

Review

Survey on AI Applications for Product Quality Control and Predictive Maintenance in Industry 4.0

Tojo Valisoa Andrianandrianina Johanesa ^{1,2,*} , Lucas Equeter ²  and Sidi Ahmed Mahmoudi ¹ 

¹ ILIA Laboratory, Faculty of Engineering, University of Mons, 7000 Mons, Belgium; sidi.mahmoudi@umons.ac.be

² GMECA Laboratory, Faculty of Engineering, University of Mons, 7000 Mons, Belgium; lucas.equeter@umons.ac.be

* Correspondence: tojovalisoa.andrianandrianinajohanesa@umons.ac.be

Abstract: Recent technological advancements such as IoT and Big Data have granted industries extensive access to data, opening up new opportunities for integrating artificial intelligence (AI) across various applications to enhance production processes. We cite two critical areas where AI can play a key role in industry: product quality control and predictive maintenance. This paper presents a survey of AI applications in the domain of Industry 4.0, with a specific focus on product quality control and predictive maintenance. Experiments were conducted using two datasets, incorporating different machine learning and deep learning models from the literature. Furthermore, this paper provides an overview of the AI solution development approach for product quality control and predictive maintenance. This approach includes several key steps, such as data collection, data analysis, model development, model explanation, and model deployment.

Keywords: Industry 4.0; artificial intelligence; product quality control; predictive maintenance



Citation: Andrianandrianina Johanesa, T.V.; Equeter, L.; Mahmoudi, S.A. Survey on AI Applications for Product Quality Control and Predictive Maintenance in Industry 4.0. *Electronics* **2024**, *13*, 976. <https://doi.org/10.3390/electronics13050976>

Academic Editor: Chunping Li

Received: 21 January 2024

Revised: 26 February 2024

Accepted: 27 February 2024

Published: 4 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The industrial sector and its actors have always been in a state of continuous evolution, primarily driven by competitiveness and a desire for efficiency gains. The fourth industrial revolution emerged in response to rapid technological advances. This has led to a series of innovative practices known today as Industry 4.0. The objective of this transformation is to create an interconnected system where automated means of production are able to communicate with each other and with humans through message exchanges [1]. Technologies such as Internet of Things (IoT), Cloud Computing, and Big Data play a crucial role in driving this intelligent production [1]. As a result of automating and monitoring production processes through these technologies, large amounts of data have been made available [2]. This data availability opens up opportunities for the use of Data Analysis and AI to create various applications aiming to improve production processes. These applications have great potential to further automate and revolutionize applications in supply chain optimization [3], energy management [4], product quality control [5], and predictive maintenance [6,7].

Product quality control plays an essential role in manufacturing, as the main goal of most industries is to deliver high-quality products to their customers. Defective products detected at the end of the production can not be commercialized and could lead to significant revenue losses. Detecting signs of defects as soon as they appear on production lines is crucial in avoiding significant losses. Constant maintenance that ensures the quality of the equipment and machines is also very important for manufacturing processes. Addressing equipment failure post-incident prolongs production downtime and creates significant costs. A more efficient method involves adopting a preventive strategy to anticipate failures before they occur. Predictive maintenance is a preventive measure that employs machine condition data to anticipate equipment failures [6].

Statistical process control (SPC) has been used to monitor production processes [8]. It enables proactive measures to prevent defects and deviations, supporting a joint optimization of maintenance and quality [9]. Control charts are used in SPC to monitor variations in production efficiency, and detect potential deviations from established quality standards [10]. However, the analysis of these charts is complex and requires significant statistical knowledge [10]. The integration of AI can further enhance the capabilities of SPC by having a deeper understanding of the data. Industry 4.0 enables data collection through sensors distributed on production lines, and machines are now capable of recording data during the production cycle. These data can then be analyzed and used to train AI models to enhance manufacturing defect detection and machine failure prediction.

Research has been carried out on the usage of AI for the enhancement of product quality control and predictive maintenance. This paper presents a general overview as well as an experimental comparative study of existing approaches on two datasets. Furthermore, we provide an overview of the approach for developing AI solutions for these two applications.

We begin with a review of AI applications for product quality control and predictive maintenance in Section 2. In Section 3, we present an experimental study on two datasets, one related to product quality control and another to predictive maintenance. Finally, Section 4 focuses on providing an overview on the process of development of these AI solutions.

2. Literature Review

This section reviews the current applications of AI in product quality control and predictive maintenance, focusing on suitable data types and exploring various machine learning (ML) and deep learning (DL) techniques proposed in the literature.

2.1. Product Quality Control

Product quality management plays a fundamental role in the manufacturing process. It not only ensures conformity with standards during product commercialization, but also establishes customer trust. There are many studies that have been carried out about the usage of AI for product quality control, and they can be categorized into two distinct categories: defect detection and defect prediction.

2.1.1. Defect Detection

Defect detection is a critical stage of the quality control (QC) process; it enables decisions on whether a product should be approved or rejected. The work that has been carried out in this field gives significant attention to automatic defect inspection using computer vision (CV) to examine produced parts. Recent studies have primarily relied on images of the product for visual defect identification. The annotation of images obtained from both defective and non-defective parts is a fundamental requirement for training good AI models. Examples of defects detected from images using AI include electronics connectors defects [11], background texture defects [12], scratch defects [13], and surface defects [14,15].

2.1.2. Defect Prediction

Defect prediction involves monitoring the production process to predict the quality of the produced parts. The quality of products is strongly influenced by the process parameters used. The majority of the work present in the literature focuses on using process parameters along with the outcomes of quality inspections to develop AI prediction models. The data used in these works typically consist of numerical data recorded from sensors placed on the production lines during the manufacturing process. Examples of applications in this category include porosity detection in aluminum wheels [16] and the detection of geometrical defects in extruded tubes [17]. Several studies are specifically related to plastic injection molding [5,18–20].

2.2. Predictive Maintenance

Maintenance plays a crucial role in the industry by ensuring the proper functioning of machines. Predictive maintenance [21] involves predicting when a machine is likely to experience a failure. This strategy relies on data analysis techniques and AI to predict machine failures. By using data related to the operating conditions of the machines, AI models can be employed to identify anomalies and determine when maintenance is necessary. Many studies have focused on predictive maintenance using AI in the literature. They can be categorized into failure prediction and remaining useful life (RUL) predictions. Failure predictions focus on identifying imminent failures, while RUL predictions determine the remaining time before a failure occurs.

2.2.1. Failure Prediction

Failure prediction consists of real-time monitoring of machine conditions using sensor data and employing AI models to detect signs of future failure. Several research studies on failure prediction have been proposed, using numerical data and time-series data collected from IoT sensors associated with machine operating conditions. The data should represent the dynamic behavior and machine conditions over time during a given machine's operational phase. Numerical data can provide insights into the quantitative aspects of machine performance, while time-series data enable the analysis of temporal patterns to identify trends and anomalies preceding equipment failure. For example, the authors of [22] predicted machine component failures using telemetry measurements, maintenance history, and machine specifications. The authors of [23] employed time-series data to predict tool wear, gearbox, and bearing faults. The authors of [24] focused on predicting failures in rotating machines by analyzing vibration signals. The authors of [25] predicted the degradation of cutting tools.

2.2.2. Remaining Useful Life Prediction

Remaining useful life (RUL) prediction enables the estimation of the remaining operating time of machines before a failure or breakdown occurs. Knowing the remaining useful life of equipment helps in the effective planning of maintenance operations. Two major categories of data are frequently employed in the literature: numerical data and time-series data. For instance, the authors of [26] predicted the RUL of rolling-element bearings and milling cutters using time-series data related to their operating conditions. The authors of [27], for example, forecast the RUL of plastic injection machines using numerical sensor data. Several works, including [28–32], have employed the C-MAPSS [33] time-series dataset to predict the RUL of turbofan engines.

