

Article

Adaptable and Explainable Predictive Maintenance: Semi-Supervised Deep Learning for Anomaly Detection and Diagnosis in Press Machine Data

Oscar Serradilla ^{1,*}, Ekhi Zugasti ¹, Julian Ramirez de Okariz ², Jon Rodriguez ² and Urko Zurutuza ¹  

¹ Electronics and Computing Department, Mondragon Unibertsitatea, Loramendi 4, 20500 Mondragon, Spain; ezugasti@mondragon.edu (E.Z.); uzurutuza@mondragon.edu (U.Z.)

² Koniker, San Andres Auzoa 20, 20500 Mondragon, Spain; j.rokariz@koniker.coop (J.R.d.O); j.rodriguez@koniker.coop (J.R.)

* Correspondence: oserradilla@mondragon.edu

Featured Application: Deep learning-based predictive maintenance on press machine production data, addressing adaptability, novelty detection, and diagnosis requirements. It has applications on industrial use cases requiring semi-supervised predictive maintenance to detect and diagnose novel failure types.



Citation: Serradilla, O.; Zugasti, E.; Ramirez de Okariz, J.; Rodriguez, J.; Zurutuza, U. Adaptable and Explainable Predictive Maintenance: Semi-Supervised Deep Learning for Anomaly Detection and Diagnosis in Press Machine Data. *Appl. Sci.* **2021**, *11*, 7376. <https://doi.org/10.3390/app11167376>

Academic Editors: Carsten Felden, Nicola Castellano, Henning Baars, Bruno Maria Franceschetti, Michiyasu Nakajima, Fanny-Eve Bordeleau and Trinck Sebastian

Received: 30 June 2021

Accepted: 9 August 2021

Published: 11 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

Abstract: Predictive maintenance (PdM) has the potential to reduce industrial costs by anticipating failures and extending the work life of components. Nowadays, factories are monitoring their assets and most collected data belong to correct working conditions. Thereby, semi-supervised data-driven models are relevant to enable PdM application by learning from assets' data. However, their main challenges for application in industry are achieving high accuracy on anomaly detection, diagnosis of novel failures, and adaptability to changing environmental and operational conditions (EOC). This article aims to tackle these challenges, experimenting with algorithms in press machine data of a production line. Initially, state-of-the-art and classic data-driven anomaly detection model performance is compared, including 2D autoencoder, null-space, principal component analysis (PCA), one-class support vector machines (OC-SVM), and extreme learning machine (ELM) algorithms. Then, diagnosis tools are developed supported on autoencoder's latent space feature vector, including clustering and projection algorithms to cluster data of synthetic failure types semi-supervised. In addition, explainable artificial intelligence techniques have enabled to track the autoencoder's loss with input data to detect anomalous signals. Finally, transfer learning is applied to adapt autoencoders to changing EOC data of the same process. The data-driven techniques used in this work can be adapted to address other industrial use cases, helping stakeholders gain trust and thus promote the adoption of data-driven PdM systems in smart factories.

Keywords: fault detection; diagnosis; predictive maintenance; deep learning; autoencoder; explainable artificial intelligence; transfer learning; semi-supervised; press machine; industry 4.0

1. Introduction

Press machines are rotating machines that form input material by cuts and deformations, and they represent one of the biggest type of machine tools. They play an important role in manufacturing processes, being a key component on stamping production lines of metal-formed components. Maintenance is critical to ensure a right machine working condition, consistent manufacturing quality, prevent downtime, and reduce costs [1].

The emergence of industry 4.0 has facilitated industrial companies to monitor their machines and assets based on cyber-physical systems that collect and upload data to the cloud. This ever-increasing big data collection can promote industrial analytical tools that provide additional information on asset health, a step forward towards smart factories.

Besides, research on deep learning applications has grown in recent years, achieving state-of-the art results in many field including industrial applications [2].

Nowadays, many factories either use predetermined maintenance, where the components are replaced periodically without observing their degradation after a certain number of cycles or execution time, or corrective maintenance, which replaces the broken parts of an asset when it stops working or when its users detect defects. In contrast, predictive maintenance monitors the condition of a machine/component to detect early signs of failure, diagnose its root cause, and predict its evolution until failure, enabling the maintenance application when assets need it by anticipating maintenance requirements. The PdM strategy extends components' working life with respect to predetermined maintenance, and avoids damages by performing interventions before failures occur, before corrective maintenance [3]. Therefore, companies adopting PdM can achieve an overall equipment effectiveness of 90% [4] and it can achieve a 10 times return on investment [5].

The work on predictive maintenance by Jimenez-Cortadi et al. [6] argues that data processing for predictive maintenance is an essential step. In addition, Welz [7] defines PdM stages as 1st anomaly detection, 2nd diagnosis, 3rd prognosis, and last mitigation.

Each industrial use case has its own maintenance requirements and characteristics, usually differing from one another and thus requiring algorithm adaptation. Industrial companies work hard to ensure their machines are working correctly, so the majority of collected data belongs to correct working conditions. There are also issues with the registration of failures: sometimes this process does not exist, but when it does, inconsistent labeling may be used. Additionally, forcing assets to have failures is expensive, and many times it is impossible to replicate desired failure types. These are the reasons why research interest on semi-supervised models for PdM is increasing.

The review by Venkatasubramanian et al. [8] gathers the main characteristics of an anomaly detection and fault diagnosis system to address industrial process challenges. According to the review, the system should be able to detect and diagnose the anomalies quickly to facilitate fast intervention and mitigation. Besides, real-time computation and storage handling may be required, prioritizing the minimization of modeling efforts. The system should also have a trade-off between robustness and performance. The system should model the normal behaviour of the process and be able to identify novel abnormal operations. In addition, the system should distinguish different failure types. Moreover, the system should adapt to the changing process operating conditions and environmental conditions to ensure it continues working correctly. Finally, the model should provide explanations about the root cause of the anomaly so that operators can evaluate it and act based on their expertise.

Serradilla et al. state in [9] that most common deep learning-based semi-supervised anomaly detection works for PdM are based on autoencoders [10], variational autoencoders [11], generative adversarial networks [12], or recurrent neural networks [13,14]. In addition, semi-supervised diagnosis for PdM is usually addressed by self-organizing map (SOM) [15] and *t*-distributed stochastic neighbor embedding (*t*-SNE) [16] models. Furthermore, transfer learning techniques on deep learning models enable reusability and adaptability to changes in environmental and operational conditions (EOC). Several works including transfer learning techniques are Wen et al. [17] with application on motor vibration anomaly detection, Wen et al. [18] implementing univariate anomaly detection inspired in U-net, and Martinez et al. [19] who combine Bayesian search and convolutional neural networks for anomaly detection.

There exist many of state-of-the-art data-driven techniques that address PdM diagnosis based on classification, using samples of the target failure class for learning. However, diagnosis on semi-supervised models is much harder than on supervised classification models given that target failure data are not available for model training. In a semi-supervised scenario, anomaly detection requires modeling normality of correct data on a training dataset to detect anomalous samples with respect to learned normality, whereas

diagnosis should group anomalous data instances by similarity to discover failure types, and then give insight on possible causes for each failure type to facilitate maintenance.

Three related works that address PdM diagnosis in a semi-supervised way are reviewed in this paragraph. Zhang et al. [11] use a variational autoencoder to detect anomalies on a bearings dataset of an experimental setup and diagnose the anomalies using a discriminator architecture. The second work is by Zope et al. [20], who use different state-of-the-art and classical data-driven algorithms including deep learning to detect anomalies on an interacting quadruple tank system, and they use several diagnosis techniques according to the used model. Another related work is by Brito et al. [21], who use different XAI techniques for unsupervised fault detection and diagnosis in rotating machinery, applying machine learning models that are not deep learning models.

Research interest on transfer learning in the field of PdM is increasing given that it enables adapting trained models to data changes efficiently. Lee et al. in [22] use transfer learning to adapt a pre-trained 2D deep learning model to acceleration data of a rotating machine for PdM, reducing training resources. Maschler et al. apply transfer learning to transfer knowledge between models learned with different data subsets of turbofan simulation dataset [23]. In addition, Maschler et al. in [24] use transfer learning for incremental or sequential learning applied to industrial image recognition. They argue that transfer learning is suitable to solve the challenges of industrial real-life scenarios.

The objective of this work is to obtain a semi-supervised anomaly detection model trained with correct condition machine data that identifies novel anomalies, where the most accurate model among different data-driven models including deep learning is selected. Afterwards, the diagnosis of these novel anomalies should be facilitated with support of the selected model, assessing anomaly severity, grouping data by failure similarity, and highlighting anomalies in data signals to help domain technicians identify their root cause. Finally, the model should be reusable and adaptable to data variability, and it should have the capability of real-time execution.

This work's main contributions are described in this paragraph. First, it addresses PdM on an industrial process in a semi-supervised way, demonstrating the capabilities of deep learning models to address industrial requirements, and contributing to the press machine maintenance field. Moreover, this work contributes to the little researched semi-supervised diagnosis by applying state-of-the-art techniques of clustering, projection, competitive learning, and explainable artificial intelligence in an accurate deep learning-based anomaly detection model, which has been guided and validated by domain expertise. Finally, this model's adaptability to changes in EOCs have been tested and addressed using transfer learning, contributing to the field of transfer learning for PdM with process sensor data. This way, the model reusability and adaptability possibilities provided by transfer learning are demonstrated, reducing the training resources required on an industrial use case.

