SN Computer Science (2023) 4:628
https://doi.org/10.1007/s42979-023-02104-5

ORIGINAL RESEARCH

A Tool to Combine Expert Knowledge and Machine Learning for Defect Detection and Root Cause Analysis in a Hot Strip Mill

Samuel Latham[1] · **Cinzia Giannetti**[1]

Received: 16 April 2023 / Accepted: 28 June 2023

Abstract
Width-related defects are a common occurrence in the Hot Strip Mill process which can lead to extra processing, concessions, or scrapping. The detection and Root Cause Analysis of these defects is a largely manual process and is vulnerable to several negative factors including human error, late feedback, and knock-on effects in successive steel strip products. Automated tools which utilize Artificial Intelligence and Machine Learning for defect detection and Root Cause Analysis in hot rolling have not yet been adopted outside of surface defect detection and roller force optimization. In this paper, we propose an automated tool for the detection and Root Cause Analysis of width-related defects in the hot rolling process which utilizes a combination of expert knowledge and several Machine Learning models. Through this, we aim to increase the scope, and encourage further development, of Machine Learning applications within the Hot Strip Mill process. Both classical algorithms and Computer Vision methods were used for the Machine Learning component of the tool, namely, classification trees and pre-trained convolutional neural networks. The tool is trained and validated using data from an existing hot rolling mill and thus the challenges of collecting and processing real-world legacy data are highlighted and discussed. The Machine Learning models used are shown to perform optimally by validation performance metrics. The tool is found to be suitable for the specified purpose and would be further improved with more training data.

Keywords Steel Industry · Hot Strip Mill · Automation · Data Analytics · Machine Learning · Classification · Defect Detection · Root Cause Analysis · Knowledge Integration · Legacy Data

Introduction

Steel manufacturers across the globe are producing millions of tonnes of steel strip each year using the Hot Strip Mill (HSM) process and they are constantly working towards improving product quality and reducing waste. A large number and variety of defects, however, continue to be present each year and new techniques to detect and resolve them quickly and efficiently are being continuously explored. Within the HSM process, there are also a number of subprocesses from which these defects and their root causes may occur and originate from. Currently, procedures for the detection of these defects and their root causes typically require some level of manual interaction, such that operators must flag defects, while analysts review them and confirm their root causes.

Within the HSM, the current manual process has several drawbacks. First, it is prone to human error which can result in defects and root causes being wrongly labelled or even left unrecorded. In manual processes, as the size and complexity of a task increases, it becomes more difficult for humans to manually process data [1]. Second, humans may not be readily available to perform an immediate analysis, leading to defects and their root causes being reviewed at a much later time from when they are originally flagged. Automation has a major advantage over manual processes due to its ability to run continuously and process new, unexpected inputs almost instantaneously. In this regard, the key difference between automation and manual processes is that the former is proactive, while the latter is reactive [2]. Finally, this combination of human error and reactivity can lead to successive strips of a defective product being affected due to inefficiency or the inability to implement optimal countermeasures on the

✉ Samuel Latham
865767@swansea.ac.uk

Cinzia Giannetti
c.giannetti@swansea.ac.uk

[1] Faculty of Science and Engineering, Swansea University, Fabian Way, Swansea SA1 8EN, Wales

Published online: 22 August 2023

production line before the next strip enters the HSM or the subprocess in which the defect occurred. Despite this, automation is not without its disadvantages. It should be noted that automated processes and analytics are usually evaluated manually before use to determine whether they are fit for purpose, relying on some necessary human interaction. This can alternatively be seen as beneficial as using both manual and automated methods provide differentiating and, therefore, broader results for the same task [3].

Techniques for defect detection and Root Cause Analysis (RCA), such as the 5 Whys [4], have been in use for many decades. Standardized frameworks utilizing these types of technique also gained popularity in the 1980 s after the development of Lean 6 Sigma [5]. The ability to determine the origins, causes, and effects of defects using such methods continues to assist in providing a better understanding of manufacturing processes and their underlying problems. Over the years, these methods have evolved to include the use of expert knowledge and Artificial Intelligence (AI) [6]. Methods that utilize AI possess the ability to automate informed data-driven decision-making processes. In early development, such technologies were typically referred to as expert systems and they have been used to assist and guide diagnoses in RCA systems and prescribe solutions for given root causes [7].

As the rate of data collection in manufacturing processes increases continuously, it becomes increasingly difficult to make use of this data in an efficient manner [8]. Such vast amounts of data, particularly older and noisy legacy data, are often unorganized and require a great deal of processing to transform them into a suitable standard for analysis [9]. Nowadays, it is common to include Machine Learning (ML) algorithms in automated analytics. By acquiring and processing data, such that the resulting features show differing characteristics, these algorithms can distinguish between the labels they represent [10–13]. As previously mentioned, this can be used to emulate basic tasks using both expert knowledge and ambiguous data. Currently, defect detection and RCA applications in steel manufacturing typically focus on individual subprocesses. While a number of these applications may utilize AI and ML, they are not yet mainstream outside of surface defect detection, roller model optimisation, and life estimation of parts. Examples of these applications include the estimation of roller force using Artificial Neural Networks (ANNs) and other traditional ML algorithms [14], predicting the remaining useful life of rollers using ANNs [15], and the recognition of edge defects using Convolutional Neural Networks (CNNs) [16]. In previous studies, ML models have also been developed to identify the root causes behind Necking in the Roughing Mill and Width Pull in the Finishing Mill [17, 18]. However, these studies only focus on particular subprocesses. While previous work clearly demonstrated the potential for automated defect detection using ML in the HSM, existing applications are not yet used at scale and integrated to achieve an end-to-end system capable of detecting defects, determining their root causes, and providing feedback to operators in an automated way.

In this paper, we propose an automated tool for the detection and RCA of width-related defects in a HSM that combines expert knowledge and multiple ML models, including those created in previous studies. Through this, we intend to increase the scope of automated ML applications in the HSM and demonstrate further development and use of such technologies in the steel industry. In Sect. 2 of this article, we review the HSM process, possible width-related defects that occur in this process, and their root causes. In Sect. 3, we provide an overview of ML applications in manufacturing and the steel industry, and discuss the ML technologies used to build the proposed tool. In Sect. 4, we describe the approach taken to produce the proposed tool as well the integrated ML models. We also describe the decision-making process of the proposed tool and provide further detail of the ML technologies utilised during the process. In Sect. 5, we discuss the results and performance of the newly created classification models and display the implementation of the proposed tool. In Sect. 6, we draw final conclusions on the experiments and proposed tool in this paper.

Background

Overview of the Hot Strip Mill Process

In the HSM process, a steel bar is run through several subprocesses in which the primary aim is to roll it into a strip of a given width and thickness before it is wound into a coil and placed into storage. The bar is reheated in one of two furnaces to 1250°c. The Horizontal Scale Breaker then removes as much oxidation from the bar as possible, so that debris from iron oxide is not rolled into the bar in subsequent subprocesses, thus limiting the possibility of surface defects. This is followed by the Roughing Mill subprocess in which the sides of the bar are rolled using vertical rollers to achieve the required width.

Following the Roughing Mill subprocess, the bar is referred to as a strip as it now has a longer, thinner profile. The strip is then wound into a coil in the Coilbox which attempts to evenly distribute temperature throughout the strip before passing it on to the next subprocess. The strip is unwound from the Coilbox and passed through the Crop Shear in which the head and tail ends of the strip are cut, providing them with a straight edge which improves it's stability throughout the following subprocesses, reducing the risk of shape defects, such as cobbles and warping. The Finishing Scale Breaker removes oxidation from the strip

again before it is passed to the Finishing Mill. In the Finishing Mill, the strip is rolled to the required thickness using a series of seven horizontal rollers. The strip is then cooled to 600°c using a runout table of water with a laminar flow before being wound into a coil in the Coiler subprocess, after which it is stored before being passed onto another process (Fig. 1).

Width-Related Defects in the Hot Strip Mill

This paper focuses on width-related defects with the intention of providing a foundation upon which other types of defects can be included using tools built using similar methods. There are a number of subprocesses from which the root causes of these defects can originate, namely, scheduling, roughing, finishing, and the coiler (Figs. 2, 3).

Scheduling describes data known prior to a steel strip entering the Roughing subprocess. There are several failure modes of width defects that can be determined immediately using query data and conditional checks. Bar specification is the first check that is carried out, following any quality-related failure. Each bar has Spread and Squeeze limits which, in the case of width-related defects, describe whether the bar has the capability to reach the customer's specified dimensions. Single furnace operation can also cause bars to have low temperature during the HSM process. In single furnace operation, bars are heated on a tighter schedule and thus may not be present in the furnace for long enough to reach the required temperature. Delays occur when a bar is held in storage for an extended time, sometimes days, allowing dirt and scale to build up and affect the absorption and distribution of heat around the bar in the furnace.