2.3. AI Models for Product Quality Control and Predictive Maintenance

In this section, we present existing machine learning and deep learning methods that focus on data used in product quality control and predictive maintenance.

2.3.1. Machine Learning Methods

Machine learning is a branch of AI that enables computers to learn from data in order to acquire decision-making capabilities and perform tasks without human intervention [34]. ML models are trained on a set of examples, known as training data, in order to be able to make predictions on newer or unseen data.

To predict defects and failures in manufacturing, the data used for model development should cover both normal conditions and instances of failures or defects. Various ML algorithms have been proposed in the literature in order to create these models:

- **Support vector machine (SVM):** SVM is an ML technique commonly employed for classification and regression tasks [35]. In classification, SVM creates hyperplanes to effectively separate different classes using support vectors [35]. In regression, the objective is to identify a function that closely matches data points within a defined margin [35]. SVM has been employed in various studies including predicting part quality

in plastic injection [5], identifying defects in laser additive manufacturing [36], predicting tool wear during milling operations [25], and estimating the RUL of machine components [37].

- **K-nearest neighbors (KNN):** KNN [38] is a machine learning technique used for classification and regression tasks. To classify a new data point, the algorithm looks at the K closest data points from the training set and assigns the majority class among those neighbors to the new data point [38]. For regression, it predicts the value based on the average of the K nearest data point values [38]. It does not require training but its performance can be affected by the chosen K value and distance metric. In [39], KNN was used for fabric defect detection based on features extracted from thermal camera images. In the context of additive manufacturing quality control, KNN demonstrated effective porosity prediction, as presented in [40]. The effectiveness of KNN was also highlighted in [41], a comparative study on predictive maintenance.
- **Naive Bayes:** Naive Bayes [42] is a probabilistic machine learning algorithm that relies on Bayes' theorem. It assumes feature independence given the class label. The algorithm calculates probabilities for a data point belonging to each class, and predicts the class with the highest probability. Naive Bayes can also be used for regression tasks [43]. The Naive Bayes algorithm, combined with particle swarm optimization (PSO) [44], was used in [45] to effectively detect product defects. In [46], a Naive Bayes approach using vibration signals successfully identified specific bearing faults.
- **Regression (linear and logistic):** Linear regression [47] is an ML algorithm that searches for a linear relationship between input features and the target variable by fitting a straight line to the data points. The effectiveness of this algorithm depends on the linearity assumption of the data. Logistic regression [48], on the other hand, is used for classification by calculating the probability of belonging to a class using a logistic function that produces values between 0 and 1. In [49], a logistic regression model is proposed for predicting product quality in the rolling process. Linear regression is applied in [50] to forecast machine failure in a turbine generator for maintenance scheduling in oil and gas platforms. The study presented in [51] employs multiple linear regression (MLR) [52] to estimate the RUL of bearings based on vibration data.
- **Decision tree:** Decision tree [53] is an ML technique used for both classification and regression tasks. It constructs a model in the form of a tree, where each internal node represents a feature and each leaf node represents a class or a predicted value. It recursively splits the data based on the feature that provides the best separation at each node up to a certain stopping criterion. In [54], the authors used a J48 decision tree model to predict part quality in the injection molding process. The study presented in [55] employed decision trees in combination with a genetic algorithm to predict the RUL of an aircraft air conditioning system.
- **Ensemble learning:** Ensemble methods [56] are machine learning techniques that combine predictions from multiple models to predict the target variable. Predictions are often combined using voting for classification tasks and averaging for numerical value prediction. There are different ensemble methods, such as bagging, XGBoost [57], random forest [58], and gradient boosting machines [59]. XGBoost successfully predicts manufacturing defects in [16,19]. In [60], a gradient boosting model is suggested for predicting steel product quality. In predictive maintenance, random forest models are used to predict the RUL of machines in [27,61].

2.3.2. Deep Learning Models

Deep learning is an advanced approach that uses artificial neural networks with multiple interconnected layers of neurons [62]. It is particularly effective for processing large amounts of data and can handle both structured and unstructured data [63]. There are different proposed architectures of neural networks, each adapted to particular types of data. In this section, we will present different architectures that have been proposed for product quality control and predictive maintenance in the literature.

- **Multi-layer perceptron (MLP):** MLP [64] models are the most well known deep learning models. MLP consists of interconnected layers of neurons. Neurons in intermediate layers assign weights to inputs from the previous layer, sum them with a bias, and apply an activation function. The last layer gives the output value depending on the nature of the problem, whether it is a classification or regression task. The study presented in [20] applied MLP in an online defect detection system for the injection molding process. In [32], MLP was employed to predict the RUL of aircraft turbofan engines. An MLP model demonstrated strong performance in predicting machine component failures in [22].
- **Convolutional neural network (CNN):** Convolutional neural networks [65] are a widely employed DL method for image processing applications, such as image classification, object detection, and image segmentation. Its key component, the convolutional layer, uses filters to detect patterns in input images through convolutions. CNNs can learn features automatically without the need for manual features extraction. It is common in the literature to use transfer learning, which involves taking a pretrained CNN model on one task and adapting it to a new, similar task. Many pretrained model architectures have been proposed [66]. In [12], a CNN model was used for background-texture-based defect detection. In [14], a modified You Only Look Once (YOLO) approach [67] was transformed into a fully convolutional model; this was introduced for real-time surface defect detection in steel strips. The work in [15] reviews and presents other examples of defect detection in images of products. Regarding predictive maintenance, the authors of [30,68] applied CNNs to predict RUL.
- **Recurrent neural network (RNN):** An RNN [69] is a type of network that was introduced to process sequential data such as time-series data or text. RNNs take an ordered sequence of data as input to predict one or more output values. They are designed to extract important information from sequential data and use it for prediction. There are different types of RNN architectures, including vanilla RNN [69], long short-term memory (LSTM) networks [70], and gated recurrent units (GRUs) [71]. The authors of [72] proposed an approach to predict the RUL of aircraft engines using a basic LSTM model. The study presented in [23] demonstrates the effectiveness of a system based on a gated recurrent unit (GRU) model for predicting tool wear, gearbox faults, and bearing faults.
- **Generative adversarial network (GAN):** A GAN [73] is a type of deep artificial network that enables the generation of synthetic data from a given real dataset. This model primarily consists of two components: a generator and a discriminator, each with a specific role during the model training process. The generator transforms a random noise vector into synthetic data that resembles the original dataset. On the other hand, the discriminator is used to differentiate between the synthetic and real sample by classifying them accordingly. A steel surface defect detection method utilizing GANs was introduced in [74]. Works such as [75–77] propose approaches using GANs for predicting the RUL of machines using data extracted from multiple sensors.
- **Autoencoder:** Autoencoders [78] are artificial neural networks consisting of two main components: the encoder and the decoder. The encoder processes input data and transforms them into a lower-dimensional encoded representations within a latent space. The decoder performs the inverse operation by taking the encoded representation and decoding it to reconstruct the original data. Its role is to recreate a version of the initial input as close as possible. The main objective of the autoencoder is to minimize the difference between the input data and the reconstructed data. An autoencoder-based model is used in [18] for quality prediction in the injection molding process. The study presented in [79] proposed a deep learning model composed of a variational autoencoder (VAE) [80] and a recurrent neural network (RNN) for predicting the RUL of machines. The VAE is used to reduce the dimensionality of the data and extract features from sensor data, while the RNN is used for RUL prediction.

- **Transformer:** Transformers [81] were initially created for natural language processing tasks such as language translation and text summarization. Similarly to autoencoders, transformers mainly consist of two parts: an encoder and a decoder. These are both composed of multiple layers of self-attention and feed-forward neural networks. The transformer is designed to learn to produce outputs by focusing only on relevant information extracted from the input data using the attention mechanism. Recently, several studies have explored the application of transformers in other tasks, such as image processing [82] and time-series data analysis [83]. Transformers enable parallel processing of data, overcoming the sequential processing limitations of RNNs [84]. Studies like [85,86] have used transformers for surface defect detection. In addition, a system employing the transformer was proposed for predicting RUL of Li-ion batteries in [87].