2. Materials and Methods

2.1. Process Data

The experimentation dataset used in this work has been collected from a press machine of a stamping production line, facilitated by a private company. Press machines are machine tools that create metal-formed components by applying pressure, forming input material with cuts and deformations. Figure 1 shows an image of press machine and its main components.

This use case's press machine has a conventional electrical drive, composed by an electric engine connected to a flywheel and clutch-brake mechanism, and crank and connecting rod system that turns cinematic chain rotary movement into slide linear movement. The slide moves perpendicular to the floor, going from the top dead center (0°) to bottom dead center (180°), and then back to top dead center (360°), to complete the stamping cycle of 360° . The slide forms input material by stamping in the die, which is a two-part element specifically designed to address the requirements of finished blank presented in right part of Figure 1.

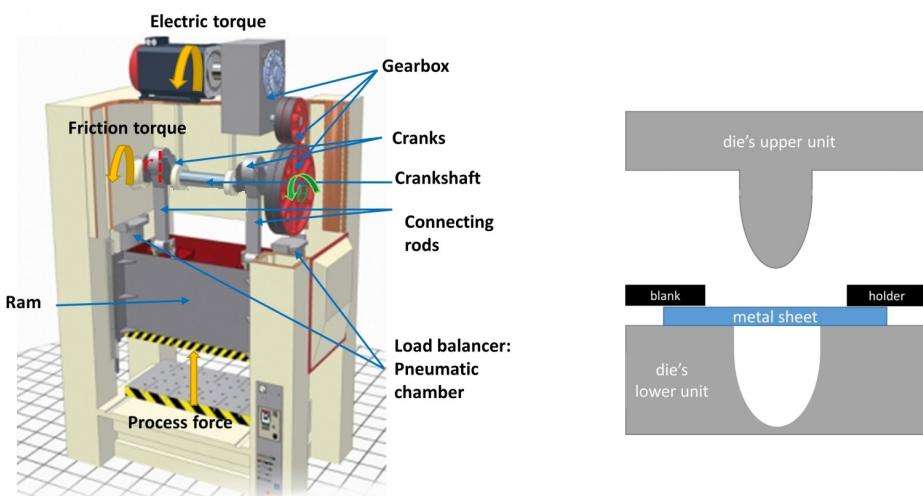


Figure 1. Image of a servo press machine with its main components on the left, and slide and ram scheme on the right [25].

The press machine used in this work is a transfer press. This contains working stations that perform different forming processes simultaneously with the same stroke. The input material is formed in all working stations, and there is a transfer element that advances components from one working station to the next one in the time frame between strokes. Once the component is formed in all the working stations, it exits the process.

The data were collected by sensors for each stroke, containing both single measure and evolution variables. Single measures like the die identifier or stroke number contain just one value that gives information about the stroke. Moreover, evolution variables refer to the data collected from sensors during each stroke in time-series format. The dataset is composed by the following variables: 2 speed sensors, 2 power consumption sensors, 6 force sensors, and 2 position sensors. These sensors are generic and have been integrated by the manufacturing company. A PLC collects the sensor data, synchronizes it and uploads it to the cloud, providing access to the synchronized data through a web API. Due to an NDA, no further details about the sensors/sensor placement or the API can be provided.

With the objective of reducing industrial data variability to facilitate data analysis, data was grouped by similar environmental and operational conditions. This grouping was achieved by filtering data by stroke identifier and workorders.

2.2. Experimenting Procedure

The first task of this PdM work was creating an anomaly detection model only trained with correct machine data. This model had to provide a damage index for each stroke sample, which was used to classify as anomalous strokes when their value surpassed an anomaly threshold, or correct when they did not. Moreover, the damage index could be used to estimate anomalous strokes' severity: when their distance to the anomaly threshold is higher, the magnitude of the anomaly should also be higher.

Afterwards, the selected anomaly detection model was used for diagnosis, and finally, its adaptation to data of different workorders and different dies was analyzed. All the experiments were designed to address industrial use case's requirements and validated with domain technicians. The anomaly detection and diagnosis experiments have been performed with data of the same die ID and workorder. To test model adaptability, two experiments have been performed: testing with data of the same die ID and different workorders, and testing with data of the same workorder but different die.

To analyze the results from domain perspective, 4 synthetic failures were designed by modifying correct stroke data. Moreover, different versions of each failure were created by changing their magnitude. Synthetic failure types and their variations are presented in Table 1.

Table 1. Synthetic failures and corresponding signal modifications used for model validation.

Synthetic Failure	Signal Modification and Variations
Force increment, simulating harder input material	Increase forces on full signal 4 axis (+10%, +25%, +50% and +100%)
Die misadjustment, simulating press force anticipation	Anticipation of press force application (only 50° and 100°)
Machine degradation, higher power consumption for same forces	Increase power consumption on full signal (+10%, +25%, +50% and +100%)
Unbalanced loads	Exchange press input and press output forces on full signal

The methodology to create synthetic failures is described in this paragraph. As described in Table 1, there are four synthetic failure types that can have different signal modifications. For each failure type and modification magnitude, all strokes of validation and test data splits have been used to create the corresponding failure validation and failure test sets deterministically. As a result, 11 sets of failure types having the same number of strokes as original splits have been created.

The first failure type simulates that input material is harder than the process is designed for, so technicians should revise the machine and make adjustments to adapt it for new material's characteristics; this way, machine breakage may be prevented. There are two main process causes that can derive into this failure type. The first cause is using input materials of inadequate characteristics, such as having greater thickness, greater hardness, or greater mechanical resistance. The second cause is a wrong parameterization of the die's height. When its height is lower than required for the process' characteristics, the machine has to perform additional efforts that increase the damage risk on its structure and transmission components. For reference, a misadjustment of 1mm requires a force increase of 10–20%.

The second failure type simulates die's misplacement, which can cause defects in the formed components in addition to damaging the machine. This failure type arises when the press' ram initiates the forming process in a height higher than it was designed for, by the use of chocks or elevation cylinders. This configuration derives into the anticipation of forces, which increases the machine's working energy unnecessarily. Accordingly, the damage risk increases and early degradation on the machine's components may arise. The third synthetic failure increases power consumption while keeping original press forces to simulate a degradation failure in the press transmission system, requiring more power than a healthy machine to perform the same process. This failure indicates anomalies in the machine's cinematic chain, including gears' wear or breaking, bushings' seizure, and clutch's capacity reduction.

The final synthetic failure simulates that load distribution has changed, which could be caused by bad machine configuration and may result in a premature degradation of its components. The forming processes applied on input working stations are usually related to bending, which require higher forces than folding and cutting that are performed in more advanced working stations. The configuration of each working station is independent, and therefore the forces that the press machine applies are the result of adding all the forces applied along its working stations. A wrong configuration in the work stations near the machine's output can increase the load applied in this part of the machine significantly, thus causing the misalignment of forces by performing higher loads in the output side than in the input side. The machine is not designed to work correctly in this situation and therefore, its components can be damaged. Another possible application of this synthetic failure can be detecting cracks and misalignments in the input connecting rods. When the applied forces are higher in the output working stations than in the input ones, the input connecting rods may be unable to support the applied forces.

Two original stroke signals and four failure signals, one failure for each failure class, are presented in Figure 2. In addition, different magnitudes of these failure types have the same shape but vary according to specifications of Table 1.

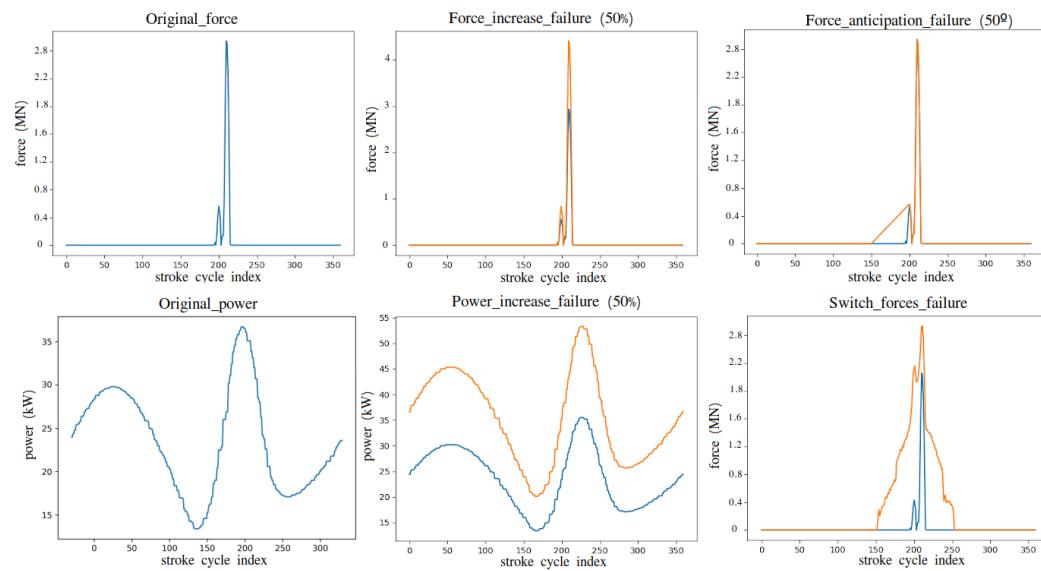


Figure 2. Two correct signals on the left, and the four synthetic failures on the right. Blue signal represents normal stroke data and orange signal represents one corresponding synthetic failure signal.