In the Roughing Mill subprocess, Necking occurs when a strip suddenly loses width at its head or tail end due to edger rollers engaging the strip too early [17, 19]. Anti-necking control (ANC) aims to combat this but, if activated too early, can cause further necking. ANC is used in the first three passes of the Roughing Mill subprocess. Flare can occur if ANC is activated too late, although this is very uncommon. A flared strip is one that suddenly gains width at its head or tail end due to edger rollers engaging the strip tool late

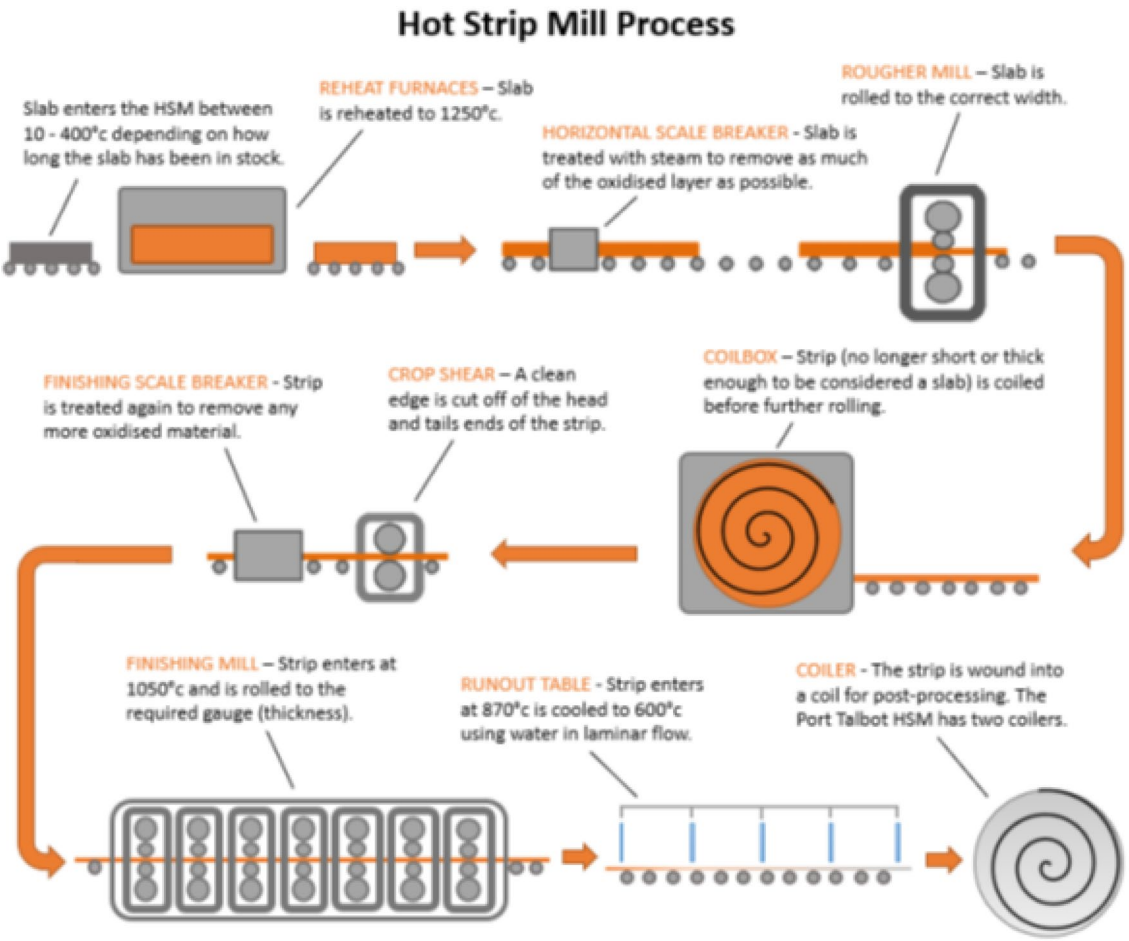

Fig. 1 Diagram of the HSM process

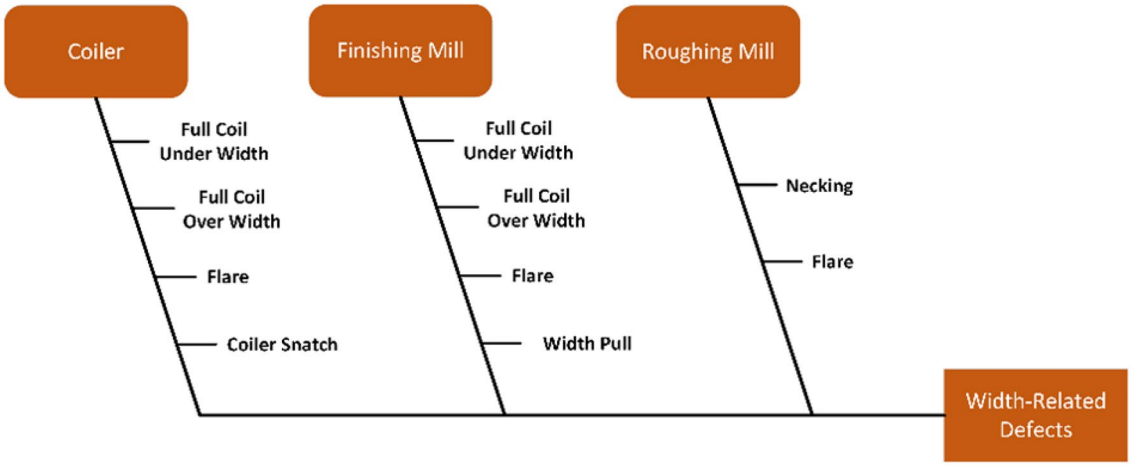

Fig. 2 Fishbone diagram showing possible width-related defects in the HSM

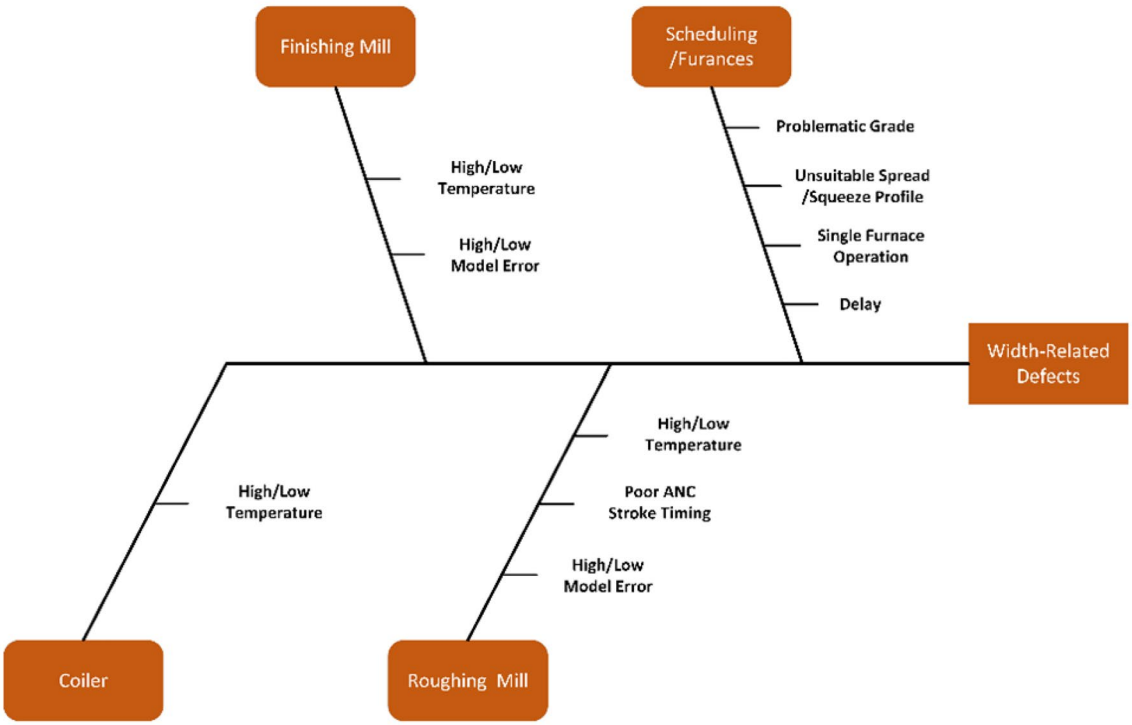

Fig. 3 Fishbone diagram showing the possible causes of width-related defects in the HSM

[17, 19]. Anti-flare control (AFC) is used to combat this in the fourth to sixth passes of the Roughing Mill subprocess; however, it should be noted that flare due to a poorly timed AFC stroke is an extremely rare occurrence. The Roughing subprocess also has a width model which applies offsets to the edger roll positions. This model, which calculates a prediction of width based on other bar properties, is compared to the measured width value at the end of the subprocess to determine its error. If the model error is greater than 5 mm or less than -5 mm, then the offsets created by the Roughing width model is found to have contributed to width defects.

In the Finishing Mill subprocess, Width Pull occurs when the head or tail end of the strip suddenly loses width, similar to necking [20, 21]. The main root causes of Width Pull are high or low temperatures. Although tolerances exist for temperature, readings can sometimes be ambiguous, meaning temperature can still be a contributing factor, although tolerance alarms may not be triggered. This is normally

discovered once an analyst has examined signal data, usually long after a defect has occurred. Within the Finishing Mill, high and low tolerances also exist for width. This means that conditional checks may be carried out prior to checks for Necking to determine whether the strip is fundamentally under or over width to begin with. The Finishing Mill subprocesses also has a width model which can apply offsets to roller positions. In the same manner as the Roughing Mill width model, an error is calculated for the Finishing Mill width model. If this error is greater than 5 mm or less than -5 mm, then the offsets created by the Finishing Mill width model is found to have contributed to width defects.

At the beginning of the coiler subprocess, sudden tension can cause the head end of the strip to elongate. This is known as coiler snatch, and, similar to width pull in the finishing mill subprocess, is usually caused by either high or low temperatures. The same width tolerances from the Finishing Mill also exist in the Coiler subprocesses, so that conditional checks can be carried out prior to Coiler Snatch to determine whether the strip is fundamentally under or over width to begin with.

Related Work

An Overview of Defect Detection and Root Cause Analysis in Manufacturing and the Steel Industry

The amount of data collected in manufacturing processes across the globe is continuously increasing and the challenge of efficiently collecting data and analysing it is still a prominent challenge [8]. Although there are some arguments that there is still much work to be done before defect detection and RCA tools which utilise AI achieve a suitable standard for use in manufacturing process [22], many existing examples show have already shown that it is possible to use these tools to process and analyse large quantities of data quickly and efficiently [13, 23, 24]. In particular, ML has become a very popular technology for use in these tools in manufacturing processes. The aim of ML is to identify patterns in new, unseen data by learning from features produced from historical data [18]. The aim of a defect detection or RCA tool which utilises ML is to provide quick or immediate feedback of defects or root causes [25].