2.4. Conclusions and Discussion

After taking a look over the current State-of-the-art, we remark that models based on CNNs have been widely used to detect defects from images of products in the literature. These models appear to be well-suited for image processing. Other models based on GANs and transformers have also recently emerged for defect detection. ML approaches such as SVMs and KNN have also been explored, but these models require prior feature extraction from images. Defect detection can assist operators in conducting quality control at the end of production. However, computer vision techniques can only be applied to visible surface defects and can not identify non-visual surface defects. Unfortunately, other performance-related quality criteria should be considered because quality is not just about external appearance.

In the literature, defect prediction [16–20] from process data is carried out using structured numerical data. XGBoost, gradient boosting, and random forest are among the most commonly used ML models. Many studies employ ensemble learning, as it can provide better performance by combining predictions from multiple models, instead of a single one. In other studies that use a DL approach, MLP models were employed as they are suitable for structured data.

Data used in predictive maintenance typically consist of numerical data or multivariate time-series. These kinds of data are generally related to the operating conditions of the machines. In terms of numerical data, commonly used methods include ensemble learning (XGBoost, random forest, gradient boosting) and MLP. For time-series data, existing approaches often use RNN models. Other recent approaches suggest the use of transformer models. AI-based predictive maintenance approaches have been demonstrated to be quite effective in the literature, whether for predicting machine failure or RUL.

Despite the advancements in the application of AI for product quality control and predictive maintenance, we have identified some limitations in existing works in the literature:

- **Data imbalance:** Data imbalance is a prevalent challenge in applying AI to predictive maintenance and product quality control. Machine failure instances and defective product examples are often rare compared to normal cases, resulting in imbalanced datasets. AI models are designed to minimize overall error rates and may perform well on normal cases, but struggle with predicting machine failures and identifying defects. For example, this can be observed in [16], where the performance of detecting normal cases is significantly higher than that of detecting defects due to data imbalance. Various techniques [88] have been employed to address this issue, but data imbalance remains a significant challenge for AI applications in many fields.
- **Explainability and interpretability:** Some traditional machine learning models, such as decision trees, are explainable by default. However, more complex models like ensemble and deep learning models are not inherently explainable. Deep learning models are often viewed as complex black-box models, presenting a challenge in understanding decision-making processes and internal logic. This opacity may cause regulatory compliance issues regarding the accountability and transparency of their

decisions. Explainability and interpretability techniques [89] should be applied to establish confidence in the decision of the model. Furthermore, these techniques can help identify the root causes of a product defect or a machine failure. A better understanding of these causes could aid in optimizing the production process by adjusting parameters related to these incidents. Most of the existing studies [11,12,14–18,22–30,32] on product quality control and predictive maintenance have not addressed the explainability of the proposed solutions.

- **Real-time detection instead of prediction:** Most existing works related to failure prediction [22–25] primarily focus on predicting the current state of equipment rather than forecasting its future state; while these approaches perform well in real-time failure detection, they do not allow for anticipating failures, as the prediction occurs after the failure has already occurred. A reliable predictive maintenance system should be capable of predicting future failures based on the current state, equipment history, and other information related to the environmental conditions of the equipment.
- **Domain dependency and the need for industry-specific datasets:** Domain dependence is one of the main challenges of applying AI for predictive maintenance and product quality control. It can be observed in the literature that the field of predictive maintenance is not only relevant to the manufacturing industry but also to other sectors. The C-MAPSS dataset [33], which is related to aircraft engines, has been used to evaluate predictive maintenance proposals [28–30,32], but this dataset is not specifically related to the manufacturing industry. It is challenging to ensure that the proposed approaches will be effective when applied to real-world cases in industry. Transfer learning [90] techniques allow adapting a model built in one domain to another, but the input data of the model could not be the same. Domain dependence is a persistent issue in the application of AI in industry, and this can impact the performance of the model. A model may perform well in benchmarking data but exhibit very poor performance in a real-world industrial application.
- **Overlooking component interactions:** Existing works [22–25] on predictive maintenance focus on predicting the failures of individual components. The degradation of one component may be linked to other components, further complicating the identification of failure causes. Failing to account for these interactions can lead to inaccurate failure predictions. It is crucial to monitor the overall machine state and consider interactions between components in the design of AI models for predictive maintenance.
- **Single-quality criterion consideration:** One limitation of existing works on defect prediction [16–20] is that they typically focus on a single quality criterion. To establish an effective quality control system, the dataset should include data from partial quality inspections that address all relevant criteria. Sources of defects may differ across various quality criteria, introducing an additional challenge in the application of AI for predicting manufacturing defects.

3. Experimental Study

To ensure a comprehensive review, we conducted an experimental study that includes various approaches highlighted in the previous section. This section enhances the theoretical insights from Section 2 by applying existing AI models in real-world industrial conditions. For this purpose, we employed two public datasets: one associated with predicting product quality in the plastic injection process and another related to predicting machine component failures. We selected these datasets because they originate from real-world industry cases. In the two datasets, we have a categorical target, which indicates a classification problem. Therefore, we chose to use precision, recall, and F-score as metrics [91], which are commonly used in the literature for evaluating model performance in a classification problem.

3.1. Quality Prediction in Plastic Injection

This experimental study is about quality prediction in plastic injection molding process. The data were collected from sensors in the injection machine during the production of plastic road lenses [92]. The dataset comprises 1451 records and consists of 13 process parameters, including the temperature of the melted material, mold temperature, filling time, plasticizing time, cycle time, closing force, peak closing force value, peak torque value, average torque value, peak back pressure value, peak injection pressure value, screw position at the end of holding pressure, and injection volume. In addition to these parameters, the dataset contains an attribute indicating the quality of the lenses. The authors of [92] have defined four quality classes based on the standard “UNI EN 13201-2:2016” for lenses in motorized roads. According to this standard, the general uniformity U_0 of the lens should be greater than 0.4 [92]. Samples with U_0 less than 0.4 are categorized as “Waste” (Class 1) and should be discarded for not meeting the standard. Those with a uniformity between 0.4 and 0.45 are labeled as “Acceptable” (Class 2), meeting the standard but falling short of the company’s higher quality target. “Target” (Class 3) includes samples with a uniformity between 0.45 and 0.5, and are considered optimal. Samples with U_0 greater than 0.5 are labeled as “Inefficient” (Class 4) and should be avoided, as producing lenses with such uniformity exceeds standard requirements. The class distribution in the dataset is illustrated in Figure 1, which shows that the dataset is quite balanced.

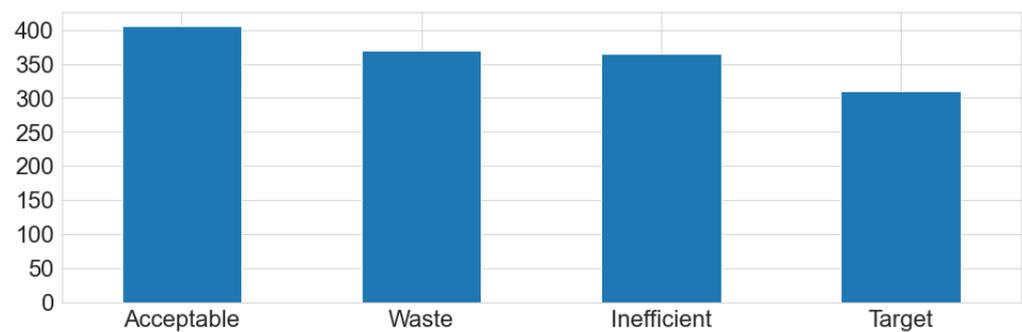


Figure 1. Class distribution in the road lens dataset.