For data preparation, data were first grouped by die ID and workorders, and used these data subsets for experimenting. Each workorder contains process data collected in a specific continuous time interval no longer than a week, which is delimited by production working periods. This grouping system is suitable for press machines given that each die works under similar operating conditions and performs the same forming process.

Then, data were analyzed divided for model creation and validation based on industrial production data order: first 80% strokes were used for training and validation, which was further divided into 90% for training and 10% for validation, whereas last 20% was used for testing. The resulting number of stroke samples used for training, validation and testing in anomaly detection are 516, 58, and 143, respectively. Similarly, the number of stroke samples used for training, validation and testing for adaptability are 495, 55, and 137, as summarized in Table 2. The training data split contains strokes collected in a time period of 12 h, but this number can vary regarding the number of strokes and time period of each workorder.

Afterwards, data were cleaned and preprocessed to prepare it for the PdM models, and then compared their results. For data cleaning, domain technicians defined filters using plots, correlations, and univariate statistical measures such as mean, in order to discard any anomalous stroke resulted from acquisition problems or missing data. Models were trained under the assumption that 90% of training data was correct and the remaining could have outliers. This assumption was concluded after domain technicians analyzed training strokes using domain-based ratios and expertise to quantify the probability of outliers in the data.

Next, a model evaluation and ranking strategy was defined, which uses the F1 metric for overall performance analysis, and precision and recall for further analysis. These metrics were selected because they evaluate errors based on target failure class and work well with imbalanced datasets.

Table 2. Number of strokes available for training, validation and testing in anomaly detection and adaptability experiments. The indication “for each type” refers to each of the synthetic failure types and their magnitude combinations, which consists of 11 types.

Dataset	Data Type	Strokes in Train Set	Strokes in Validation Set	Strokes in Test Set for Each Type
For anomaly detection	Correct	516	58	143
For anomaly detection	Synthetic failure	0	55 (5 for each type)	143
For transfer learning	Correct	495	55	137
For transfer learning	Synthetic failure	0	55 (5 for each type)	137

Validation was complemented with a questionnaire to be filled by domain technicians, which contained plots of 14 selected strokes among normal and synthetic failure data. Its objective was to evaluate the technicians’ capability to differentiate synthetic anomalies by visual comparison with plots of correct strokes of the same signal. The survey was answered by 9 domain technicians with high expertise on the field. Figure 3 shows an example question of stroke data belonging to the form.

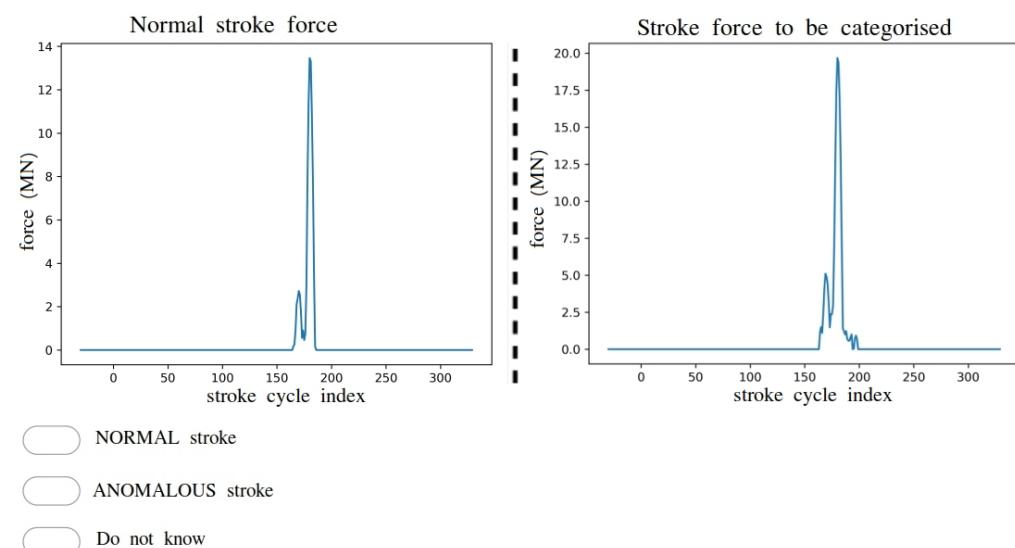


Figure 3. A sample of stroke question in the questionnaire for domain technicians.

2.3. Predictive Maintenance Techniques

The algorithms evaluated for anomaly detection are principal component analysis (PCA) [26], extreme learning machine (ELM) [27], one-class support vector machines (OC-SVM) [28], an autoencoder based on 2-dimensional convolutional neural networks (2D-CNN-AE) [29], and null space [30]. The parameters for training all mentioned models are contained in Table 3, whereas the architecture of 2D-CNN-AE and its details are presented in Figure 4.

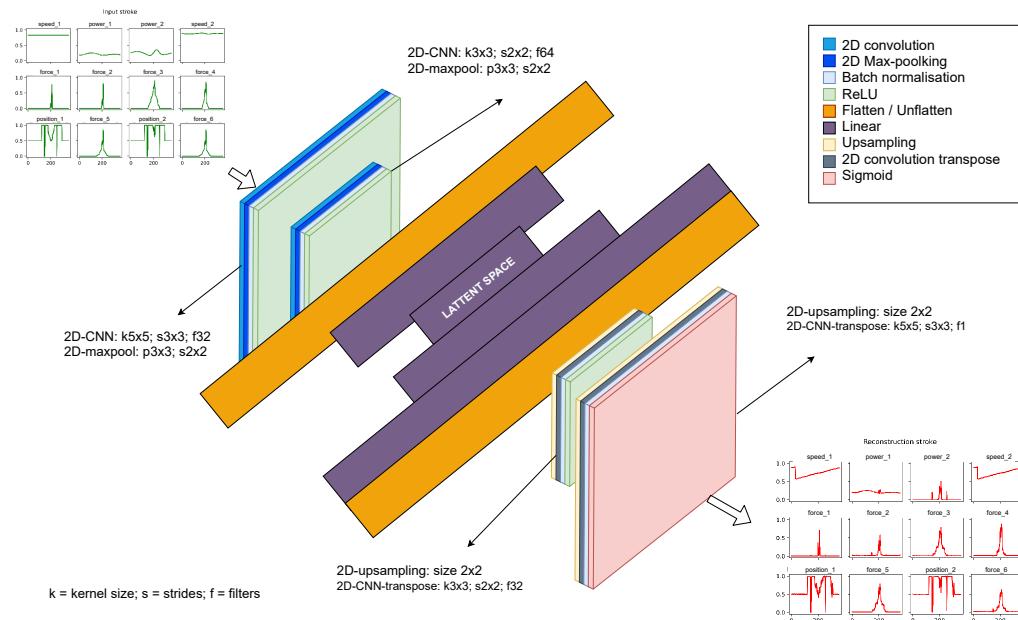


Figure 4. Architecture of 2D-CNN-AE and its parameters.

Table 3. Parameters for training anomaly detection models. Cycle and trad. feats refer to the way anomaly detection models prepare and process the data.

Anomaly Detection Model	Training Parameters
PCA cycle	num. components = 90%; loss func. = RMSE
PCA trad. feats	num. components = 90%; loss func. = RMSE
ELM cycle	$h_dim = 10$; target = 1; loss func. = absolute diff.
ELM trad. feats	$h_dim = 10$; target = 1; loss func. = absolute diff.
OC-SVM cycle	kernel = rbf; kernel coef. = $1/n_features$; nu = 1%
OC-SVM trad. feats	kernel = rbf; kernel coef. = $1/n_features$; nu = 1%
Null-space	alpha = 5; beta = 5 Input dim: batchxnum_cyclesx12x1
2D-CNN-AE	2 FE layers of 2D CNN&maxpool&relu&flatten&FFNN optimizer = adam (lr = 0.001, betas = (0.9, 0.999), eps = 1×10^{-8}) early stopping, patience = 2 train loss = MSE; loss func = RMSE

The PCA, ELM, and OC-SVM algorithms have been executed in two ways. The first was training and performing predictions for strokes' evolution variables, setting a threshold on the loss using percentile, and then counting the number of cycle samples surpassing the threshold and setting another threshold on this number to determine if that stroke sample is anomalous or not. The other way was to extract statistical features for each sensor variable and then perform anomaly detection in these features with a percentile threshold. The extracted features were mean, variance, maximum, and minimum values. Null-space and 2D-CNN-AE use evolution variables as input, not requiring specific feature extraction. The input of 2D-CNN-AE has been formatted to batchxnum_cyclesx12x1, where batch refers to a group of strokes, num_cycles are the observations of each input variable, 12 are the input variables, and the last dimension indicates that each stroke has one dimension and it is required by the library to perform 2D convolutions.