There are many examples of applications from the last several decades which have been developed for the purpose of defect detection and RCA in a variety of manufacturing processes using a number of different ML technologies [26]. One example uses a support vector machine (SVM) classifier to distinguish between images of defective metal sheets [27]. Another uses K-Means Clustering to distinguish between defects of failed semiconductor products and their root causes [28]. One more example uses a long–short-term memory (LSTM) neural network to detect the presence of defects in images of fabric [29].

Defect detection and RCA tools which utilise ML are not yet mainstream outside of roller model optimisation [14] and, more prominently, surface defect detection [30–32]. Such applications are also used on a limited, niche scale, meaning that they are performed as individual operations which have limited connection to tools or data in other subprocesses within the steel rolling process. Although introducing new technologies into such a large and established process can be challenging, there is still much untapped potential for defect detection and RCA tools to provide process-wide automated analyses. Such development could also lead to other benefits within the process such time and resources saved by automating manual processes and instead allocating these towards more complex tasks which require human interaction.

Machine Learning Methods

Convolutional Neural Networks

Over recent decades, CNNs have become one of the most popular methods of image classification [33]. While statistical models can be used as an alternative, they generally struggle to retain contextual information when deriving patterns from image data. On the other hand, CNNs have the ability to emulate the human eye by recognizing patterns in image data while retaining context [34]. A disadvantage of CNNs is that a vast amount of data are required for training. This can be difficult to overcome in real-world applications when addressing problems, such as limited data sets and class imbalances. Alternatively, larger data sets can take longer to train and require more computational power [35]. Although they can be used in a range of applications in the steel industry, CNNs have been widely used in surface defect detection [30].

CNNs are directed graphs consisting of an input layer, an output layer, and a number of hidden layers in between [36]. Image data enters the network through the input layer before it is processed through the hidden layers to extract features which are used to determine a classification label at the output layer. The hidden layers of a CNN typically consist of convolutional layers, which are followed by an activation function, pooling layers, and a fully connected layer. Convolutional layers use filters for purposes, such as edge detection, such that image features are made more distinguishable. The activation function which follows this layer overwrites negative values in the convolution to 0, so that redundant data are removed. The pooling layer makes individual image features distinguishable by reducing data dimensionality into segments using a smaller kernel than the prior convolution. The fully connected layer maps the

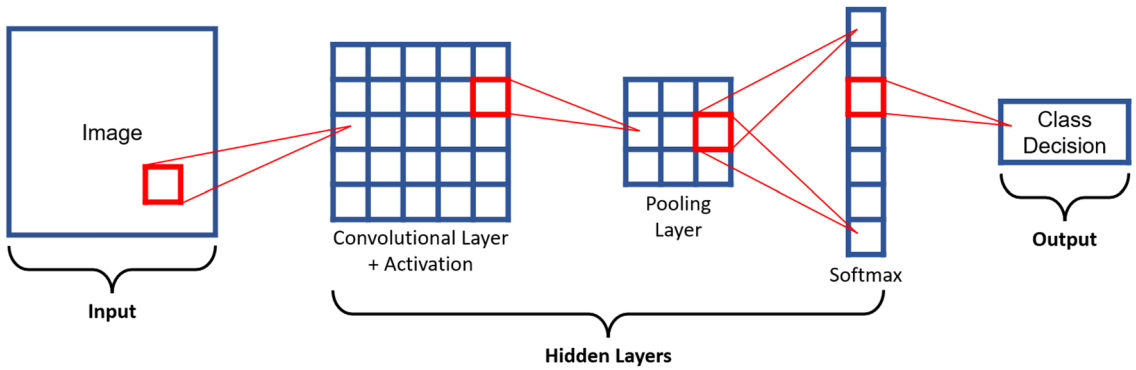

Fig. 4 Diagram of a basic CNN architecture

values from the previous layer into a one-dimensional array of probability values, each representing the probability of the input image belonging to each class within the given classification task. The final classification label is chosen based on these probabilities (Fig. 4). There are many different existing CNN architectures of various constructed from varying combinations of these layers for different classification tasks. No one CNN architecture is suited to every classification task [37].

As previously discussed, training CNNs from scratch can require vast amounts of time and computational resources and can also be challenging when available data sets are limited, unbalanced, and noisy as is often the case with industrial legacy data. To combat this, previously trained networks can be partially re-trained using a small data set specific to the classification task at hand [38]. This is referred to as Transfer Learning and it can assist in boosting classification performance while dramatically reducing training times. Transfer Learning is typically performed by replacing the last few layers, including the output layer, of the pre-trained network with new layers whose feature extraction is tailored to the new data set when re-training. It aims to utilise the features learned from the data used to train the initial model to help in making decisions on data in a new classification scenario [37].

The pre-trained CNN architecture we have chosen for the transfer learned image classification models used in the tool is GoogLeNet. While this architecture has a large, complex structure, it is not computationally expensive to re-train [39]. The GoogLeNet architecture also includes an 'Inception Module' which is a series of convolutions which enable the network to examine a wide range of features in an image without dramatically increasing computational

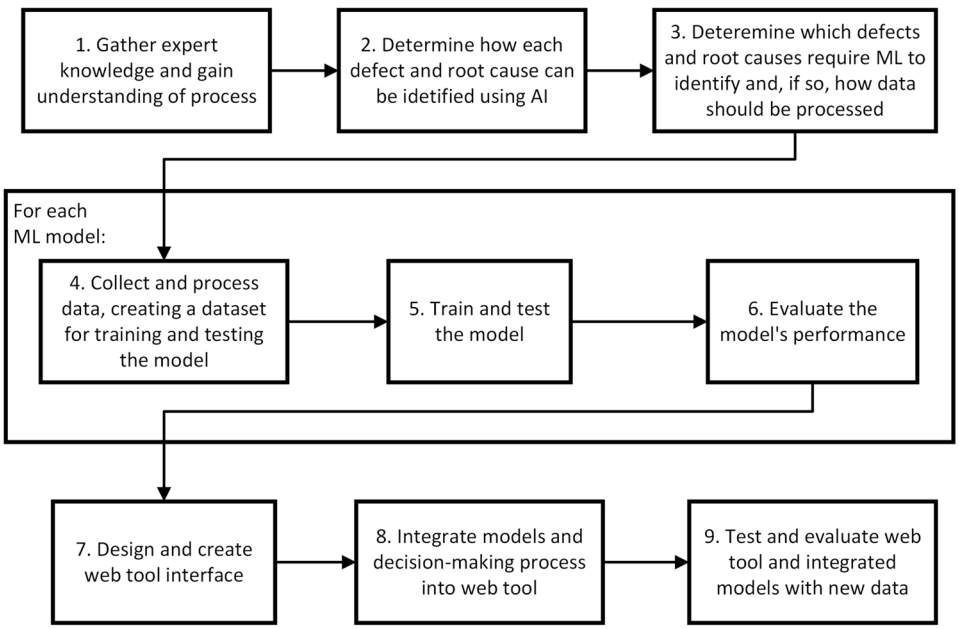

Fig. 5 Diagram describing the workflow followed to create the proposed web tool

Fig. 6 Flowchart showing the decision-making process of the proposed tool

Table 1 Roughing Mill classification task samples

Label	Total (Label)
Necking	158
Flare	176
Okay	158
Total (Dataset)	492

Table 2 Finishing Mill classification task samples

Label	Total (Label)
Width Pull	223
Okay	225
Total (Dataset)	448

expense [39]. GoogLeNet is trained on a data set of over 14 million images, known as the ImageNet data set.

Classical Machine Learning

Classical ML requires feature extraction to be completed before learning, as opposed to Deep Learning in which these are part of the same process. Features used in classical ML algorithms; however, are typically basic numerical features which can be quicker and simpler to compute, and thus may not require as much training data as a Deep Learning algorithm to produce a reliable model [40]. In a previous study to determine temperature-related failure modes of Width Pull in the Finishing Mill subprocess, a variety of classical ML algorithms were used [18]. These included Trees, Naïve Bayes, K-nearest neighbor, support vector machines, ANNs, and ensembles. A Coarse Tree model was found to be the best performing model and is, therefore, used as the selected model for temperature-related failure modes of poor width performance in the Finishing Mill for this tool. The same model can be used to classify temperature-related failure modes of Coiler Snatch, since the same data and feature sets are used but, as shown above a limited data set is available for the Coiler Snatch experiment.

Classification of Time Series as Images

Time series data are commonly used in ML and classification tasks, and there are a wide range of techniques available for processing this type of data prior to learning. These techniques are chosen based on the characteristics of the time series data and how these characteristics can be best extracted or projected from the raw data. It is also important to consider whether a time series is univariate or multivariate, and whether it is domain specific. Time series data can be described as trending, cyclical, seasonal, or random. A trending time series is one which shows a relatively linear increase or decrease in its values over time. Cyclical time series exhibit some level of repeatability in frequencies over a non-fixed period of time, while seasonal time series see repeatability in values and frequencies over a fixed period of time, although both types can still be affected by outside factors. Random time series exhibit little repeatable behaviour in their values or frequencies [41].