We conducted a performance comparison of various ML and DL models. We selected a set of models that are commonly employed in quality control applications for numerical data. The set of models includes SVM, logistic regression, decision tree, random forest, XGBoost, KNN, Naive Bayes, and MLP. Time-series models like recurrent neural networks were not applicable in this study due to the absence of timestamp information in the dataset.

We divided the dataset into a training set, a validation set, and a test set, and the size of each subset is presented in Table 1. Figure 2 illustrates the class distribution within each subset of the data.

The validation set is employed to control overfitting during the training of the MLP model. To ensure a fair comparison of models, machine learning models were trained exclusively on the training set.

Table 1. Number of records in each subset of the road lens dataset.

Data Partition	Size
Training set	870 (60%)
Validation set	290 (20%)
Test set	291 (20%)
Total	1451

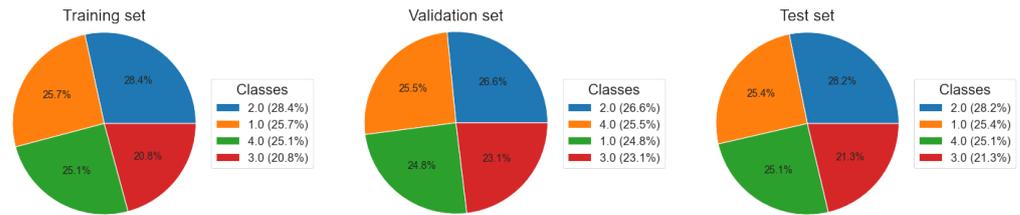


Figure 2. Class distribution in the training, validation, and test sets: road lens dataset.

The results of the performance evaluation for various models on the test set are presented in Table 2. Each model was fine-tuned by manually adjusting hyperparameters. The results show that ensemble models present good performances in terms of precision, recall, and F1-score. Random forest shows the best scores for all performance metrics and was also identified as the top-performing model by the authors in [92]. Following random forest, MLP demonstrates a value of 0.95 for all three metrics, followed by XGBoost, gradient boosting, and extra trees. The random forest was configured with 100 decision trees. The architecture of the MLP model is presented in Table 3. It was trained for 120 epochs and optimized using the RMSprop (root mean squared propagation) optimization algorithm with a learning rate of 0.00658.

Table 2. Performance of models on the test set: road lens quality prediction.

Model	Precision	Recall	F1-Score
SVM	0.87	0.86	0.86
Naive Bayes	0.85	0.83	0.83
KNN	0.92	0.91	0.91
Logistic regression	0.80	0.79	0.79
Decision tree	0.91	0.91	0.91
Random forest	0.98	0.98	0.98
XGBoost	0.94	0.93	0.93
Gradient boosting	0.95	0.94	0.94
Extra trees	0.94	0.93	0.93
MLP	0.95	0.95	0.95

Table 3. MLP model architecture summary: road lens quality prediction.

Layer (Type)	Output Shape	Number of Parameters
dense (Dense)	(None, 80)	1120
dense_1 (Dense)	(None, 50)	4050
dropout (Dropout)	(None, 50)	0
dense_2 (Dense)	(None, 4)	204
Total params: 5374		
Trainable params: 5374		
Non-trainable params: 0		

Models such as KNN and decision tree also showed good F-score values of 0.91, while the remaining models—SVM, Naive Bayes, and logistic regression—presented relatively lower performances. It can be assumed that these models are simpler and struggle to identify complex relationships among different features in the dataset.

Figure 3 illustrates the confusion matrix for random forest and MLP. It shows that the two models are able to effectively determine the quality of the parts with only seven misclassified parts for random forest and 15 misclassified parts for MLP. The result indicates that a well-balanced dataset contributes to achieving high-performing models.

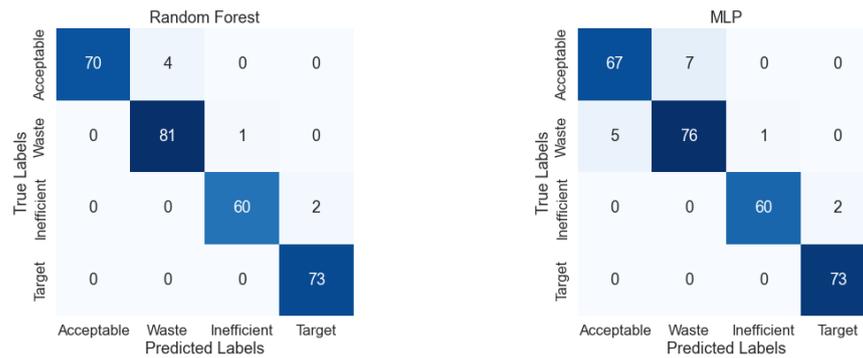


Figure 3. Confusion matrices of random forest and MLP: road lens quality prediction.

3.2. Machine Component Failure Prediction

The objective of this use case is to predict machine component failures based on real-time data collected from the machine and its operating conditions. This allows for anticipating failures before they occur and planning maintenance operations in advance.

The dataset used in this section, provided by Microsoft [93], includes detailed information related to the operational conditions of 100 machines in an anonymous industrial process. The dataset was also used in the work of [22], where the authors proposed a prediction of failures for four machine components. The dataset consists of five distinct data sources [22]:

- **Telemetry:** includes measurements of machine pressure, vibration, rotation, and voltage.
- **Errors:** log of recorded machine errors.
- **Machine:** provides machine characteristics such as age and model.
- **Maintenance:** contains the history of all machine component replacements.
- **Failures:** information on the history of failed component replacements.

The initial dataset comprises 876,100 records of telemetry measured at hourly intervals. The date and time are rounded to the nearest hour for errors, maintenance, and failures data.

Since we have five data sources, it is necessary to gather the different data sources to create features that can best describe the health condition of a machine at a given time. For this purpose, we applied the same feature engineering techniques used in [22] to merge the different data sources. To achieve this, additional information was extracted from the initial data sources to enrich the dataset [22]. The means and standard deviations of telemetry measurements over the previous 3 h and 24 h were computed to create a short-term and long-term history of telemetry data. This allows for better anticipation of failures and provides an early warning in case of a failure [22]. The “errors” data source helped determine the number of errors of each type in the last 24 h for each machine [22]. From the “maintenance” data source, the number of days elapsed since the last component replacement was calculated [22]. The “Failures” data source was used to create the label. Records within a 24 h window preceding the replacement of a failed component were labeled with the corresponding component name (comp1, comp2, comp3, or comp4), while other records were labeled as “None” [22]. A more detailed description of the applied feature engineering process can be found in [22].

After the feature engineering, the final dataset comprises 290,642 records with 29 features, including machine information such as the machine identifier (machineID), age, and the model of the machine; 3 h and 24 h rolling measurements for voltage, rotation rate, pressure, and vibration; error counts during the last 24 h for different error types; time since the last replacement for each component; and a ‘datetime’ column indicating the registration of each record at regular three-hour intervals.

We compared multiple ML and DL models by evaluating them on the same test set. As the dataset includes a “datetime” column indicating the timestamp of the data, we incorporated recurrent neural networks that take a sequence of records as input. To achieve this, we selected a sequence length of eight records to analyze information over the past

24 h. The class of the last record in the sequence is considered as the sequence class. To avoid overlaps, sequences containing a record corresponding to a failure were excluded if the last record had a 'none' class. We excluded these records along with the first seven records for each machine from the test set used for the evaluation of models that take one record at a time. This ensures a fair comparison between models handling a single record and those handling a sequence of eight records.

To develop models, we first divided the data into train, validation, and test sets. To achieve this, we followed the same partitioning as presented in [22]. The records until 31 August 2015 1:00:00 are used as the training set to train the model. Those between 1 September 2015 1:00:00 and 31 October 2015 1:00:00 serve as the validation set, and those starting from 1 November 2015 1:00:00 are reserved to compose the test set.