For diagnosis by isolating failures on latent space features, t-SNE [31] was used for visualization of latent space variables; ordering points to identify the clustering structure (OPTICS) [32] algorithm as a density-based clustering algorithm; gaussian mixture models (GMM) [33] as parametric clustering algorithm; and SOM [34] to project data in a new space based on competitive learning. Based on a literature review of clustering algorithms, these are the reasons for selecting each algorithm. GMM enables the parametrization with

number of clusters, which is not required for anomaly detection but is a requirement for diagnosis. With the objective to enable a less strict clustering, OPTICS has been selected to cluster the space based on density, which has the additional advantage of grouping outlier instances in another group. Finally, SOM does not require specification of parameters that limit the number of clusters, so it projects data to a new space where clusters are formed naturally.

Explainable artificial intelligence (XAI) techniques enabled to track the loss throughout the 2D-CNN-AE model and link it with variables. Its objective was to analyze which input variables and concretely which cycle points were causing the anomaly. For this purpose, the shapley additive explanations (SHAP) [35] library was used, explaining predictions by stroke. Concretely, GradientExplainer was the selected method, which according to its documentation [36], gradient explainer combines Integrated Gradients [37], SHAP, and SmoothGrad [38] into a single expected value equation.

Finally, model adaptability to same die data in new workorder, and to other die data have been tested. First, the original model and threshold were tested, and then experimented transfer learning by freezing convolutional layers and only retraining inner linear layers. This procedure should be enough to adapt the model for data of similar EOC, where convolutional layers are used as feature extractors and retraining linear layers adjusts the model to data variations.

All the experiments and data processing were executed in python version 3.7.9, using the following libraries: tensorflow 2.3.1 [39], scikit-learn 0.23.2 [40], MiniSom 2.2.9 [41] and SHAP 0.39.0 [36]. All the hyperparameters used for algorithm execution have been specified, and when any is missing, it takes the default parameters of the library for the specified version.

3. Experimenting Results

This section describes and interprets the experimental results of this work. It is split into three subsections to facilitate their presentation and interpretation. Section 3.1 compares state-of-the-art and classical semi-supervised data-driven models for PdM anomaly detection on a subset of press machine data. Section 3.2 uses the best performing anomaly detection algorithm to implement diagnosis based on root cause analysis, which is supported on clustering, projection, competitive learning and XAI techniques. Finally, Section 3.3 analyses the selected anomaly detection model's ability to adapt to different EOCs by testing with different die and workorder datasets, supported on transfer learning.

3.1. Anomaly Detection

The first PdM stage is creating an anomaly detection system capable of distinguishing normal and anomalous working conditions on monitored assets. For this task, a data subset collected in the same workorder was selected, and additionally belonged to the same die. This step selected data strokes that shared EOC, thus reducing data variability to facilitate model creation and validation.

Then, given monitored variables were in different scales, data was normalized or standardized, depending on the algorithm. Variables were standardized to have a mean of 0 and standard deviation of 1 before inputting data to PCA to maximize variance, whereas variables were normalized to the range of [0, 1], bringing them to the same range while keeping dispersion before inputting to the rest of algorithms.

Then, the algorithms presented in Section 2.3 were executed under the assumption that at least 90% of monitored strokes belonged to correct working condition, which resulted in selecting the parameters of Table 3. The results of these algorithms using F1-score are collected in Table 4.

Table 4. F1 score of anomaly detection models trained under the assumption that at least 90% of training data is correct. CNN-AE's p90 and p95 refer to the threshold used for anomaly detection, indicating percentile 90 and percentile 95 of correct validation data, respectively.

Algorithm	(1) Force Increase				(2) Force Anticipation		(3) Power Increase				Switch Forces
	10%	25%	50%	100%	120–170°	70–170°	10%	25%	50%	100%	
PCA cycle	0.27	0.74	0.97	0.97	0.09	0.93	0.10	0.16	0.65	0.97	0.97
PCA feats	0.67	0.67	0.67	0.67	0.67	0.67	0.02	0.09	0.62	0.67	0.67
ELM cycle	0.17	0.21	0.24	0.25	0.18	0.21	0.15	0.15	0.15	0.16	0.23
ELM feats	0.07	0.22	0.68	0.76	0.23	0.52	0.06	0.06	0.07	0.07	0.91
OC-SVM cycle	0.67	0.68	0.68	0.68	0.66	0.66	0.66	0.66	0.66	0.66	0.68
OC-SVM feats	0.41	0.67	0.67	0.67	0.47	0.67	0.08	0.06	0.04	0.02	0.67
Null space	0.82	1.00	1.00	1.00	1.00	1.00	0.47	1.00	1.00	1.00	1.00
CNN-AE p90	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
CNN-AE p95	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

The results of Table 4 show that null-space and 2D-CNN-autoencoder are the algorithms that work better than the rest on average, but autoencoder is even capable of detecting the smallest versions of failures. After analyzing its precision and recall, several correct strokes were being classified as failure, so the threshold could be better adjusted. For this task, a search for the best percentile threshold on correct validation data from 1 to 100 was performed, selecting the one that obtained the best F1 score using synthetic failures of validation data. The best threshold found for 2D-CNN-AE was by using percentile 95, achieving a F1 score of 0.99 for each failure type and thus outperforming the rest algorithms. Similarly, null-space algorithm achieved an average F1 score of 0.92, which is lower than autoencoder but has the advantage of not requiring failure data for threshold selection. Both algorithms also provide a damage index that takes higher values when they are fed with more higher magnitude synthetic failures.

The analysis of domain technician's survey results showed that they were unable to detect the smallest failures with variations of 10% or 50° failure anticipation, but they precisely detected more notorious variations like 50%, 100%, 100° anticipation and switch forces. The average F1 score of the survey is 0.82. This analysis validates algorithms' results also from domain perspective, given null-space and 2D-CNN-AE obtain on average a F1 score of and 0.92 and 0.99, respectively, in comparison with the average 0.82 F1 score of the questionnaire that gathers domain expertise results.

At this point, the online data processing capability of 2D-CNN-AE anomaly detection model has been evaluated in a Nvidia 2080Ti graphics processing unit, which was measured by the mean time required calculate 10 stroke data. The autoencoder model was used to make predictions of 50 stroke chunks, each containing 10 stroke data, and the average elapsed time was 7 milliseconds. In addition, null space algorithm requires 20 milliseconds to process each stroke sample. This performance test has validated model's real-time data processing capabilities.

3.2. Diagnosis

After performing anomaly detection, isolating different failure types and diagnosing their root cause is the next stage of predictive maintenance. For this stage, clustering, visualization, projection, and XAI techniques were used in the 2D-CNN-AE anomaly detection model of previous section given this achieved the best results overall.

The first experiment required forward passing through the encoder of original test data and synthetic failure data generated from test data, which turned each stroke into a 32-dimensional feature vector. Its objective was to test the ability to differentiate failure types in this compressed space. Initially, a 2-component t-SNE to feature vectors was applied with the objective to create a 2D space where clusters are visualized. The result of this experiment with 32-dimensional feature vector where disperse clusters given that the space might be too big or noisy for t-SNE to find clear relations. Therefore, experimenting with dimensionality reduction techniques was performed while keeping data variability to facilitate t-SNE.

Similar to the original *t*-SNE work [31], PCA has been used for dimensionality reduction to speed up computation and suppress noise without distorting pairwise distance, with the objective to facilitate *t*-SNE's work and analyze whether clearer clusters are created. PCA requires standardization of features, and 6 components were selected to keep 95% of initial variability. After PCA, these features were inputted to *t*-SNE that was configured with 2 components, 10 learning rate, 10,000 maximum iterations, and remaining parameters as default for sklearn. A grid search on perplexity was performed, which is a parameter that represents number of nearest neighbors in manifold learning algorithms, so values from where 5–70 range in strides of 5, and range 70–200 in strides of 10 were tested.

The results showed three groups of data where differentiation between increase forces and increase power consumption failures was difficult given that failures with smallest data variation (10%) were very near in the new space. The experiment was repeated, and then performed clustering using gaussian mixture models (GMM) with 4 number of clusters, with the objective to isolate the 4 failure types in *t*-SNE's embedded space of two dimensions. Low values of perplexity increased data sparsity in the new space, whereas high values increased cluster compactness. However, beyond a certain perplexity value, data distribution did not change much and clusters remained stable, so this point was selected for final *t*-SNE results analysis. This point was achieved with perplexity of 145, which is shown in Figure 5: it contains real failure labels on the left, whereas results of GMM with 4 number of clusters are presented in the right.