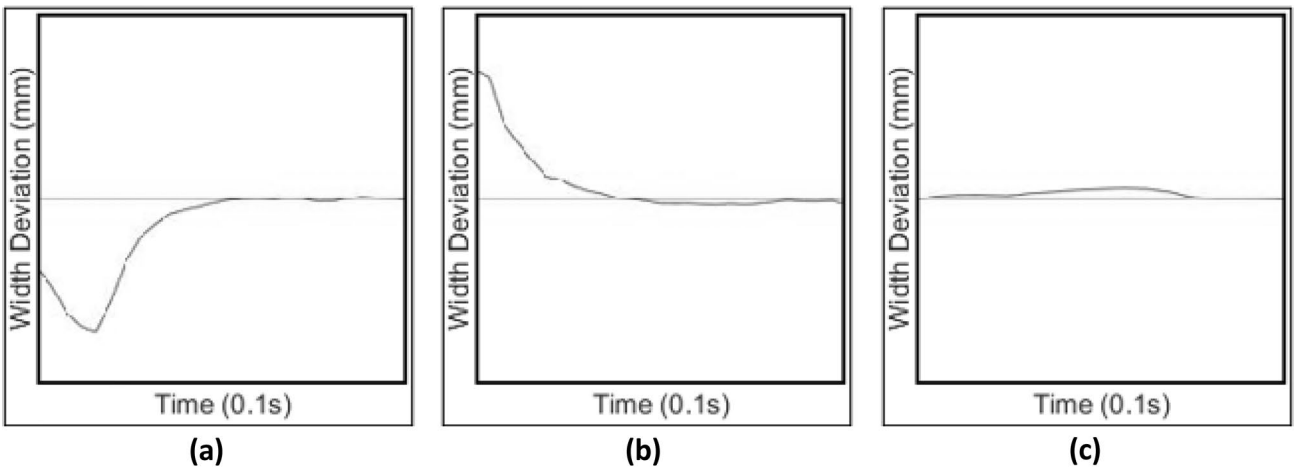

Fig. 7 Examples of images contained in the data set used for the Roughing Mill classification task. Images show the width deviation of an individual sample and the constant y-axis value of 0. Borders and axis labels are only included for illustration purposes within these figures. **a** A sample showing no signs of defective behaviour and, therefore, considered to be Okay. **b** A sample showing characteristics of Flare. **c** A sample showing characteristics of Necking

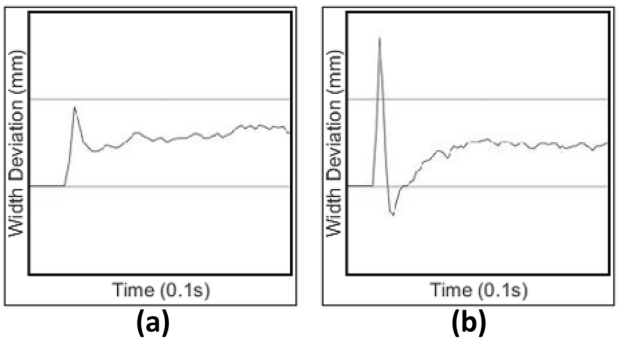

Fig. 8 Examples of images contained in the data set used for the Finishing Mill classification task. Images show the width deviation of an individual sample and the two y-axis values representing high and low width tolerances. Borders and axis labels are only included for illustration purposes within these figures. **a** A sample showing no signs of defective behaviour and, therefore, considered to be Okay. **b** A sample showing characteristics of Width Pull at the head end of the bar

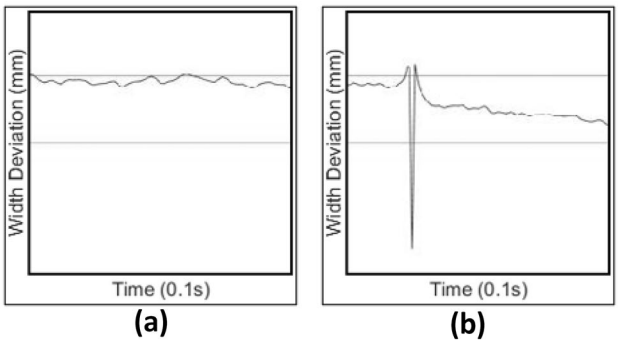

Fig. 9 Examples of images contained in the data set used for the Coiler classification task. Images show the width deviation of an individual sample and the two y-axis values representing high and low width tolerances. Borders and axis labels are only included for illustration purposes within these figures. **a** A sample showing no signs of defective behaviour and, therefore, considered to be Okay. **b** A sample showing characteristics of Coiler Snatch at the head end of the bar

Table 3 Coiler classification task samples

Label	Total (Label)
Coiler snatch	22
Okay	22
Total (Dataset)	44

In ML, when data are processed, such that new information is presented, the process is called feature extraction. When data are processed, such that characteristics of the original or raw data are projected, the process is called feature selection. The most basic feature extraction that can be performed on time series data, and numeric data in general, is the calculation of basic numerical features, such as minimum, maximum, and standard deviation. However, unless time series data contains predictable values on a known scale, this method may not produce an accurate representation of the time series and loses valuable information in the process. It is possible to use transformation functions to compute alternative representations of the raw time series data, such as fast Fourier transform, which represents time series or signal data in the frequency domain [42], and Dynamic Time Warping, which compute the distance of two time series of different lengths [43]. While these may be used to extract intended features, valuable information from the original data may be lost in the process.

Another method is to represent time series data in image form, converting it into an image classification task. One popular approach to this is to encode time series data as an image using a transformation function and mapping values to an image or an RGB matrix. Commonly used examples of this are Gramian Angular Fields, which encodes temporal correlations between multivariate time series [44], and spectrograms, which encode frequency transformations, such as fast Fourier transform [45]. As previously mentioned, the transformation functions used in the encoding process may sacrifice valuable information from the original data. Another approach to representing time series data in image form is to encode the original time series itself without a transformation function. While the use of this approach may be limited to domain-specific applications [17, 46, 47], it can be used when characteristics are repeatable between images, meaning that the type of time series is not a majorly impacting factor. With this approach, computational cost

Table 4 Hyperparameters for grid search optimisation experiment

Max epochs	Mini batch size
5	2
10	4
15	8
20	16
25	32

Table 5 K-fold cross validation splits for each data set in each experiment

Fivefold cross validation split	Roughing Mill dataset	Finishing Mill dataset	Coiler dataset
Split 1	99	90	9
Split 2	99	90	9
Split 3	99	90	9
Split 4	99	90	9
Split 5	98	89	8
Total number of images	492	448	44

may also be reduced as transformation functions do not need to be calculated.

The time series used in this article's experiments are univariate, random, and domain specific. We, therefore, represent them as images, using simple pre-processing calculations to reduce noise and project the most important aspects of the raw time series while limiting computational cost.

Methodology

Workflow for Creating the Proposed Tool

The first step to creating the proposed tool was to gather expert knowledge from operators and analysts themselves, and to gain a deep understanding of the HSM process, width-related defects, and their root causes. Following this, we reviewed the tools currently used to determine the presence of defects and their root causes, and, from this, could determine how each defect and root cause could be identified using AI. The next step was to determine which of these issues would require ML to determine their presence due to their ambiguity when analysed manually.

For the ML models created in both previous studies [17, 18] and in this study, we collected and processed the required data, creating a data set from training and testing. After training and testing the model, we then evaluated the its performance and suitability for their classification tasks and within the web tool.

After evaluating all of the required ML models, we designed and implemented the web tool interface and, following this, integrated the decision-making process described in Sect. 4.2 as well as the required ML models, both of which were then tested and evaluated using new data samples separate from those used in the originally training and testing of each ML model.

In previous studies, we carried out steps 4, 5, and 6 for ML models for RCA. Specifically, we created an image classification model to classify ANC stroke timing in the Roughing Mill subprocess [17], and a Tree model to classify temperature-related behaviour in the Finishing Mill and Coiler subprocesses [18]. In this paper, we carry out steps 4, 5, and 6 to create image classification models for the purpose of classifying width-related defects throughout the HSM subprocess. The first model aims to distinguish between Necking, Flare, and normal width behaviour in the Roughing Mill, the second aims to distinguish between Width Pull and normal width behaviour in the Finishing Mill, and the third aims to distinguish between Snatch and normal width behaviour in the Coiler. After evaluating these models, we carry out steps 7, 8, and 9, and conclude by evaluating the ML models and web tool created in this paper, and their suitability for the intended purpose (Fig. 5).

Overview of Data-Driven Decision-Making in the Proposed Tool

For each HSM subprocess, the proposed tool loads the relevant time series data and, once finished, classifies this data

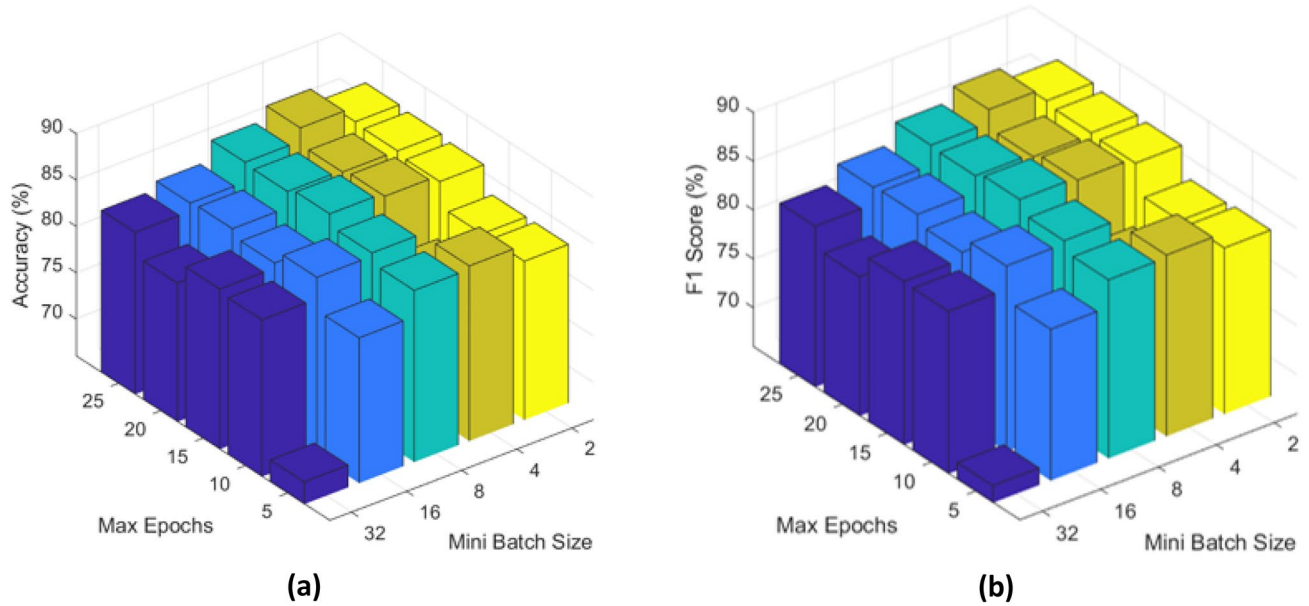

Fig. 10 3D bar charts showing the average fivefold cross validation performance of each hyperparameter value combination for Grid Search Optimisation in the Roughing Mill Necking and Flare classification task. **a** Accuracy. **b** F1 Score

Fig. 11 Classification results of the final model for the Roughing Mill Necking and Flare classification task when trained on all available data and tested using new, unseen testing data

using the models described in this section. The flowchart in Fig. 6 describes the overall decision-making process as the strip, and, therefore, data, progresses through the HSM (Fig. 6).