Table 4 presents the number of records for each subset for both RNN models and other models. The difference in the number of records between RNN models and other models is a result of using sequences of eight records specifically as input for RNN models.

Table 4. Data partitioning for component failure prediction: RNN models vs. other models.

Data Partition	Size (RNN Models)	Size (Other Models)
Training set	189,517 (67%)	193,528 (67%)
Validation set	46,346 (16%)	47,804 (17%)
Test set	47,047 (17%)	47,047 (16%)
Total	281,910	288,379

Figures 4 and 5 depict the distribution of classes in the training, validation, and test subsets for the two cases. We can observe that subsets have similar class distributions which indicates good data partitioning. Class distributions in those subsets are also similar to those of the initial dataset.

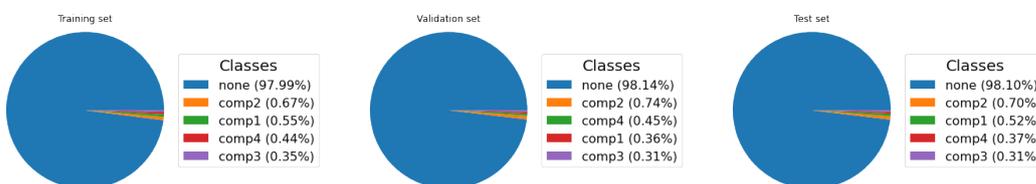


Figure 4. Class distribution in the training, validation, and test sets for RNN models.

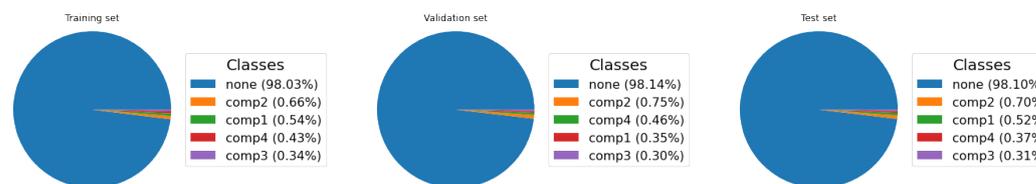


Figure 5. Class distribution in the training, validation, and test sets for other models.

Table 5 presents the performance results of the different models used in our experiment. Fine-tuning of hyperparameters for each model was performed manually. We can observe that the SVM, the ensemble models (random forest, XGBoost, gradient boosting, extra trees), and the DL models (MLP, SimpleRNN, LSTM, and GRU) all demonstrated high performance with an F-score exceeding 0.95. XGBoost and GRU outperform other models with an F-score of 0.98, followed by random forest with an F-score of 0.97. The architecture of the GRU model is presented in Table 6. The GRU model was trained for 50 epochs and optimized using the Adam optimization algorithm with a learning rate of 0.0005. The XGBoost model is set up with 70 estimators and the random forest model is configured with 100 estimators.

Table 5. Performance of models on the test set: component failure prediction.

Model	Precision	Recall	F1-Score
SVM	0.95	0.96	0.96
Naive Bayes	0.61	0.95	0.71
KNN	0.59	0.34	0.39
Logistic regression	0.93	0.84	0.88
Decision tree	0.99	0.94	0.96
Random forest	0.99	0.96	0.97
XGBoost	0.99	0.97	0.98
Gradient boosting	0.98	0.76	0.77
Extra trees	0.98	0.95	0.96
MLP	0.96	0.97	0.96
SimpleRNN	0.98	0.94	0.96
LSTM	0.98	0.95	0.96
GRU	0.99	0.97	0.98

Table 6. GRU model architecture summary: component failure prediction.

Layer (Type)	Output Shape	Number of Parameters
gru (GRU)	(None, 56)	14,784
dense (Dense)	(None, 5)	285
Total params: 15,069		
Trainable params: 15,069		
Non-trainable params: 0		

The confusion matrices for these three models are shown in Figure 6. We can observe that the primary sources of prediction errors are associated with the prediction of failure for components 1 and 3. The results obtained confirm that ensemble learning models could give good results for failure prediction with a numerical dataset. On the other hand, recurrent neural networks could capture relevant information in a time-series for failure prediction.

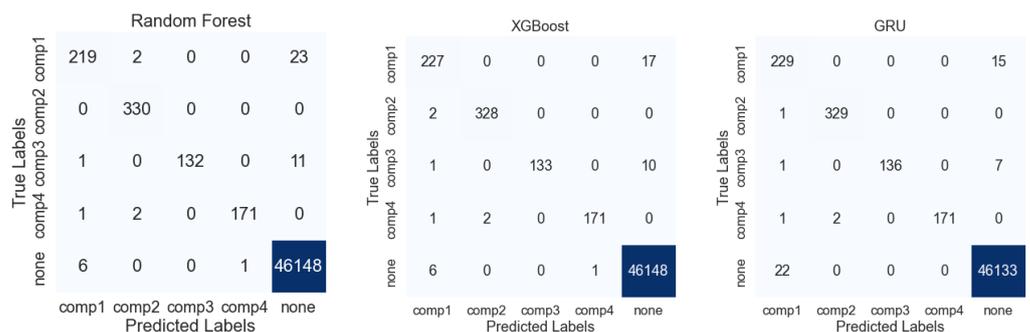


Figure 6. Confusion matrices of random forest XGBoost and GRU: component failure prediction.

4. AI Solution Development for Product Quality Control and Predictive Maintenance

In this section, we will present an overview of AI solution development approach for product quality control and predictive maintenance, building upon the studies conducted in the previous sections. The approach takes into consideration the limitations identified in existing works, specifically addressing data imbalance and the explainability of AI models.

This comprehensive approach encompasses data collection, feature engineering, data pre-processing, data analysis, model development, model explanation, and model deployment. The entire process is summarized in Figure 7.

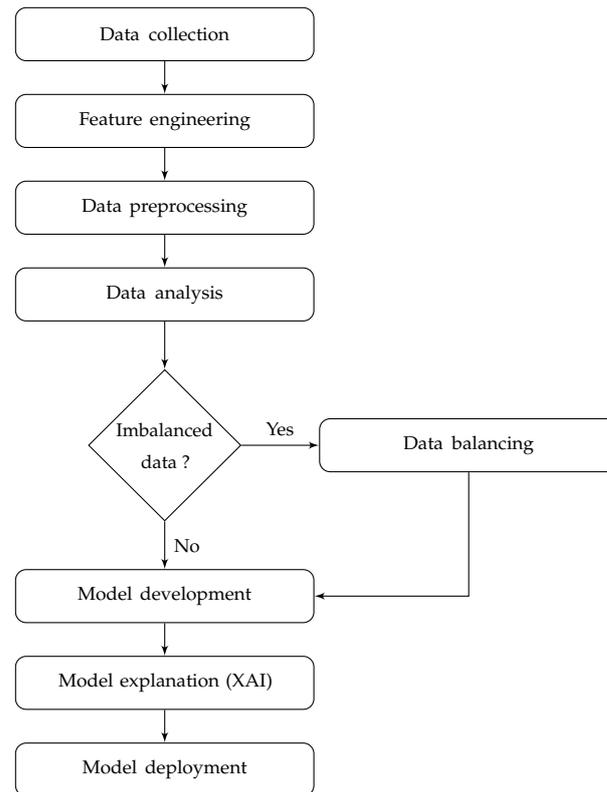


Figure 7. AI solution development approach for product quality control and predictive maintenance.

4.1. Data Collection and Feature Engineering

To develop a data-driven quality control and predictive maintenance system, the first step is data collection. Subsequently, it is essential to aggregate data from various sources through feature engineering, creating a dataset that accurately represents the addressed problem. During the data collection process, data annotation should also be performed. The goal is to provide the model with the most relevant information for detecting and predicting manufacturing defects and machine failures.