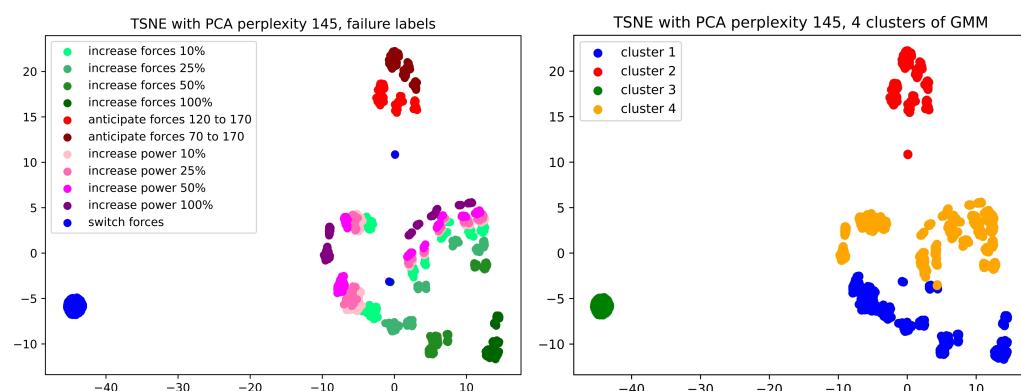


Figure 5. 2 *t*-SNE space images, containing failure labels in the left part, and GMM clustering labels in the right part.

This *t*-SNE diagnosis has clearly differentiated the four clusters, except for several increase forces (10%, 25%, and 50%) strokes that were assigned to power consumption failure. Moreover, several increases in power consumption (10%, 25%, and 50%) were assigned to the increase forces failure cluster. The smallest magnitudes of these two failure classes are still too close to the healthy data for the clustering algorithm to differentiate between them. This technique has the advantage of enabling results visualization. However, given its difficulty for hyperparameter tuning without the information of number of clusters and failure labels, it can be hard to implement with semi-supervised models. Another disadvantage of *t*-SNE is its possibility to create non-existent patterns on data given its adaptation to it.

To complement *t*-SNE experiments, another clustering algorithm was applied to latent space data after PCA, aiming at creating clusters that separate different failure types automatically. The OPTICS density clustering algorithm was selected for this task, and the minimum number of samples parameter from 20 to 140 was grid searched to find the configuration where the algorithm detected 4 clusters. The min num samples parameter that created 4 clusters is 80, and its results are presented in Table 5. Moreover, the algorithm creates a group for outlier samples, which do not belong to any discovered clusters according to their distance in the feature space.

Table 5. Clustering results using OPTICS algorithm configured with the hyperparameter of minimum number of samples equal to 80, evaluated with precision (prec) and recall (rec) metrics.

Number Clusters	(3) Power Increase		(2) Force Anticipation		(1) Force Increase		(4) Switch Forces	
	prec	rec	prec	rec	prec	rec	prec	rec
outliers	0.41	0.74	0.20	0.72	0.40	0.72	0.00	0.00
1	0.75	0.26	0.00	0.00	0.25	0.09	0.00	0.00
2	0.00	0.00	1.00	0.28	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	1.00	0.19	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00

According to Table 5, the failure switch forces is correctly isolated given the cluster 4 has a precision and recall of 1 for this failure type. Cluster 3 contains only instances of force increase failure, but not all of them are gathered given the recall is lower than 1; the remaining instances are assigned several to cluster 1 and the rest to the outliers group. Similarly, cluster 2 contains only force anticipation failure data, and the other instances are assigned to outliers group. In addition, cluster 1 contains mainly power increase failure data, but it also contains some force increase instances, being an overlap of two failures; the remaining power increase instances are assigned to the outliers group. Finally, the outliers group gathers an important number of instances belonging to force increase, force anticipation, and power increase failure types, but none of switch forces failure type.

All in all, there is a small overlap between force increase and power increase failure types. In addition, many force increase, force anticipation, and power increase instances are assigned to outliers group. At this point, the results of clustering with lower min samples was analyzed, which generates more clusters. This showed that lower magnitude versions of each failure type had higher probability to be assigned together than higher magnitude ones, which were correctly separated.

To continue diagnosis analysis, correct stroke data and the versions of highest magnitude for each failure type were projected to latent space using 2D-CNN-AE: 100% force increase, 100° force anticipation, 100% power increase and switch forces. Projected feature vectors were afterwards z-scaled, and normalized in [0,1] range to have all variables in a comparable distribution and range. These data were used to fit a SOM, projecting 32 input neurons to a 20×20 feature space map, and configuring its hyperparameters as sigma = 5, learning_rate = 0.5, neighborhood function = bubble and random training of 6000. This map projects similar instances on the original space to neurons that are near in the new space, which are represented in light colors. It also projects different instances to different groups of neurons in the new space, being separated with high distance neurons that are represented with dark colors.

The result of SOM is exhibited in Figure 6, which shows a clear separation among all failure types, where instances of the same class belong to near and light color neurons, and at the same time are separated with instances of other classes with high distance dark color neurons. In addition, normal data are separated from failures by high distance dark neurons, but at the same time, being in the middle of the map is reasonable as it is the root data used to create all failure data. Few outlier samples are located in big distance SOM cells; these belong to outlier strokes that were previously identified as outliers given their damage indexes were much higher than the majority of correct strokes'.

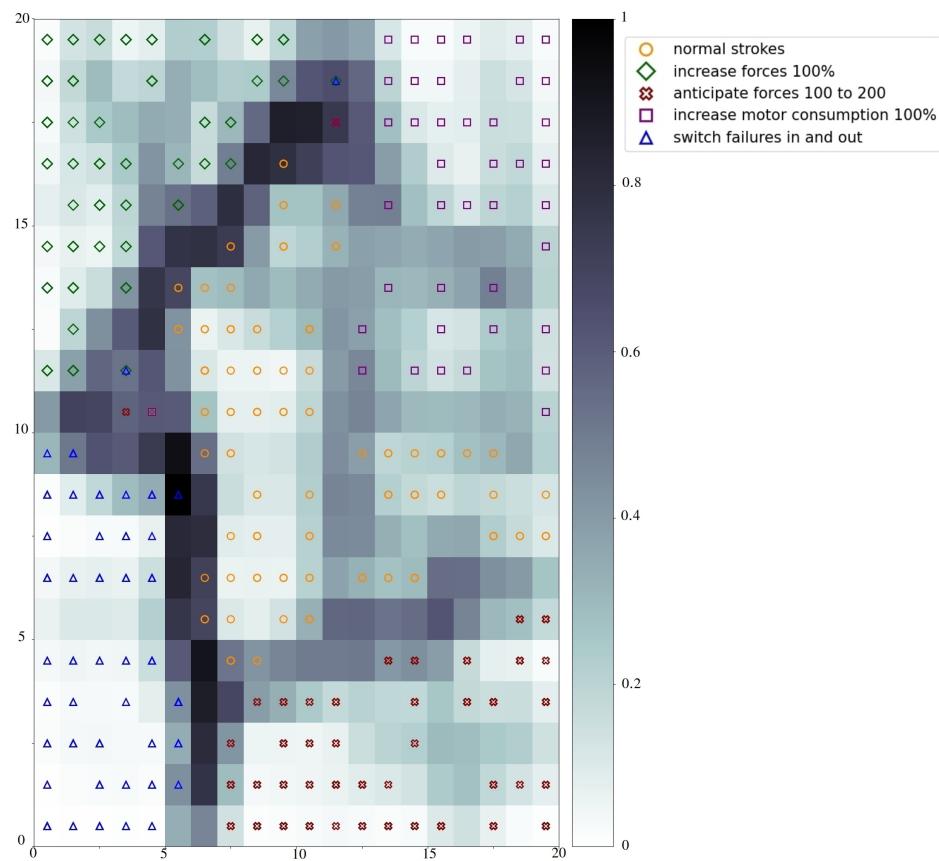


Figure 6. SOM of correct data and main failure types, represented by different forms and colors in a 20×20 grid.

The last diagnosis tool was designed to facilitate fault diagnosis for domain technicians is based on XAI, given that deep learning models are not explanatory by themselves. A final layer was added to the 2D-CNN-AE model of Figure 4 to calculate the RMSE between reconstructed and input data, using the Equation (1). In the equation, n is the number of cycles in the stroke, i indicates a cycle index, m indicates total number of features of the stroke, and j indicates feature index.

$$\begin{aligned} \text{mean_squared_loss_of_feat}_j &= \frac{\sum_{i=1}^n (y_{ij} - \hat{y}_{ij})^2}{n}; \\ \text{loss} &= \sqrt{\frac{\sum_{j=1}^m \text{mean_squared_loss_of_feat}_j}{m}} \end{aligned} \quad (1)$$

SHAP libraries' GradientExplainer class was fitted with samples of correct validation strokes and their losses, so that the explainer learns which is data normality. Then, this explainer was used to diagnostic anomalous strokes categorized by the anomaly detection model. Thus, the explainer propagates each stroke's loss gradient along all layers of the autoencoder until reaching the its input, where SHAP values of each input feature for each evolution variable point are estimated. Afterwards, the absolute value of this SHAP value matrix was calculated, the maximum value of this new matrix was searched, and min-max normalized. Finally, this last matrix is used to plot an indicator of damage with original input data by drawing red rectangles for each evolution variable point, whose transparency is inversely proportional to matrix values. Thus, a matrix value near to 1 has little transparency and will be clear, whereas values near to 0 will be hardly noticeable.