In previous studies, we have created standalone ML models for the purpose of root cause classification with the intention of reducing workload on analysts. In this paper, we focus on the development of ML models for the purpose of defect detection. Through this, we create the building blocks necessary to create the proposed tool, which aims to fully automate the analysis process from defect occurrence to operator feedback (Fig. 5).

Image Classification Models for Width-Related Defects

Two different approaches have been taken to create the classification models used in the proposed tool. The chosen approaches were based on which data provided the most significant information on each defect. Width deviation, for example, provides the most significant information about a strip's width and shape when analysing shape defects, such as necking, flare, width pull, and coiler snatch. The first approach is image classification using pre-trained CNNs. This has been used for classification tasks in which visual characteristics are the main factor in distinguishing between failure modes. Specifically, it was used to classify image representations of time series data to distinguish between early and well-timed ANC strokes in the Roughing Mill subprocess [17].

The second approach is the use of classical ML algorithms. This has been used for classification tasks in which simple numerical features and simple classification algorithms, such as trees, K-nearest neighbours, and support vector machines, are sufficient for overcoming ambiguity in data and distinguishing between failure modes. Specifically, it

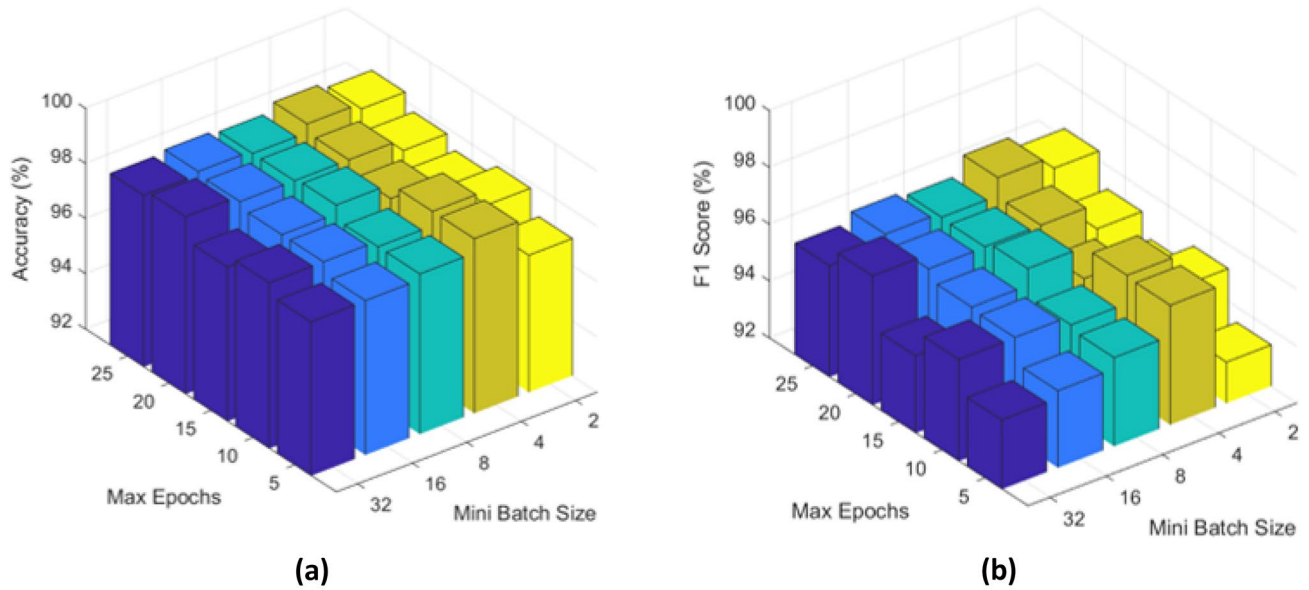

Fig. 12 3D bar charts showing the average fivefold cross validation performance of each hyperparameter value combination for Grid Search Optimisation in the Finishing Mill Width Pull classification task. **a** Accuracy. **b** F1 Score

was used to classify temperature-related failure modes in the Finishing Mill and Coiler subprocesses using simple numerical features derived from time series data including mean, peak value, and standard deviation.

For the remaining models evaluated in this paper, we use a similar approach to that used in the ANC stroke timing classification task [17] by representing time series data as images and performing Transfer Learning using the GoogLeNet architecture [39]. For each of the three models created in this paper, time series data of width deviation is used as it provides an overview of how the shape of each steel strip changes throughout each subprocess. Each subprocess has its own width deviation time series data. The models, data sets, and time series image representations used in the Roughing Mill Necking and Flare, Finishing Mill Width Pull and Coiler Snatch classification tasks are described in Sects. 4.3.1, 4.3.2, and 4.3.3, respectively.

Flare and Necking in the Roughing Mill Subprocess

The purpose of this model is to distinguish between Necking, Flare, and normal, or 'Okay', behaviour in Roughing Mill width performance. A data set consisting of samples labelled as having these defective behaviours was compiled. This data set contains 158 samples with Necking and 176 samples with Flare. 158 samples with no defective behaviour were chosen to represent the Okay label during training and testing (Table 1).

Each sample consists of a single variable-length time series collected directly from the steel mill along with a constant y-axis value of 0, as deviation refers to width measurements relative to 0 mm. This line remains central in the time series image. This time series describes width deviation at the exit of the Roughing Mill subprocess. Characteristics of Necking and Flare are easily recognizable in image form as opposed to numerical or statistical representations due to the ambiguity and fluctuations of values in raw time series data. Representation of Necking, Flare, and even Okay samples is based on the changes in shape and the position of the time series within its image representation; features that a pretrained CNN is purposely built to learn and detect (Fig. 7).

Width Pull in the Finishing Mill Subprocess

The purpose of this model is to distinguish between strips that are either Okay or show characteristics of Width Pull in the Finishing Mill subprocesses. The data set used for training and testing this model contains 223 samples labelled as having Width Pull and 225 Okay samples with no defective behaviour (Table 2).

Similar to the data used in the Roughing Mill experiment, each sample in this data set consists of a single variable-length time series. In this experiment, however, two y-axis values are used to represent upper and lower width tolerances. These values can change between different strip profiles and specifications. Features of Width Pull are recognizable by elongation, and thus a loss of width, at the head end of the strip (Fig. 8).

Snatch in the Coiler Subprocess

The purpose of this model is to distinguish between strips with characteristics of Coiler Snatch and those with Okay behaviour in the Coiler subprocesses. The data set used for training and testing this model contains 22 samples labelled as having Coiler Snatch and 22 Okay samples with no defective behaviour (Table 3).

Samples in this experiment also consist of a single variable-length time series and two non-constant y-axis values to represent positive and negative width tolerances. Similar to Width Pull in the Finishing Mill subprocess, features of Coiler Snatch can be recognized by elongation at the head end of the strip (Fig. 9).

Grid Search Optimisation and K-Fold Cross Validation

Grid search optimisation is a method which helps us to determine the optimal training parameters for an image classification algorithm [48]. In a previous image classification study, we determined that Max Epochs and Mini Batch Size were the only options which had a significant measurable impact on model performance [17]. A higher Max Epochs value usually results in higher accuracy but also carries the risk of long training times and overfitting, whereas a lower Max Epochs value may learn too quickly and underfit [49]. A lower Mini Batch Size value can increase a model's ability to generalize features but carries the risk of both underfitting and overfitting by either learning from the wrong features or generalizing features too quickly [50]. We have, therefore, decided to use the Cartesian Product of Table 4 as the set of test cases for each experiment.

K-fold cross validation has also been used in conjunction with Grid Search Optimisation to determine how well each model can generalize features during training when using a subset of the full training data set [51]. We used a k of 5, such that the training data in each data set is split into five components of 20%. Therefore, 80% of the total training data set is used to complete five training and testing runs, each run discarding one of the training data set splits as described in Table 5. A full fivefold cross-validation experiment is completed for each grid search optimisation hyperparameter combination.

Results and Discussion

Image Classification Experiments

Model for Flare and Necking in the Roughing Mill Subprocess

The fivefolds of the Roughing Mill Flare and Necking data set have been used to train and test each combination of the Grid Search Optimisation training options, as described in Table 4. Figure 10a, b shows the average accuracy and F1 Score of the fivefold cross validation experiments for each for each Grid Search Optimisation hyperparameter combination. Accuracy describes the overall percentage of samples in the test data set which have been classified correctly. While accuracy is a useful metric for performance, F1 Score is derived from recall and precision, and is considered to be more reliable as it provides further insight into model performance with regard to true negative and false positive rates [52]. Precision describes the percentage of samples which truly belong within the class they are assigned, whereas recall describes the percentage of samples belonging to a given class that are correctly assigned.