4.2. Data Pre-Processing

Before developing an AI model, data pre-processing plays a critical role in cleaning and preparing the data. It involves several essential steps, such as removing missing, incorrect, or outlier values, as well as duplicates, ensuring that the data used are consistent and reliable. Additionally, feature selection plays a significant role in reducing the dimensionality of the dataset. Some features may have little influence on the decision of the model and could potentially degrade its performance. Feature selection methods [94] can help choose the most informative features for the model. Furthermore, data can be measured on different scales or units, potentially leading to biases in the model. Data normalization [95] enables scaling the data to a common range using techniques such as min–max normalization, Z-score normalization, and decimal scaling normalization. This allows models to treat the data fairly and avoid biases caused by measurement differences.

4.3. Data Analysis

Data analysis is an important step in better understanding data and identifying patterns, trends, and relationships between features that can be useful in model development. Data visualization is a commonly used technique to graphically represent data and provide an overview. Graphs like variation plots, histograms, scatter plots, and box plots are often used to explore features and relationships between variables. Correlation analysis [96] can also assist in identifying highly correlated features and removing them by retaining only a representative single feature. Data imbalance is also an important point to analyze.

It is necessary to check whether the data classes are evenly distributed. Significant data imbalance could lead to model performance issues as it may be biased towards the majority classes. In such cases, resampling techniques or synthetic data generation can be used to balance the classes and improve model performance.

4.4. Model Development

Model development involves splitting the pre-processed data into training set, validation set, and test set. The training set is used to train the model by searching for the best configuration. The validation set serves to identify overfitting during the training process, while the test set is employed to assess the model's performance on unseen data. This enables an evaluation of the ability of the model to generalize to new data and measure its real-world performance. Often, the best model type and configuration for a given problem are not known in advance. Developing a quality control and predictive maintenance system requires testing different models. A step of selecting the best-performing models is then performed based on appropriate evaluation metrics. The goal is to develop models capable of accurately predicting manufacturing defects and machine failures.

4.5. Model Explanation

After developing the model, a crucial step is to ensure transparency of the model decisions. Some models, like most traditional ML models, are by default explainable. However, more complex models like DL models require the use of explainability techniques [89]. These techniques play a pivotal role in making the logic behind AI models transparent and understandable for industry professionals. The principle of these methods is to identify the most important parameters in the input data during model prediction. Furthermore, these explainability techniques can help in determining why defects or failures occur by providing insight into the impact of each parameter on the output. Understanding these reasons can improve how machines are used, cut down on failures, and reduce the number of defective products. The selection of the appropriate technique depends on both the type of data and the nature of the model.

4.6. Model Deployment

The final step of the solution involves deploying the models on the production lines. The deployment of AI models in the industry typically occurs on local servers or in Cloud Computing. However, these solutions can lead to latency issues, resulting in a delay between the request for data and receiving the data. This delay can be critical for applications such as quality control and predictive maintenance, where real-time predictions are essential. An alternative to these solutions is Edge AI [97], which involves deploying AI models on embedded resources in production lines to bring sensors closer for data processing. This approach allows for real-time prediction of equipment status without the need to transmit data to a remote server. However, deploying AI models in Edge AI also presents significant challenges concerning resource limitations and the need for lighter AI models. It is imperative to compress and optimize to ensure the deployment of models on resource-constrained edge devices.

5. Conclusions

In this work, we have presented a review of existing research on data-driven product quality control and predictive maintenance using AI. Product quality control facilitates the early detection and prediction of manufacturing defects on production lines. Predictive maintenance allows to predict equipment failures before they occur. Several solutions were published to address these two applications using different AI approaches and demonstrate their performances. However, we have observed that there are gaps that need to be researched in future works. Future research in predictive maintenance and product quality control should prioritize addressing data imbalances in industrial contexts to improve the performance of AI models. Incorporating model explanation techniques is essential to

validate decision-making processes, particularly in opaque models like DL models. Shifting from real-time failure detection to real-time failure forecasting in predictive maintenance is essential, requiring anticipation of failures based on current equipment states, historical data, and environmental conditions. Considering component interactions and expanding defect prediction beyond single-quality criteria are fundamental directions for refining the accuracy of AI applications. We have presented an overview of an approach that includes the various steps required for developing an AI solution for the two applications presented in this paper, taking into account data imbalance issue and model explanation.

Author Contributions: Conceptualization, T.V.A.J., L.E. and S.A.M.; methodology, T.V.A.J., L.E. and S.A.M.; formal analysis, L.E. and S.A.M.; investigation, T.V.A.J., L.E. and S.A.M.; writing—original draft preparation, T.V.A.J.; writing—review and editing, T.V.A.J., L.E. and S.A.M.; visualization, L.E. and S.A.M.; supervision, L.E. and S.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Service Public de Wallonie Recherche under grant n° 2010235–ARIAC by DIGITALWALLONIA4.AI.

Data Availability Statement: Publicly available datasets were used in this study. These datasets can be found here: <https://github.com/airtlab/machine-learning-for-quality-prediction-in-plastic-injection-molding> (Road lens dataset, accessed on 25 September 2023) and https://github.com/ashishpatel26/Predictive_Maintenance_using_Machine-Learning_Microsoft_Casestudy/tree/master/data (Microsoft dataset, accessed on 23 March 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	artificial intelligence
IoT	Internet of Things
SPC	statistical process control
ML	machine learning
DL	deep learning
QC	quality control
CV	computer vision
RUL	remaining useful life
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
SVM	support vector machine
KNN	K-nearest neighbors
MLR	multiple linear regression
MLP	multi-layer perceptron
CNN	convolutional neural network
YOLO	You Only Look Once
RNN	Recurrent Neural Network
LSTM	long short-term memory
GRU	gated recurrent unit
GAN	generative adversarial network
VAE	variational autoencoder

References

1. Moeuf, A.; Lamouri, S.; Pellerin, R.; Tamayo-Giraldo, S.; Tobon-Valencia, E.; Eburdy, R. Identification of critical success factors, risks and opportunities of Industry 4.0 in SMEs. *Int. J. Prod. Res.* **2020**, *58*, 1384–1400. [CrossRef]
2. Kotsiopoulos, T.; Sarigiannidis, P.; Ioannidis, D.; Tzovaras, D. Machine learning and deep learning in smart manufacturing: The smart grid paradigm. *Comput. Sci. Rev.* **2021**, *40*, 100341. [CrossRef]
3. Dash, R.; McMurtrey, M.; Rebman, C.; Kar, U.K. Application of artificial intelligence in automation of supply chain management. *J. Strateg. Innov. Sustain.* **2019**, *14*, 43–53.