As Figure 7 shows, the developed diagnosis algorithm based on XAI can detect which signals are causing the anomaly in a multivariate approach, and without being previously trained for these failure types. The image shows original stroke data in green, stroke data

modified with one specified failure type in red, and the background is shadowed in red with the explanation metric presented in previous paragraph. This demonstrates that the algorithm is capable of detecting which features and concretely in which cycle points the failure data is not normal. This tool will be used by domain technicians to isolate anomalies detected in a semi-supervised manner.

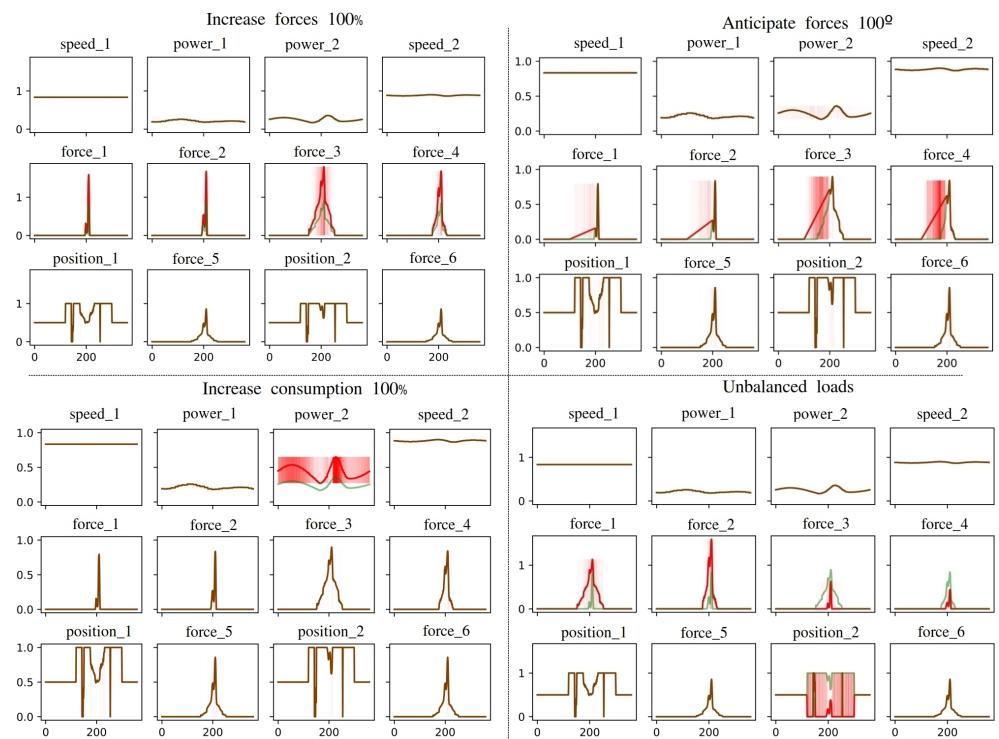


Figure 7. Diagnosis using XAI on the same stroke with 4 failure types, shadowing evolution variables' points that cause them in red color.

This tool was also used to diagnose several outliers detected in training data by the anomaly detection algorithm, aiming to analyze their root cause. Figure 8 segments anomalous training points and analyzes one of these with the XAI-based diagnosis tool. The main difference of this outlier with respect to normal points is the increase of power consumption. This difference is clearly identified by the tool.

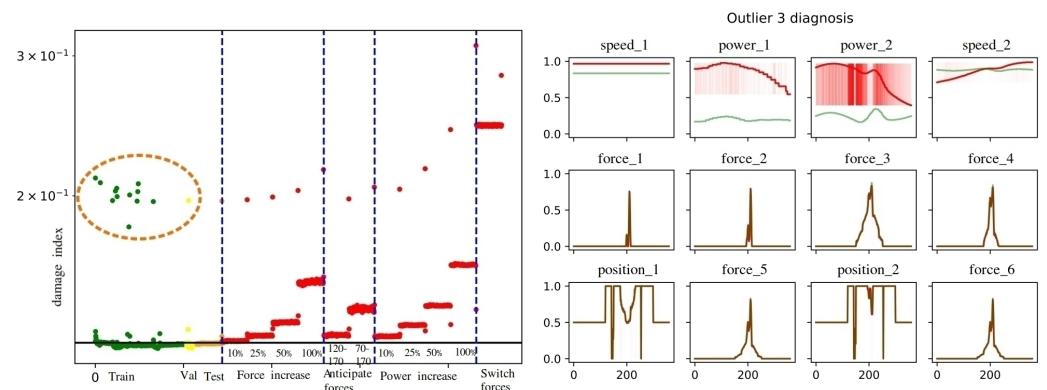


Figure 8. The right side contains diagnosis of outliers in training data using XAI. The left side contains predictions of 2D-CNN-AE damage index on train, validation, test and synthetic failure data, and training outliers are outlined in a dashed circle. Horizontal black line represents the anomaly detection threshold and dashed blue line splits data by each failure type, which are labeled based on their magnitude.

3.3. Adaptability

At this point, model's accuracy, novel identifiability, failure isolability, explanation facility, damage index estimation, and online data processing capability have been tested. The final requirement for the model is to be reusable, adapting to data changes over time. This requirement was tested using another data subset collected a week after the subset used for anomaly detection and diagnosis, which belonged to the same die. To test these experiments, synthetic failures were generated from the new test data split by using the same procedure explained in Section 2.2.

Firstly, the 2D-CNN-AE model trained in Section 3.1 was loaded and executed in the subset of data of the same die 7 days later, calculating F1 score using correct and all failure types on test data. A statistical comparison between original workorder's data and data of new workorder is presented in Figure 9, which shows differences mainly in speed and consumption standard deviation. The model does not work well with new data without being modified, obtaining an average F1 score of 0.77 on failure data. The other models developed at anomaly detection section achieved even lower F1 scores, meaning that they were unable to model new data. At this point, transfer learning experiments were executed in 2D-CNN-AE model, expecting it to adapt to new data given it belonged to similar EOC than data used for training the model. Transfer learning was applied by only training linear layers of the model while keeping original convolutional layers.

Statistical difference between two workorders of the same die

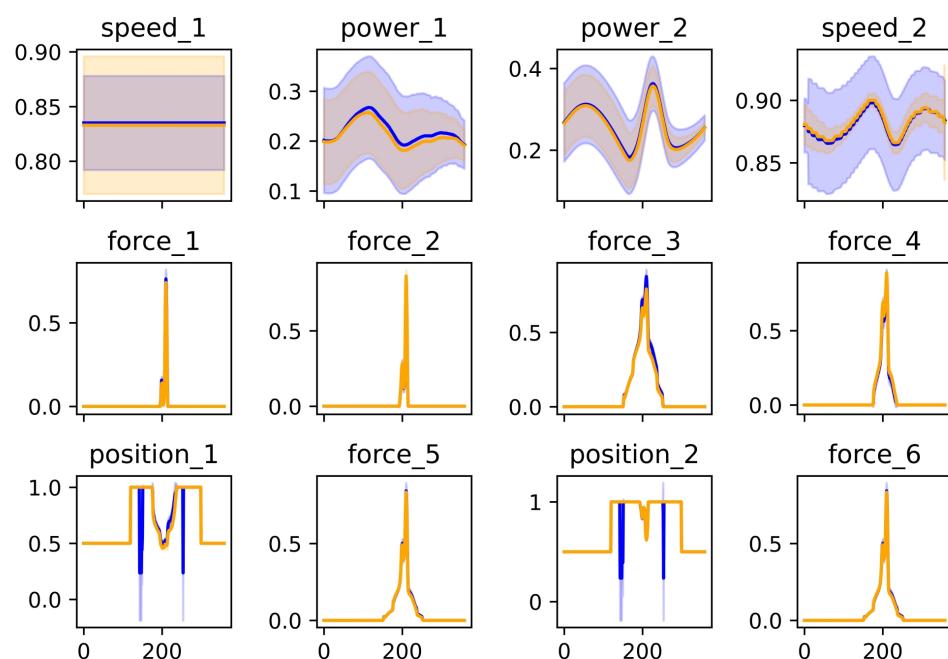


Figure 9. Statistical distribution of two continuous workorders' stroke variables with a time difference of 7 days, represented by blue and orange colors. The average sensor signal or a workorder is represented by a continuous line, and its standard deviation is represented by a semi-transparent envelope range of that signal's color.

The results showed that only 10 normal strokes were required for model retraining and 5 normal strokes for validation to achieve a F1 score of 1 in all failure types. Figure 10 contains images of damage indexes on test correct data before and after retraining, showing that all indexes moved below the anomaly threshold after transfer learning. This means that threshold adaptation is not required when internal layers of the autoencoder are retrained for new data of the same die. In addition, Figure 11 contains damage indexes of correct and

each failure type data before and after retraining the model. It contains different groups of damage index values for each failure type, which correspond to failure magnitudes.