Figure 10a, b shows that model performance increase with the number of Max Epochs. This is likely the result of longer training time due to a larger number of iterations. A Max Epochs value of 5, the lowest value used in the experiment, shows the worst performance across for all Mini Batch Size values. While there is no linear pattern shown by the results of Mini Batch Size values, the results show that, in this classification task, the Mini Batch Size value should be decreased as Max Epochs increase. There is, however, a plateau in performance when using these lower Mini Batch Size values and, thus, a Mini Batch Size of 4 and Max Epochs value of 25 have been selected for use in the final Roughing Mill Necking and Flare image classification model. The classification results of a randomly selected fold from this experiment using the selected hyperparameter values is shown in Fig. 11.

Model for Width Pull in the Finishing Mill Subprocess

The fivefolds of the Finishing Mill Width Pull data set have been used to train and test each combination of the Grid Search Optimisation training options, as described in Table 4. Figure 12a, b shows the average accuracy and F1 Score of the fivefold cross validation experiments for each for each Grid Search Optimisation hyperparameter combination.

Figure 12 shows little variation between the results of different hyperparameter values. However, the best performing models are still those that use higher Max Epochs values. Shorter training times and less complexity may be a contributing factor towards this good performance as, although this the data set used in this classification task uses a similar number of samples to the Roughing Mill classification task, the model only needs to distinguish between two classes, hence the smaller distribution in performance between high and low Max Epoch value. Performance also increases, albeit to a plateau, as the Mini Batch Size value decreases. This is shown by the performance of those models which use a Mini Batch Size of 2 and show slight decrease in performance from those with a Mini Batch Size of 4. A Mini Batch Size of 4 and Max Epochs value of 25 have, therefore, been selected for use in the final Finishing Mill Width Pull image classification model. The classification results of a randomly selected fold from this experiment using the selected hyperparameter values is shown in Fig. 13.

Model for Snatch in the Coiler Subprocess

The fivefolds of the Coiler Snatch data set have been used to train and test each combination of the Grid Search Optimisation training options, as described in Table 4. Figure 14a, b shows the average accuracy and F1 Score of the fivefold cross validation experiments for each for each Grid Search Optimisation hyperparameter combination.

In Fig. 14, it is shown that a model performance increases with the Max Epochs value. It is possible that this is a result of the relatively small data set size used in this classification task as a lower Max Epochs value results in shorter and possibly insufficient training times. Model performance is also shown to increase as the Mini Batch Size value decreases. It is possible that the relatively small data set used in this classification task, which would provide the algorithm with smaller portions of data in a larger number of iterations. A Max Epochs value of 25 and a Mini Batch Size of 2 have, therefore, been selected as the final hyperparameter values for the final Coiler Snatch image classification model. Figure 15 shows a confusion matrix which shows the classification results of a randomly selected fold from this experiment using the selected hyperparameter values.

A Tool for Defect Detection and Root Cause Analysis Which Combines Machine Learning Models and Expert Knowledge

The proposed tool was created and simulated in a local web environment. While loading times may differ after deployment depending upon available computational power, the simulation shown in this section provides an insight into how the tool visualizes and processes data from the HSM process, and how the decision-making process described earlier

Fig. 13 Classification results of the final model for the Finishing Mill Width Pull classification task when trained on all available data and tested using new, unseen testing data

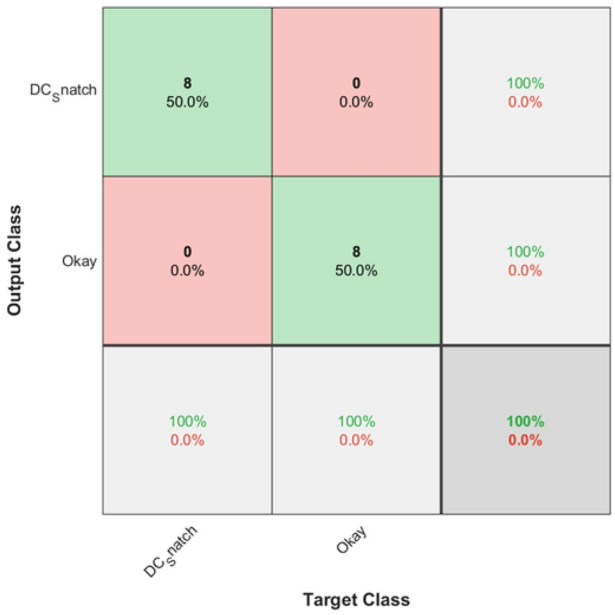

Fig. 15 Classification results of the final model for the Coiler Snatch classification task when trained on all available data and tested using new, unseen testing data

in this article is followed. Figure 16 displays the full tool interface, while subsequent figures show individual classifications made by the models described in the previous sections (Figs. 17, 18, 19 and 20).

The results of the classification models shown in the previous sections and in prior works [17, 18] show that the proposed tool performs optimally for the tasks it has been designed to compute when combined with expert knowledge. The tool, displayed in the figures above, shows that an end-to-end methodology can be adopted, such that applications for defect detection and RCA can be implemented and combined across multiple subprocesses, from loading data

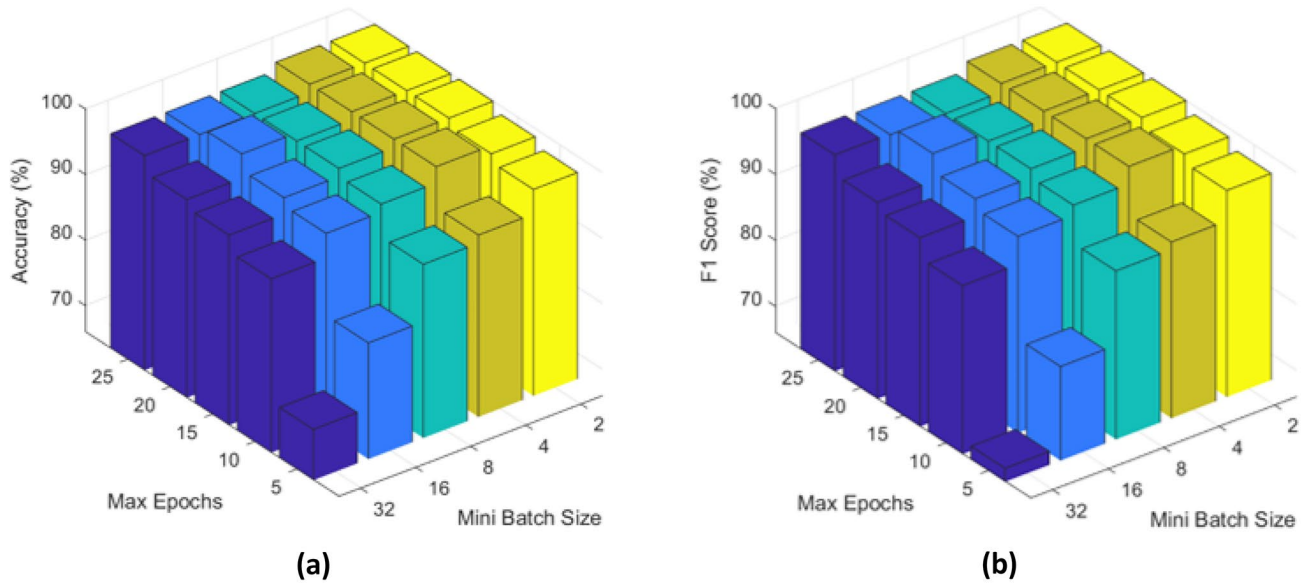

Fig. 14 3D bar charts showing the average fivefold cross validation performance of each hyperparameter value combination for Grid Search Optimisation in the Coiler Snatch classification task. **a** Accuracy. **b** F1 Score