4. Laayati, O.; Bouzi, M.; Chebak, A. Smart energy management: Energy consumption metering, monitoring and prediction for mining industry. In Proceedings of the 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, 2–3 December 2020; pp. 1–5.
5. Silva, B.; Sousa, J.; Alenya, G. Machine learning methods for quality prediction in thermoplastics injection molding. In Proceedings of the 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 9–10 December 2021; pp. 1–6.
6. Abidi, M.H.; Mohammed, M.K.; Alkhalefah, H. Predictive maintenance planning for industry 4.0 using machine learning for sustainable manufacturing. *Sustainability* **2022**, *14*, 3387. [[CrossRef](#)]
7. Colantonio, L.; Equeter, L.; Dehombreux, P.; Ducobu, F. A systematic literature review of cutting tool wear monitoring in turning by using artificial intelligence techniques. *Machines* **2021**, *9*, 351. [[CrossRef](#)]
8. Oakland, J.; Oakland, J.S. *Statistical Process Control*; Routledge: London, UK, 2007.
9. Dutoit, C.; Dehombreux, P.; Lorphèvre, E.R.; Equeter, L. Statistical process control and maintenance policies for continuous production systems subjected to different failure impact models: Literature review. *Procedia CIRP* **2019**, *86*, 55–60. [[CrossRef](#)]
10. Ramezani, J.; Jassbi, J. Quality 4.0 in action: Smart hybrid fault diagnosis system in plaster production. *Processes* **2020**, *8*, 634. [[CrossRef](#)]
11. Wang, K.J.; Fan-Jiang, H.; Lee, Y.X. A multiple-stage defect detection model by convolutional neural network. *Comput. Ind. Eng.* **2022**, *168*, 108096. [[CrossRef](#)]
12. Wang, T.; Chen, Y.; Qiao, M.; Snoussi, H. A fast and robust convolutional neural network-based defect detection model in product quality control. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3465–3471. [[CrossRef](#)]
13. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [[CrossRef](#)]
14. Li, J.; Su, Z.; Geng, J.; Yin, Y. Real-time detection of steel strip surface defects based on improved yolo detection network. *IFAC-PapersOnLine* **2018**, *51*, 76–81. [[CrossRef](#)]
15. Chen, Y.; Ding, Y.; Zhao, F.; Zhang, E.; Wu, Z.; Shao, L. Surface defect detection methods for industrial products: A review. *Appl. Sci.* **2021**, *11*, 7657. [[CrossRef](#)]
16. Uyan, T.C.; Otto, K.; Silva, M.S.; Vilaça, P.; Armakan, E. Industry 4.0 foundry data management and supervised machine learning in low-pressure die casting quality improvement. *Int. J. Met.* **2023**, *17*, 414–429. [[CrossRef](#)]
17. García, V.; Sánchez, J.S.; Rodríguez-Picón, L.A.; Méndez-González, L.C.; Ochoa-Domínguez, H.d.J. Using regression models for predicting the product quality in a tubing extrusion process. *J. Intell. Manuf.* **2019**, *30*, 2535–2544. [[CrossRef](#)]
18. Jung, H.; Jeon, J.; Choi, D.; Park, J.Y. Application of machine learning techniques in injection molding quality prediction: Implications on sustainable manufacturing industry. *Sustainability* **2021**, *13*, 4120. [[CrossRef](#)]
19. Obregon, J.; Hong, J.; Jung, J.Y. Rule-based explanations based on ensemble machine learning for detecting sink mark defects in the injection moulding process. *J. Manuf. Syst.* **2021**, *60*, 392–405. [[CrossRef](#)]
20. Chen, J.C.; Guo, G.; Wang, W.N. Artificial neural network-based online defect detection system with in-mold temperature and pressure sensors for high precision injection molding. *Int. J. Adv. Manuf. Technol.* **2020**, *110*, 2023–2033. [[CrossRef](#)]
21. Mobley, R.K. *An Introduction to Predictive Maintenance*; Elsevier: Amsterdam, The Netherlands, 2002.
22. Cardoso, D.; Ferreira, L. Application of predictive maintenance concepts using artificial intelligence tools. *Appl. Sci.* **2020**, *11*, 18. [[CrossRef](#)]
23. Zhao, R.; Wang, D.; Yan, R.; Mao, K.; Shen, F.; Wang, J. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1539–1548. [[CrossRef](#)]
24. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufer, M.; Verstockt, S.; Van de Walle, R.; Van Hoecke, S. Convolutional neural network based fault detection for rotating machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [[CrossRef](#)]
25. Lee, W.J.; Wu, H.; Yun, H.; Kim, H.; Jun, M.B.; Sutherland, J.W. Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data. *Procedia Cirp.* **2019**, *80*, 506–511. [[CrossRef](#)]
26. Wang, B.; Lei, Y.; Yan, T.; Li, N.; Guo, L. Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. *Neurocomputing* **2020**, *379*, 117–129. [[CrossRef](#)]
27. Aslantaş, G.; Özsarac, M.; Rumelli, M.; Alaygut, T.; Bakirli, G.; Birant, D. Prediction of Remaining Useful Life for Plastic Injection Molding Machines Using Artificial Intelligence Methods. *J. Artif. Intell. Data Sci.* **2022**, *2*, 8–15.
28. Ceylan, U.; Yakup, G. Siamese Inception Time Network for Remaining Useful Life Estimation. *J. Artif. Intell. Data Sci.* **2021**, *1*, 165–175.
29. Ellefsen, A.L.; Ushakov, S.; Æsøy, V.; Zhang, H. Validation of data-driven labeling approaches using a novel deep network structure for remaining useful life predictions. *IEEE Access* **2019**, *7*, 71563–71575. [[CrossRef](#)]
30. Li, H.; Zhao, W.; Zhang, Y.; Zio, E. Remaining useful life prediction using multi-scale deep convolutional neural network. *Appl. Soft Comput.* **2020**, *89*, 106113. [[CrossRef](#)]
31. Chen, Z.; Wu, M.; Zhao, R.; Guretno, F.; Yan, R.; Li, X. Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Trans. Ind. Electron.* **2020**, *68*, 2521–2531. [[CrossRef](#)]
32. Kang, Z.; Catal, C.; Tekinerdogan, B. Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks. *Sensors* **2021**, *21*, 932. [[CrossRef](#)] [[PubMed](#)]

33. Ramasso, E.; Saxena, A. Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets. *Int. J. Progn. Health Manag.* **2014**, *5*, 1–15. [[CrossRef](#)]
34. Nozari, H.; Sadeghi, M.E. Artificial intelligence and Machine Learning for Real-world problems (A survey). *Int. J. Innov. Eng.* **2021**, *1*, 38–47.
35. Gunn, S.R. Support vector machines for classification and regression. In *ISIS Technical Report*; University of Southampton: Southampton, UK, 1998; Volume 14, pp. 5–16.
36. Liu, T.; Huang, L.; Chen, B. Real-time defect detection of laser additive manufacturing based on support vector machine. *J. Phys. Conf. Ser.* **2019**, *1213*, 052043. [[CrossRef](#)]
37. Huang, H.Z.; Wang, H.K.; Li, Y.F.; Zhang, L.; Liu, Z. Support vector machine based estimation of remaining useful life: Current research status and future trends. *J. Mech. Sci. Technol.* **2015**, *29*, 151–163. [[CrossRef](#)]
38. Kramer, O.; Kramer, O. K-nearest neighbors. In *Dimensionality Reduction with Unsupervised Nearest Neighbors*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 13–23.
39. Yıldız, K.; Buldu, A.; Demetgul, M. A thermal-based defect classification method in textile fabrics with K-nearest neighbor algorithm. *J. Ind. Text.* **2016**, *45*, 780–795. [[CrossRef](#)]
40. Khanzadeh, M.; Chowdhury, S.; Marufuzzaman, M.; Tschopp, M.A.; Bian, L. Porosity prediction: Supervised-learning of thermal history for direct laser deposition. *J. Manuf. Syst.* **2018**, *47*, 69–82. [[CrossRef](#)]
41. Ouadah, A.; Zemmouchi-Ghomari, L.; Salhi, N. Selecting an appropriate supervised machine learning algorithm for predictive maintenance. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 4277–4301. [[CrossRef](#)]
42. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, WA, USA, 4 August 2001; Volume 3, pp. 41–46.
43. Frank, E.; Trigg, L.; Holmes, G.; Witten, I.H. Naive Bayes for regression. *Mach. Learn.* **2000**, *41*, 5–25. [[CrossRef](#)]
44. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle swarm optimization: A comprehensive survey. *IEEE Access* **2022**, *10*, 10031–10061. [[CrossRef](#)]
45. Romli, I.; Pardamean, T.; Butsianto, S.; Wiyatno, T.N.; bin Mohamad, E. Naive bayes algorithm implementation based on particle swarm optimization in analyzing the defect product. *J. Phys. Conf. Ser.* **2021**, *1845*, 012020. [[CrossRef](#)]
46. Zhang, N.; Wu, L.; Yang, J.; Guan, Y. Naive bayes bearing fault diagnosis based on enhanced independence of data. *Sensors* **2018**, *18*, 463. [[CrossRef](#)]
47. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: New York, NY, USA, 2021.
48. Stoltzfus, J.C. Logistic regression: A brief primer. *Acad. Emerg. Med.* **2011**, *18*, 1099–1104. [[CrossRef](#)]
49. Jin, R.; Li, J.; Shi, J. Quality prediction and control in rolling processes using logistic regression. *Trans. NAMRI/SME* **2007**, *35*, 113–120.
50. Akhir, E.A.P.; Ayuni, N. Predictive analytics of machine failure using linear regression on KNIME platform. In Proceedings of the 2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR), Kumamoto, Japan, 23–25 July 2021; pp. 59–64.
51. Zhao, M.; Tang, B.; Tan, Q. Bearing remaining useful life estimation based on time–frequency representation and supervised dimensionality reduction. *Measurement* **2016**, *86*, 41–55. [[CrossRef](#)]
52. Uyanık, G.K.; Güler, N. A study on multiple linear regression analysis. *Procedia-Soc. Behav. Sci.* **2013**, *106*, 234–240. [[CrossRef](#)]
53. Kotsiantis, S.B. Decision trees: A recent overview. *Artif. Intell. Rev.* **2013**, *39*, 261–283. [[CrossRef](#)]
54. Ogorodnyk, O.; Lyngstad, O.V.; Larsen, M.; Wang, K.; Martinsen, K. Application of machine learning methods for prediction of parts quality in thermoplastics injection molding. In *International Workshop of Advanced Manufacturing and Automation*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 237–244.
55. Gerdes, M.; Galar, D.; Scholz, D. Genetic algorithms and decision trees for condition monitoring and prognosis of A320 aircraft air conditioning. *Insight-Non-Destr. Test. Cond. Monit.* **2017**, *59*, 424–433. [[CrossRef](#)]
56. Dietterich, T.G. Ensemble methods in machine learning. In Proceedings of the Multiple Classifier Systems: First International Workshop, MCS 2000, Cagliari, Italy, 21–23 June 2000; Proceedings 1; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
57. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
58. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
59. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurobot.* **2013**, *7*, 21. [[CrossRef](#)] [[PubMed](#)]
60. Takalo-Mattila, J.; Heiskanen, M.; Kyllönen, V.; Määttä, L.; Bogdanoff, A. Explainable Steel Quality Prediction System Based on Gradient Boosting Decision Trees. *IEEE Access* **2022**, *10*, 68099–68110. [[CrossRef](#)]
61. Ayvaz, S.; Alpay, K. Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Syst. Appl.* **2021**, *173*, 114598. [[CrossRef](#)]
62. Hao, X.; Zhang, G.; Ma, S. Deep learning. *Int. J. Semant. Comput.* **2016**, *10*, 417–439. [[CrossRef](#)]
63. Dey, L.; Meisheri, H.; Verma, I. Predictive Analytics with Structured and Unstructured data—a Deep Learning based Approach. *IEEE Intell. Inform. Bull.* **2017**, *18*, 27–34.
64. Taud, H.; Mas, J. Multilayer perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 451–455.