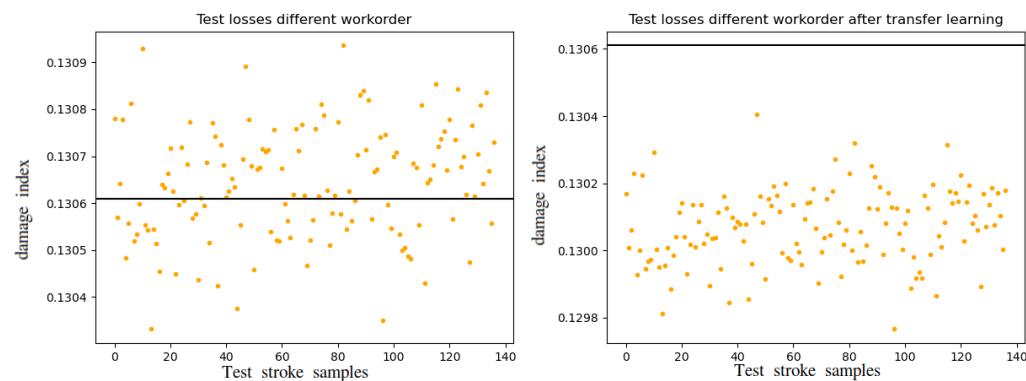


Figure 10. Damage indexes of correct test data before (left) and after transfer learning (right) using 2D-CNN-AE model in data of same die one week later. Horizontal black line represents the anomaly detection threshold.

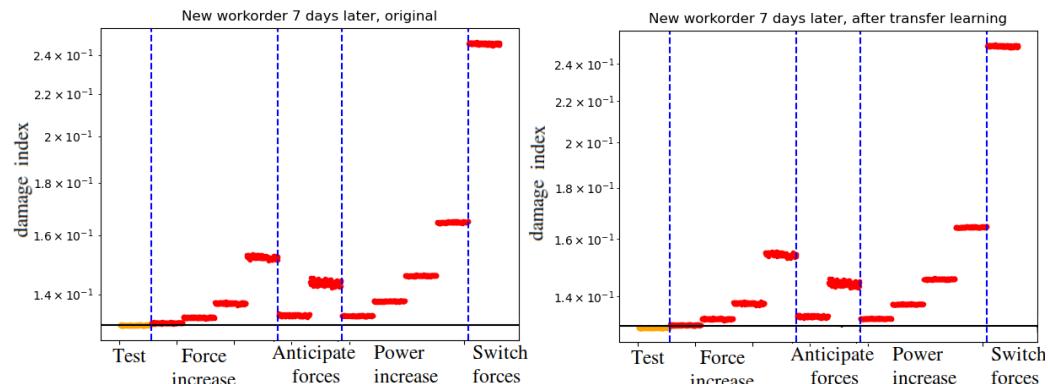


Figure 11. Damage indexes of correct and failure test data before (left) and after transfer learning (right) using 2D-CNN-AE model in data of same die one week later. Correct damage indexes are represented in orange and failure damage indexes in red. Horizontal black line represents the anomaly detection threshold and dashed blue line splits data by each failure type. In addition, each failure type contains all its magnitude versions stacked one after another incrementally, causing a staircase-like behaviour given that higher failure magnitudes result in higher damage indexes.

These experiments show that transfer learning enables model reusability for data of the same die along different time periods. Retraining only linear layers of the model requires less data, training resources, and achieves better results than training the whole model from zero. The experiments were repeated with data of other workorders of the same die, achieving comparable results to the explained in this section.

Finally, model adaptability to other die using transfer learning was tested, executing previous experiments with different workorders' data of another die. When executing the model without retraining, all damage indexes were far above the anomaly threshold. Afterwards, transfer learning was performed on linear layers with all training and validation strokes of this new die dataset, given that using a small number for retraining resulted in fast overfitting. After this transfer learning, damage indexes were analyzed, and they were still far beyond the anomaly threshold, which indicated that different dies require different anomaly threshold. This analysis is presented in Figure 12, which contains train, validation, test correct, and test failure damage indexes before and after transfer learning.

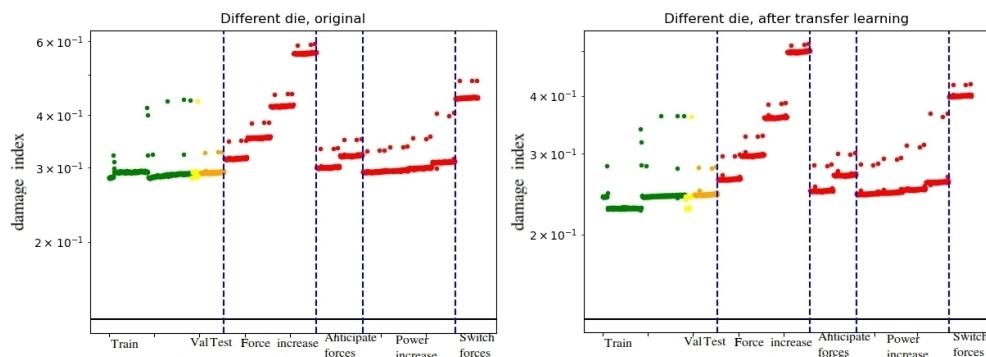


Figure 12. Damage indexes of train, validation, test correct, and test failure test data before (left) and after transfer learning (right) using 2D-CNN-AE model in data of different die. Correct damage indexes are represented in orange and failure damage indexes in red. Horizontal black line represents the anomaly detection threshold and dashed blue line splits data by each failure type.

Then, the anomaly detection threshold was adjusted in the retrained algorithm, using the techniques presented in Section 3.1. Percentile 90 and percentile 95 thresholds on correct validation data were tested, and failure validation data were also used to search for the best threshold. None of these thresholds worked correctly given that damage indexes on training and validation data are smaller than the ones for test correct data. Therefore, all test samples, either correct and failure, are above selected thresholds, which is caused by model overfitting that is unable to differentiate anomalous and correct strokes.

All in all, transfer learning works correctly to adapt the model for data of the same die in different periods of time. However, it does not work to adapt the model trained with data of one die to data of different die. The reason is that data of the same die are collected for the same forming process and under similar operating conditions, but different dies perform different forming processes and have different operating conditions. Therefore, one model for each die should be created, but then they could be reused over time.

4. Discussion and Conclusions

In this work, an accurate semi-supervised deep learning-based 2D-CNN-AE anomaly detection model only trained with correct working machine data of production press machine data has been developed. It outperforms statistical semi-supervised anomaly detection models like PCA and null-space, and traditional machine learning models such as OC-SVM and ELM. This deep learning model can be executed online, monitoring the stamping process to identify anomalous strokes and avoid machine failures.

However, one drawback of deep learning models is their difficulty to be interpreted, given they are black box models. Therefore, several diagnosis tools have been developed to facilitate domain technicians in the identification of possible machine failure types once anomalies are detected. For that purpose, the encoder part of the autoencoder has been used to extract stroke features and obtain a 32 dimensional feature vector that has been used as stroke data descriptor for the diagnosis phase of this work. These data have been used with OPTICS and GMM techniques to cluster failure types, t-SNE for 2D projection and visualization, and SOM for projection to new space using competitive learning. These techniques have successfully isolated different failure types with latent space features, even though these novel failures were not available for model training.

XAI techniques have also been integrated for diagnosis, demonstrating their ability to detect which signal parts of stroke data are responsible for causing the anomaly. In addition, a visual diagnosis tool based on XAI that highlights damaged signals has been created, with the objective to assist domain technicians in the diagnosis of failure causes based on their expertise.

In addition, this work demonstrated that transfer learning enables model adaptability to data variations of same die in different workorders, which allows reusing models with

small adjustments. In contrast, each die requires a model specifically trained for it, as their process and operational conditions are different and therefore, transfer learning does not work.

The validation of developed models was performed by comparing their results on synthetic failures with the results of a questionnaire filled by domain technicians by using the F1 score, obtaining the best algorithms a F1 of 0.99 and 0.92 with respect to an F1 of 0.82 obtained in the questionnaire, on average. In addition, the models' damage index and their numerical relation with anomaly thresholds was also validated with domain technicians, concluding that synthetic failures of the same type created with higher magnitude signal modifications resulted in higher damage index values. These tasks ensured that developed algorithms addressed use-case requirements correctly.

All in all, this work that combines clustering, visualization, projection, and XAI techniques with a deep learning model designed to address adaptability, novelty detection, and diagnosis requirements. The resulting model has an F1 score of 0.99 in anomaly detection while being explainable for root cause analysis. Future research will continue with the analysis of machine condition evolution over time and monitoring model performance.

Even though each industrial use case has its own requirements and data characteristics, the techniques implemented in this work can be reused in other PdM use cases after adaptations. This work's contributions on semi-supervised anomaly detection, semi-supervised diagnosis, and adaptability with transfer learning can increase stakeholders' confidence on developed models, facilitating the adoption of machine learning and deep learning-based predictive maintenance systems in industrial environments.