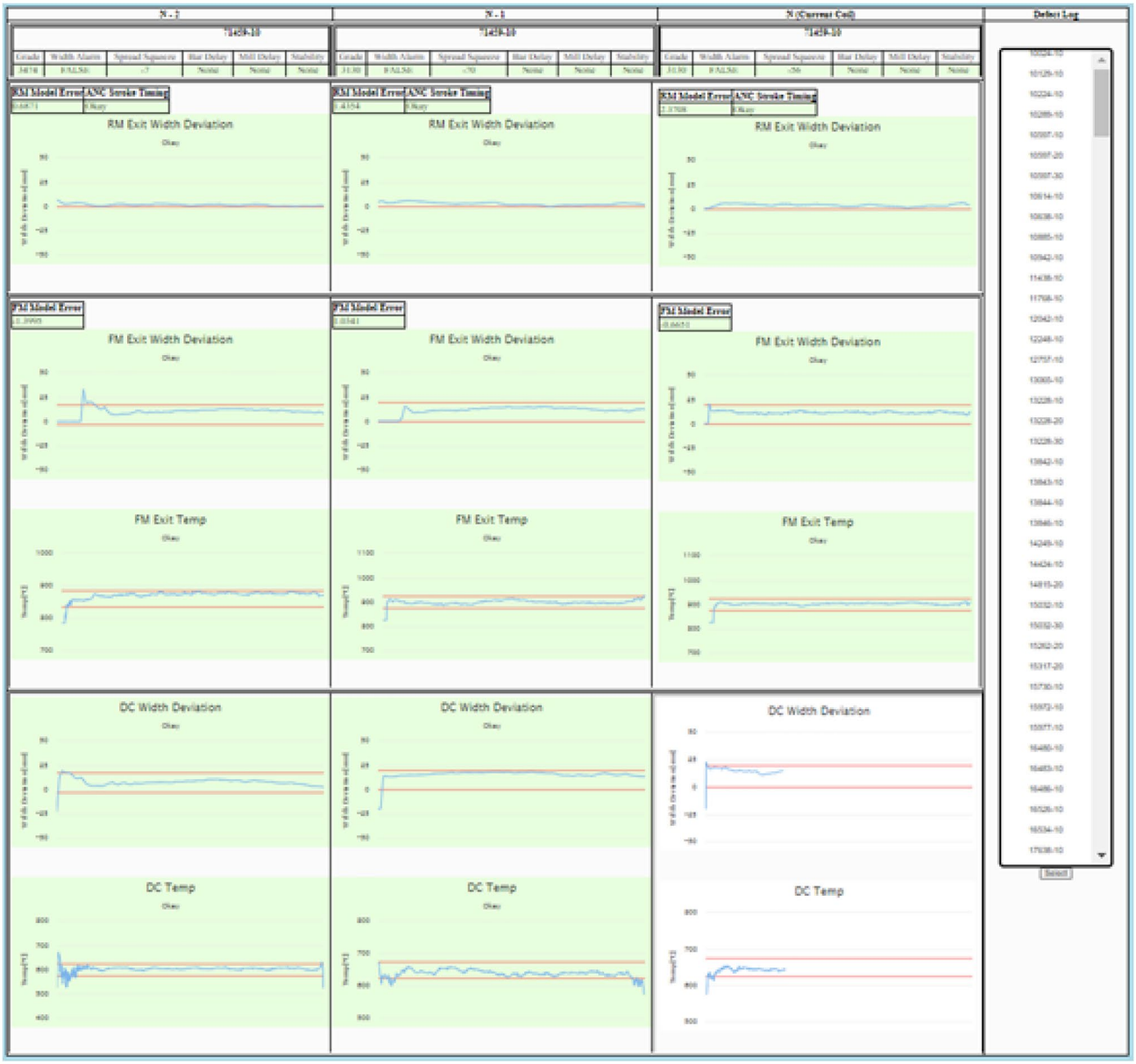

Fig. 16 Interface of the proposed tool displaying visualized data from the HSM process. In this instance, coiler subprocess data are being dynamically loaded, while data for previous coils and subprocesses have finished loading and thus been classified

upon product entry into a process to classification feedback upon product exit.

While the results show that the models and web tool performed sufficiently for the intended purpose when tested using new, unseen data during fivefold cross validation, there is slight room for improvement for the Roughing Mill Necking and Flare model. This, however, is based on its slightly below average performance compared to the other two models.

At the time of writing, all available data was used to create and test these models. Although they perform sufficiently, newly available data would only improve performance and, more importantly, the reliability of these results by increasing the overall data set size.

In a future study, once a sufficient amount of new data from the given HSM process becomes available, we would test these models with the new, unseen samples before retraining them with a larger data set and test them again before comparing these results.

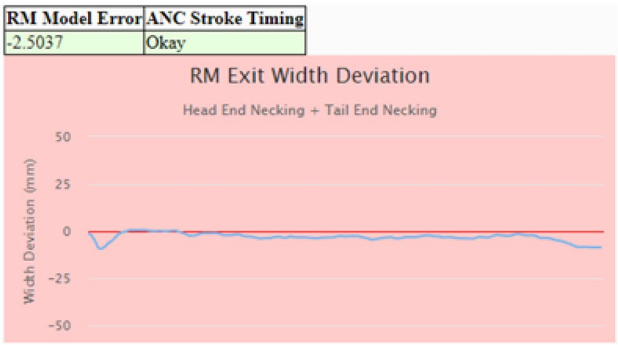

Fig. 17 Example of a strip that has passed through the Roughing Mill subprocess that has been classified as having Necking

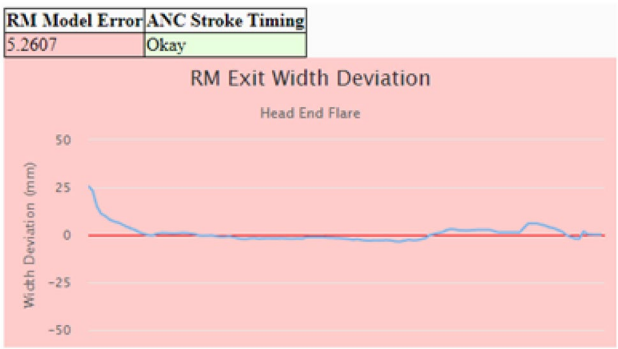

Fig. 18 Example of a strip that has passed through the Roughing Mill subprocess that has been classified as having Flare due to model error

Conclusion

In this paper, three individual classification models for raw time-series images have been created for the purpose of detecting width-related defects in the HSM process. The first model distinguishes between normal width behaviour, Necking, and Flare in the Roughing Mill subprocess. The second model distinguishes between normal width behaviour, and Width Pull in the Finishing Mill subprocess. The final model distinguishes between normal width behaviour, and Snatch in the Coiler subprocess. Each model was trained and tested using a combination of Grid Search Optimisation and fivefold cross validation. The results for the best performing models show F1 Scores of 91.1%, 99.5%, and 100% for the Roughing Mill model, Finishing Mill model, and Coiler model, respectively.

We have created a framework and working tool which integrates these models as well, as previously created classical ML models and image classification models, for defect

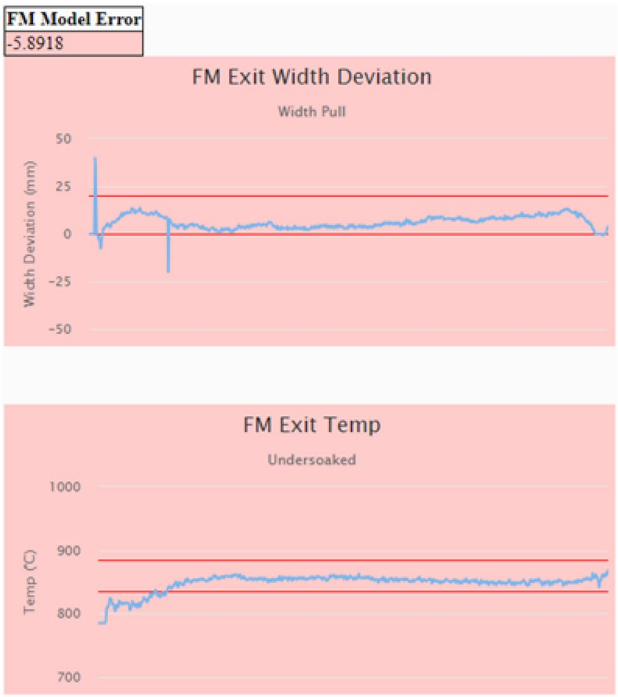

Fig. 19 Example of a strip that has passed through the Finishing Mill subprocess that has been classified as having Width Pull and due to a combination of low temperatures and model error

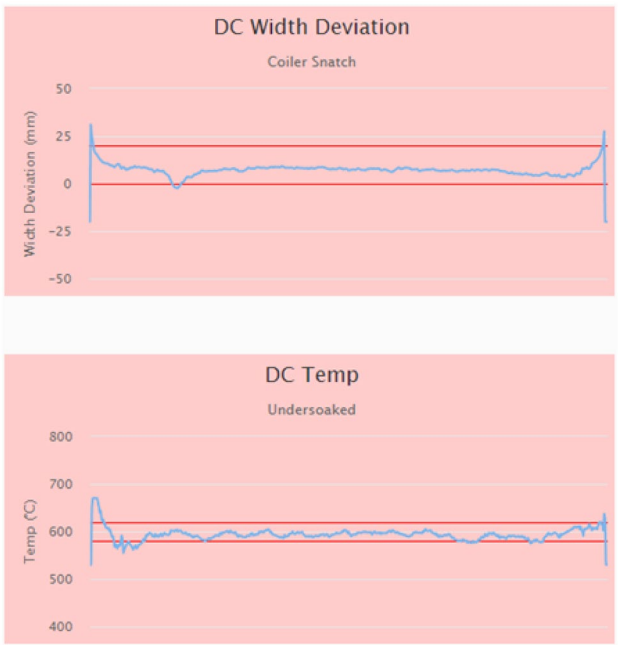

Fig. 20 Example of a strip that has passed through the Coiler subprocess that has been classified as having Snatch due to low temperatures

detection and RCA together with expert knowledge to extend the scale and scope of defect detection and RCA of width-related defects across the entire HSM process. Through this, we have provided a foundation upon which further work or similar methodologies can be used to create applications for wide-scale defect detection and RCA in industrial settings, particularly in the steel industry and within the HSM process.

In future work, we would consider a long-term evaluation to determine such a tool's effectiveness and performance after several years. We would also consider the development and integration of further ML applications and expert knowledge for the detection and RCA of other types of defects in the HSM process.

Acknowledgements The authors would like to acknowledge the Materials and Manufacturing Academy (M2A) funding from the European Social Fund via the Welsh Government (c80816) and Tata Steel Europe that has made this research possible. Prof. Giannetti would like to acknowledge the support of the UK Engineering and Physical Sciences Research Council (EP/V061798/1). All authors would like to acknowledge the support of the IMPACT and AccelerateAI projects, part-funded by the European Regional Development Fund (ERDF) via the Welsh Government.

Funding This project was funded through the Materials and Manufacturing Academy (M2A) funding from the European Social Fund via the Welsh Government (c80816) and Tata Steel Europe.

Availability of Data and Materials The data used to produce the results in this project belong to Tata Steel Europe and are, therefore, unavailable for use by those other than personnel.

Declarations

Conflict of Interest Both authors declare that they have no conflict of interest.

