transfer Learning for Visual Categorization: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[164] V. Maeda-Gutiérrez, C. E. Galván-Tejada, L. A. Zanella-Calzada, J. M. Celaya-Padilla, J. I. Galván-Tejada, H. Gamboa-Rosales, H. Luna-García, R. Magallanes-Quintanar, C. A. Guerrero Méndez, and C. A. Olvera-Olvera, "Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases," *Applied Sciences*, vol. 10, no. 4, p. 1245, 2020.

[165] S. Yadav and S. Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pp. 78–83, 2016.

[166] C. Ji, M. Du, Y. Wei, Y. Hu, S. Liu, L. Pan, and X. Zheng, "Time series classification with random temporal features," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 9, p. 101783, 2023.

[167] G. Tsafnat, P. Glasziou, M. K. Choong, A. Dunn, F. Galgani, and E. Coiera, "Systematic review automation technologies," *Systematic reviews*, vol. 3, p. 74, 2014.