Author Contributions: Conceptualization, O.S. and E.Z.; methodology, O.S.; software, O.S.; validation, E.Z., J.R.d.O. and U.Z.; formal analysis, O.S.; investigation, O.S.; resources, E.Z., J.R.d.O., J.R., and U.Z.; data curation, O.S., J.R., and J.R.d.O.; writing—original draft preparation, O.S.; writing—review and editing, O.S., E.Z., and U.Z.; visualization, O.S.; supervision, O.S., E.Z., and U.Z.; project administration, O.S., E.Z. and U.Z.; funding acquisition, O.S., E.Z., J.R.d.O., J.R., and U.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Oscar Serradilla, Ekhi Zugasti and Urko Zurutuza are part of the Intelligent Systems for Industrial Systems research group of Mondragon Unibertsitatea (IT1357-19), supported by the Department of Education, Universities and Research of the Basque Country. This work has been partially funded by the European Union's Horizon 2020 research and innovation programme's project QU4LITY under grant agreement n.825030, and by the Provincial Council of Gipuzkoa's project MEANER under grant agreement FA/OF326/2020(ES).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article given privacy issues.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

2D-CNN-AE	2 dimensional convolutional-based autoencoder
ELM	Extreme learning machine
EOC	Environmental and operational conditions
GMM	Gaussian mixture models
OC-SVM	One-class support vector machines
OPTICS	Ordering points to identify the clustering structure
PCA	Principal component analysis

PdM	Predictive maintenance
SHAP	Shapley additive explanations
SOM	Self-organizing map
t-SNE	T-distributed stochastic neighbor embedding
XAI	Explainable artificial intelligence

References

- Dhillon, B.S. *Engineering Maintenance: A Modern Approach*; CRC Press: Boca Raton, FL, USA, 2002; pp. 1–224.
- Wang, K.; Wang, Y. How AI Affects the Future Predictive Maintenance: A Primer of Deep Learning. In *Advanced Manufacturing and Automation VII*; Wang, K., Wang, Y., Strandhagen, J.O., Yu, T., Eds.; Springer: Singapore, 2018; pp. 1–9.
- UE Systems. Understanding the P-F Curve and Its Impact on Reliability Centered Maintenance. 2014. Available online: <http://www.uesystems.com/news/understanding-the-p-f-curve-and-its-impact-on-reliability-centered-maintenance> (accessed on 15 June 2021).
- Colemen, C.; Damodaran, S.; Chandramoulin, M.; Deuel, E. *Making Maintenance Smarter*; Deloitte University Press: New York, NY, USA, 2017.
- Lavi, Y. The Rewards and Challenges of Predictive Maintenance. *InfoQ*. 2018. Available online: <https://www.infoq.com/articles/predictive-maintenance-industrial-iot> (accessed on 28 May 2021).
- Jimenez-Cortadi, A.; Irigoien, I.; Boto, F.; Sierra, B.; Rodriguez, G. Predictive Maintenance on the Machining Process and Machine Tool. *Appl. Sci.* **2020**, *10*, 224. [[CrossRef](#)]
- Welz, Z.A. Integrating Disparate Nuclear Data Sources for Improved Predictive Maintenance Modeling: Maintenance-Based Prognostics for Long-Term Equipment Operation. Ph.D. Thesis, University of Tennessee, Knoxville, TN, USA, 2017.
- Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S.N. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Comput. Chem. Eng.* **2003**, *27*, 293–311. [[CrossRef](#)]
- Serradilla, O.; Zugasti, E.; Zurutuza, U. Deep learning models for predictive maintenance: A survey, comparison, challenges and prospect. *arXiv* **2020**, arXiv:2010.03207.
- Bampoula, X.; Siaterlis, G.; Nikolakis, N.; Alexopoulos, K. A Deep Learning Model for Predictive Maintenance in Cyber-Physical Production Systems Using LSTM Autoencoders. *Sensors* **2021**, *21*, 972. [[CrossRef](#)]
- Zhang, S.; Ye, F.; Wang, B.; Habetler, T.G. Semi-Supervised Bearing Fault Diagnosis and Classification using Variational Autoencoder-Based Deep Generative Models. *IEEE Sens. J.* **2020**, *21*, 6476–6486. [[CrossRef](#)]
- Bui, V.; Pham, T.L.; Nguyen, H.; Jang, Y.M. Data Augmentation Using Generative Adversarial Network for Automatic Machine Fault Detection Based on Vibration Signals. *Appl. Sci.* **2021**, *11*, 2166. [[CrossRef](#)]
- Kuo, R.; Li, C. Predicting Remaining Useful Life of Ball Bearing Using an Independent Recurrent Neural Network. In Proceedings of the 2020 2nd International Conference on Management Science and Industrial Engineering, Osaka, Japan, 7–9 April 2020; pp. 237–241.
- Yuan, J.; Wang, Y.; Wang, K. LSTM based prediction and time-temperature varying rate fusion for hydropower plant anomaly detection: A case study. In *Advanced Manufacturing and Automation VIII. IWAMA 2018*; Lecture Notes in Electrical Engineering; Springer: Singapore, 2019; Volume 484, pp. 86–94. [[CrossRef](#)]
- Schwartz, S.; Jimenez, J.J.M.; Salaün, M.; Vingerhoeds, R. A fault mode identification methodology based on self-organizing map. *Neural Comput. Appl.* **2020**, *32*, 13405–13423. [[CrossRef](#)]
- Talmoudi, S.; Kanada, T.; Hirata, Y. Tracking and visualizing signs of degradation for an early failure prediction of a rolling bearing. *arXiv* **2020**, arXiv:2011.09086.
- Wen, L.; Gao, L.; Li, X. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 136–144. [[CrossRef](#)]
- Wen, T.; Keyes, R. Time Series Anomaly Detection Using Convolutional Neural Networks and Transfer Learning. *arXiv* **2019**, arXiv:1905.13628.
- Martínez-Arellano, G.; Ratchev, S. Towards an active learning approach to tool condition monitoring with bayesian deep learning. In Proceedings of the ECMS 2019: 33rd International ECMS Conference on Modelling and Simulation, Caserta, Italy, 11–14 June 2019; Volume 33, 223–229. [[CrossRef](#)]
- Zope, K.; Singh, K.; Nistala, S.; Basak, A.; Rathore, P.; Runkana, V. Anomaly detection and diagnosis in manufacturing systems: A comparative study of statistical, machine learning and deep learning techniques. *Annu. Conf. PHM Soc.* **2019**, *11*. [[CrossRef](#)]
- Brito, L.C.; Susto, G.A.; Brito, J.N.; Duarte, M.A. An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery. *Mech. Syst. Signal Process.* **2022**, *163*, 108105.
- Lee, S.; Yu, H.; Yang, H.; Song, I.; Choi, J.; Yang, J.; Lim, G.; Kim, K.S.; Choi, B.; Kwon, J. A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment. *Appl. Sci.* **2021**, *11*, 1564.
- Maschler, B.; Vietz, H.; Jazdi, N.; Weyrich, M. Continual learning of fault prediction for turbofan engines using deep learning with elastic weight consolidation. In Proceedings of the 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020; Volume 1, pp. 959–966.

24. Maschler, B.; Kamm, S.; Weyrich, M. Deep industrial transfer learning at runtime for image recognition. *at-Automatisierungstechnik* **2021**, *69*, 211–220.
25. Olaizola Alberdi, J. Soft Sensor-Based Servo Press Monitoring. Ph.D. Thesis, Mondragon Unibertsitatea, Goi Eskola Politeknikoa, Mondragon, Spain, 2020.
26. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52.
27. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501.
28. Hejazi, M.; Singh, Y.P. One-class support vector machines approach to anomaly detection. *Appl. Artif. Intell.* **2013**, *27*, 351–366.
29. Meire, M.; Karsmakers, P. Comparison of deep autoencoder architectures for real-time acoustic based anomaly detection in assets. In Proceedings of the 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Metz, France, 18–21 September 2019; Volume 2, pp. 786–790.
30. Zugasti, E.; Iturbe, M.; Garitano, I.; Zurutuza, U. Null is not always empty: Monitoring the null space for field-level anomaly detection in industrial IoT environments. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; pp. 1–6.
31. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
32. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60.
33. Lindsay, B.G. Mixture models: Theory, geometry and applications. In *NSF-CBMS Regional Conference Series in Probability and Statistics*; Institute of Mathematical Statistics and the American Statistical Association: Boston, MA, USA, 1995; Volume 5, pp. i-iii+v-ix+1-163.
34. Kohonen, T. The self-organizing map. *Proc. IEEE* **1990**, *78*, 1464–1480.
35. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4 December 2017; pp. 4765–4774.
36. Lundberg, S. A game theoretic approach to explain the output of any machine learning model. *Github*. 2021. Available online: <https://github.com/slundberg/shap> (accessed on 28 June 2021).
37. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. *Proc. Mach. Learn. Res.* **2017**, *70*, 3319–3328.
38. Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; Wattenberg, M. Smoothgrad: Removing noise by adding noise. *arXiv* **2017**, arXiv:1706.03825.
39. Tensorflow. TensorFlow 2.3 documentation. Available online: <https://devdocs.io/tensorflow~2.3/> (accessed on 28 June 2021).
40. Scik-it learn. Scik-it learn 0.23.2 documentation. Available online: https://scikit-learn.org/stable/whats_new/v0.23.html (accessed on 28 June 2021).
41. Vettigli, G. Minisom documentation - Minimalistic and NumPy-based implementation of the Self Organizing Map. Available online: <https://github.com/JustGlowing/minisom> (accessed on 28 June 2021).