65. Bhatt, D.; Patel, C.; Talsania, H.; Patel, J.; Vaghela, R.; Pandya, S.; Modi, K.; Ghayvat, H. CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics* **2021**, *10*, 2470. [CrossRef]
66. Ribani, R.; Marengoni, M. A survey of transfer learning for convolutional neural networks. In Proceedings of the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), Rio de Janeiro, Brazil, 28–31 October 2019; pp. 47–57.
67. Shafiee, M.J.; Chywl, B.; Li, F.; Wong, A. Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *arXiv* **2017**, arXiv:1709.05943.
68. Sateesh Babu, G.; Zhao, P.; Li, X.L. Deep convolutional neural network based regression approach for estimation of remaining useful life. In Proceedings of the Database Systems for Advanced Applications: 21st International Conference, DASFAA 2016, Dallas, TX, USA, 16–19 April 2016; Proceedings, Part I 21; Springer: Berlin/Heidelberg, Germany, 2016; pp. 214–228.
69. Medsker, L.R.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 64–67.
70. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A tutorial into long short-term memory recurrent neural networks. *arXiv* **2019**, arXiv:1909.09586.
71. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.
72. Wu, Y.; Yuan, M.; Dong, S.; Lin, L.; Liu, Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* **2018**, *275*, 167–179. [CrossRef]
73. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
74. Liu, K.; Li, A.; Wen, X.; Chen, H.; Yang, P. Steel surface defect detection using GAN and one-class classifier. In Proceedings of the 2019 25th International Conference on Automation and Computing (ICAC), Lancaster, UK, 5–7 September 2019; pp. 1–6.
75. Verstraete, D.; Droguett, E.; Modarres, M. A deep adversarial approach based on multi-sensor fusion for semi-supervised remaining useful life prognostics. *Sensors* **2019**, *20*, 176. [CrossRef]
76. Li, X.; Zhang, W.; Ma, H.; Luo, Z.; Li, X. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. *Knowl.-Based Syst.* **2020**, *197*, 105843. [CrossRef]
77. Khan, S.A.; Prosvirin, A.E.; Kim, J.M. Towards bearing health prognosis using generative adversarial networks: Modeling bearing degradation. In Proceedings of the 2018 International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 20–22 February 2018; pp. 1–6.
78. Bank, D.; Koenigstein, N.; Giryas, R. Autoencoders. *arXiv* **2020**, arXiv:2003.05991.
79. Su, C.; Li, L.; Wen, Z. Remaining useful life prediction via a variational autoencoder and a time-window-based sequence neural network. *Qual. Reliab. Eng. Int.* **2020**, *36*, 1639–1656. [CrossRef]
80. Doersch, C. Tutorial on variational autoencoders. *arXiv* **2016**, arXiv:1606.05908.
81. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
82. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
83. Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; Sun, L. Transformers in time-series: A survey. *arXiv* **2022**, arXiv:2202.07125.
84. Zaheer, M.; Guruganesh, G.; Dubey, K.A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big bird: Transformers for longer sequences. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17283–17297.
85. Wang, J.; Xu, G.; Yan, F.; Wang, J.; Wang, Z. Defect transformer: An efficient hybrid transformer architecture for surface defect detection. *Measurement* **2023**, *211*, 112614. [CrossRef]
86. Shang, H.; Sun, C.; Liu, J.; Chen, X.; Yan, R. Defect-aware transformer network for intelligent visual surface defect detection. *Adv. Eng. Inform.* **2023**, *55*, 101882. [CrossRef]
87. Chen, D.; Hong, W.; Zhou, X. Transformer network for remaining useful life prediction of lithium-ion batteries. *IEEE Access* **2022**, *10*, 19621–19628. [CrossRef]
88. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
89. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A review of machine learning interpretability methods. *Entropy* **2020**, *23*, 18. [CrossRef] [PubMed]
90. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [CrossRef]
91. Vujović, Ž. Classification model evaluation metrics. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 599–606. [CrossRef]
92. Polenta, A.; Tomassini, S.; Falconelli, N.; Contardo, P.; Dragoni, A.F.; Sernani, P. A Comparison of Machine Learning Techniques for the Quality Classification of Molded Products. *Information* **2022**, *13*, 272. [CrossRef]
93. Microsoft. Predictive Maintenance Modelling Guide. 2018. Available online: https://github.com/ashishpatel26/Predictive_Maintenance_using_Machine_Learning_Microsoft_Casestudy/tree/master/data (accessed on 23 March 2023).
94. Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature selection: A data perspective. *ACM Comput. Surv. (CSUR)* **2017**, *50*, 1–45. [CrossRef]
95. Patro, S.; Sahu, K.K. Normalization: A pre-processing stage. *arXiv* **2015**, arXiv:1503.06462.

96. Gogtay, N.J.; Thatte, U.M. Principles of correlation analysis. *J. Assoc. Phys. India* **2017**, *65*, 78–81.
97. Li, E.; Zeng, L.; Zhou, Z.; Chen, X. Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 447–457. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.