References

1. Tsafnat G, Glasziou P, Choong MK, Dunn A, Galgani F, Coiera E. Systematic review automation technologies. Syst Rev. 2014;3:74. https://doi.org/10.1186/2046-4053-3-74.
2. Abbaszadegan A, Grau D. Assessing the influence of automated data analytics on cost and schedule performance. Procedia Eng. 2015;123:3–6. https://doi.org/10.1016/j.proeng.2015.10.047. (**Selected papers from Creative Construction Conference 2015**).
3. Torres FBG, Gomes DC, Hino AAF, Moro CMC, Cubas MR. Comparison of the results of manual and automated processes of cross-mapping between nursing terms: quantitative study. JMIR Nurs. 2020. https://doi.org/10.2196/1850.
4. Serrat O The five whys technique, 1st edition. pp. 307–310. Springer. 2017. https://doi.org/10.1007/978-981-10-0983-9_32
5. Sreedharan VR, Raju R. A systematic literature review of lean six sigma in different industries. Int J Lean Six Sigma. 2016;7:430–66. https://doi.org/10.1108/IJLSS-12-2015-0050.
6. Arnheiter ED, Greenland JE. Looking for root cause: a comparative analysis. TQM J. 2008;20:18–30. https://doi.org/10.1108/09544780810842875.
7. Diez-Olivan A, Ser JD, Galar D, Sierra B. Data fusion and machine learning for industrial prognosis: trends and perspectives towards industry 4.0. Inf Fusion. 2019;50:92–111. https://doi.org/10.1016/j.inffus.2018.10.005.
8. Yaqoob I, Hashem IAT, Gani A, Mokhtar S, Ahmed E, Anuar NB, Vasilakos AV. Big data: from beginning to future. Int J Inf Manag. 2016;36:1231–47. https://doi.org/10.1016/j.ijinfomgt.2016.07.009.
9. Madden S. From databases to big data. IEEE Internet Comput. 2012;16:4–6. https://doi.org/10.1109/MIC.2012.50.
10. Cinar Z, Nuhu A, Zeeshan Q, Korhan O, Asmael M, Safaei B. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability. 2020;12:8211. https://doi.org/10.3390/su12198211.
11. Dogan A, Birant D. Machine learning and data mining in manufacturing. Expert Syst Appl. 2021;166: 114060. https://doi.org/10.1016/j.eswa.2020.114060.
12. Essien A, Giannetti C. A deep learning framework for univariate time series prediction using convolutional lstm stacked autoencoders, 2019;1–6 . https://doi.org/10.1109/INISTA.2019.8778417
13. Giannetti C, Essien A. Towards scalable and reusable predictive models for cyber twins in manufacturing systems. J Intell Manuf. 2022;33:441–55. https://doi.org/10.1007/s10845-021-01804-0.
14. Li X, Luan F, Wu Y. A comparative assessment of six machine learning models for prediction of bending force in hot strip rolling process. Metals. 2020;10:685. https://doi.org/10.3390/met10050685.
15. Jiao R, Peng K, Dong J. Remaining useful life prediction for a roller in a hot strip mill based on deep recurrent neural networks. IEEE/CAA J Autom Sinica. 2021;8:1345–54. https://doi.org/10.1109/JAS.2021.1004051.
16. Wang D-C, Xu Y, Duan B, Wang Y, Song M, Yu H, Liu H. Intelligent recognition model of hot rolling strip edge defects based on deep learning. Metals. 2021;11:223. https://doi.org/10.3390/met11020223.
17. Latham S, Giannetti C. Pre-trained cnn for classification of time series images of anti-necking control in a hot strip mill, 2021;77–84 https://doi.org/10.12792/iciae2021.015
18. Latham S, Giannetti C. Root cause classification of temperature-related failure modes in a hot strip mill, 2022;36–45. https://doi.org/10.5220/0011380300003329
19. Tan L, Wang L, Zhang X, Wang F. Study of short stroke control model on hot rolling mill, 2018;108–110. https://doi.org/10.2991/eame-18.2018.21
20. Khramshin VR, Evdokimov SA, Yu AI, Shubin AG, Karandaev AS. Algorithm of no-pull control in the continuous mill train. 2015;1–5. https://doi.org/10.1109/SIBCON.2015.7147263.
21. Radionov AA, Gasiyarov VR, Karandaev AS, Usatiy DY, Khramshin VR. Dynamic load limitation in electromechanical systems of the rolling mill stand during biting. 2020;149–54. https://doi.org/10.1109/ICMIMT49010.2020.9041192.

22. Zhang J, Arinez J, Chang Q, Gao R, Xu C. Artificial intelligence in advanced manufacturing: current status and future outlook. J Manuf Sci Eng. 2020;142:1–53. https://doi.org/10.1115/1.4047855.
23. Oliveira E, Miguéis VL, Borges J. Automatic root cause analysis in manufacturing: an overview & conceptualization. J Intell Manuf. 2022;33:1–18. https://doi.org/10.1007/s10845-022-01914-3.
24. Giannetti C, Ransing R, Ransing MR, Bould DC, Gethin DT, Sienz J. A novel variable selection approach based on co-linearity index to discover optimal process settings by analysing mixed data. Comput Ind Eng. 2014;72:217–29. https://doi.org/10.1016/j.cie.2014.03.017.
25. Steenwinckel B. Adaptive anomaly detection and root cause analysis by fusing semantics and machine learning, 272–282 (2018). https://doi.org/10.1007/978-3-319-98192-5_46
26. Weichert D, Link P, Stoll A, Rüping S, Ihlenfeldt S, Wrobel S. A review of machine learning for the optimization of production processes. Int J Adv Manuf Technol. 2019;104:1889–902. https://doi.org/10.1007/s00170-019-03988-5.
27. Chittilappilly A, Subramaniam K. Svm based defect detection for industrial applications, 2017;1–5. https://doi.org/10.1109/ICACCS.2017.8014696
28. Bartova, B., Bína, V.: Early defect detection using clustering algorithms. Acta Oeconomica Pragensia, 1, 3–20 (2019) https://doi.org/10.18267/j.aop.613
29. Kumar KS, Bai MR. Lstm based texture classification and defect detection in a fabric. Measurement Sens. 2023;26:100603. https://doi.org/10.1016/j.measen.2022.100603.
30. Tang B, Chen L, Sun W, Lin Z-k. Review of surface defect detection of steel products based on machine vision. IET Image Proc. 2022. https://doi.org/10.1049/ipr2.12647
31. Huang Z, Wu J, Xie F. Automatic recognition of surface defects for hot-rolled steel strip based on deep attention residual convolutional neural network. Mater Lett. 2021. https://doi.org/10.1016/j.matlet.2021.129707.
32. Liu Y, Xu K, Xu J. Periodic surface defect detection in steel plates based on deep learning. Appl Sci. 2019. https://doi.org/10.3390/app9153127.
33. Shin H-C, Roth HR, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, Summers RM. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging. 2016;35:1285–98. https://doi.org/10.1109/TMI.2016.2528162.
34. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Commun ACM. 2017;60:84–90. https://doi.org/10.1145/3065386.
35. Yamashita R, Nishio M, Do R, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Imaging. 2018;9:611–29. https://doi.org/10.1007/s13244-018-0639-9.
36. Wang J, Yu L-C, Lai K, Zhang X. Dimensional sentiment analysis using a regional cnn-lstm model. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016;225–230 https://doi.org/10.18653/v1/P16-2037
37. Goodfellow I, Bengio Y, Courville A. Deep Learning, 2016;326–366. MIT Press
38. Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. J Big Data. 2016;3:1–40. https://doi.org/10.1186/s40537-016-0043-6.
39. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions, 2015;1–9. IEEE. https://doi.org/10.1109/CVPR.2015.7298594
40. Choi R, Coyner A, Kalpathy-Cramer J, Chiang M, Campbell J. Introduction to machine learning, neural networks, and deep learning. Trans Vision Sci Technol. 2020;9:1–12. https://doi.org/10.1167/tvst.9.2.14.
41. Jebb AT, Tay L, Wang W, Huang Q. Time series analysis for psychological research: examining and forecasting change. Front Psychol. 2015;6:727. https://doi.org/10.3389/fpsyg.2015.00727.
42. Brigham EO, Morrow RE. The fast fourier transform. IEEE Spectr. 1967;4:63–70. https://doi.org/10.1109/MSPEC.1967.5217220.
43. Cao D, Liu J. Research on dynamic time warping multivariate time series similarity matching based on shape feature and inclination angle. J Cloud Comput. 2016;5:11. https://doi.org/10.1186/s13677-016-0062-z.
44. Yang C-L, Yang C-Y, Chen Z-X, Lo N-W. Multivariate time series data transformation for convolutional neural network. 2019;188–92. https://doi.org/10.1109/SII.2019.8700425.
45. Park C, Lee D. Classification of respiratory states using spectrogram with convolutional neural network. Appl Sci. 2022;12:1895. https://doi.org/10.3390/app12041895.
46. Todeschini G, Kheta K, Giannetti C. An image-based deep transfer learning approach to classify power quality disturbances. Electric Power Syst Res. 2022;213: 108795. https://doi.org/10.1016/j.epsr.2022.108795.
47. Balouji E, Salor O. Classification of power quality events using deep learning on event images. 2017:216–21. https://doi.org/10.1109/PRIA.2017.7983049.
48. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13:281–305. https://doi.org/10.1109/PRIA.2017.7983049.
49. Afaq S, Rao S. Significance of epochs on training a neural network. Int J Sci Technol Res. 2020;9:485–8.
50. Kandel I, Castelli M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express. 2020;6:312–5. https://doi.org/10.1016/j.icte.2020.04.010.
51. Berrar D. Cross-validation. Encyclopedia Bioinform Comput Biol. 2018;1:542–5. https://doi.org/10.1016/B978-0-12-809633-8.20349-X.
52. Costa MGF, Campos JPM, Aquino G, Albuquerqueereira WC, Filho CFC.: Evaluating the performance of convolutional neural networks with direct acyclic graph architectures in automatic segmentation of breast lesion in us images. BMC Medical Imaging, 2019;19:85. https://doi.org/10.1186/s12880-019-0389-2