

## Article

# Predicting the Health Status of a Pulp Press Based on Deep Neural Networks and Hidden Markov Models

Alexandre Martins <sup>1,2,\*</sup> , Balduino Mateus <sup>1,2</sup> , Inácio Fonseca <sup>3</sup> , José Torres Farinha <sup>3,4</sup> , João Rodrigues <sup>1,2</sup> ,  
Mateus Mendes <sup>3</sup>  and António Marques Cardoso <sup>2</sup> 

<sup>1</sup> EIGeS—Research Centre in Industrial Engineering, Management and Sustainability, Lusófona University, Campo Grande, 376, 1749-024 Lisboa, Portugal

<sup>2</sup> CISE—Electromechatronic Systems Research Centre, University of Beira Interior, 62001-001 Covilhã, Portugal

<sup>3</sup> Instituto Superior de Engenharia de Coimbra, Polytechnic of Coimbra, 3045-093 Coimbra, Portugal

<sup>4</sup> Centre for Mechanical Engineering, Materials and Processes—CEMMPRE, University of Coimbra, 3030-788 Coimbra, Portugal

\* Correspondence: alex\_bmts@hotmail.com

**Abstract:** The maintenance paradigm has evolved over the last few years and companies that want to remain competitive in the market need to provide condition-based maintenance (CBM). The diagnosis and prognosis of the health status of equipment, predictive maintenance (PdM), are fundamental strategies to perform informed maintenance, increasing the company's profit. This article aims to present a diagnosis and prognosis methodology using a hidden Markov model (HMM) classifier to recognise the equipment status in real time and a deep neural network (DNN), specifically a gated recurrent unit (GRU), to determine this same status in a future of one week. The data collected by the sensors go through several phases, starting by cleaning them. After that, temporal windows are created in order to generate statistical features of the time domain to better understand the equipment's behaviour. These features go through a normalisation to produce inputs for a feature extraction process, via a principal component analysis (PCA). After the dimensional reduction and obtaining new features with more information, a clustering is performed by the K-means algorithm, in order to group similar data. These clusters enter the HMM classifier as observable states. After training using the *Baum–Welch* algorithm, the *Viterbi* algorithm is used to find the best path of hidden states that represent the diagnosis of the equipment, containing three states: state 1—“State of Good Operation”; state 2—“Warning State”; state 3—“Failure State”. Once the equipment diagnosis is complete, the GRU model is used to predict the future, both of the observable states as well as the hidden states coming out from the HMM. Thus, through this network, it is possible to directly obtain the health states 7 days ahead, without the necessity to run the whole methodology from scratch.

**Keywords:** maintenance; diagnosis; prognosis; deep neural network; hidden Markov models; machine learning



**Citation:** Martins, A.; Mateus, B.; Fonseca, I.; Farinha, J.T.; Rodrigues, J.; Mendes, M.; Cardoso, A.M. Predicting the Health Status of a Pulp Press Based on Deep Neural Networks and Hidden Markov Models. *Energies* **2023**, *16*, 2651. <https://doi.org/10.3390/en16062651>

Academic Editor: Daniel Morinigo-Sotelo

Received: 13 January 2023

Revised: 2 March 2023

Accepted: 9 March 2023

Published: 11 March 2023



**Copyright:** © 2023 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 mechanical systems of companies' production equipment suffer degradation and the remaining useful life (RUL) shortens as the equipment components deteriorate over time [1]. With the current evolution of production equipment, its complexity has also increased. Thus, modern industrial systems are often exposed to various failure modes [2]. An unexpected stoppage of equipment can cause great economic losses to a company and/or even put the health of workers at risk.

### 1.1. Maintenance

According to Kumar et al. [3], it is estimated that 85% of the total life-cycle cost of equipment is determined by decisions made during its operation, where maintenance actions are included. Therefore, it becomes necessary to obtain a maintenance system

adaptable to the criticality of the equipment [4]. “Chronologically, maintenance strategies have evolved from the naive breakdown or run-to-failure maintenance to preventive maintenance, with first static time-based preventive maintenance, and then condition-based preventive maintenance (CBM)” [5]. Thus, maintenance can be divided into three major categories: corrective maintenance (CM), time-based maintenance (TBM) and condition-based maintenance (CBM) [6]. CM is a maintenance strategy that acts on the system only to repair or replace the failed components. It is a maintenance policy that can cause unexpected equipment downtime, production stoppages and low safety and is only used for low-criticality equipment [7]. TBM and CBM are seen as types of proactive maintenance since they are done before the failure occurs. TBM is based on reliability parameters of the equipment components and its maintenance is usually done periodically and before the failure occurs, which can lead to excessive maintenance. On the other hand, CBM depends on the condition of the system being monitored, and interventions only occur when they are really necessary. It has many advantages over other strategies [8].

### 1.2. Condition-Based Maintenance—CBM

CBM is a strategy that is based on condition monitoring of equipment over its operations, enabling just-in-time maintenance [8]. According to Leoni et al. [9], one of the main advantages over preventive maintenance strategies is that it can be scheduled only based on the condition of the equipment, with no need to plan interventions based on reliability parameters. Lee et al. [10] mention that although implementing a CBM strategy is more difficult and costly, it ultimately leads to less wasted equipment life, reducing costs associated with production, parts stock, number of personnel, tools, etc. It also increases equipment availability and decreases the number of unexpected stops, improving failure prevention and simultaneously reducing the operational cost [11–13]. According to Zhang et al. [14], a CBM policy makes better use of system information, reduces the risk of failures and unnecessary maintenance. Therefore, CBM is a maintenance methodology, which, once implemented, is more cost effective than traditional methodologies, which helps to improve equipment reliability, reduce operating costs, improve safety and reduce the frequency and severity of equipment failures [12,15–20]. In addition to the points enumerated above, with the advancement of technology and the arrival of industry 4.0., companies are becoming more and more sensorized, and it becomes evident that CBM is the maintenance that is most suitable and can achieve the best results [17,21–26]. CBM is used in smart factories through the use of IoT systems, CPS (cyberphysical systems), sensor technology and AI technologies. In this way, the collection and processing of large quantities of data (big data) can be done in real time [9,10,27,28].

CBM is the maintenance responsible for detecting, through condition monitoring, the first signs of anomalies in [29] equipment. In this way, it is possible to determine when maintenance is required based on the actual condition of the equipment. Through condition monitoring, it is possible to inspect and assess the current state of a piece of equipment through data collected in real time from different types of sensors (e.g., vibration, noise, temperature, etc.) [1,16,30,31]. The better the adequacy of equipment degradation modelling as well as the better the monitoring accuracy, the better the CBM strategy becomes [11]. Moreover, the reliability of the collected data is very important for this type of maintenance, hence the need to have calibrated sensors that make reliable measurements [32,33].

Summarizing, CBM can be divided into three phases [9,10,18,34]:

- (i) Data acquisition, where useful data are collected, usually from sensors;
- (ii) Data processing, where noise components are filtered out of the physical observations and subsequently, a data analysis is performed;
- (iii) Maintenance decision-making.

### 1.3. Diagnosis and Prognosis

CBM can be divided into two types of decision: failure diagnostics, which is the ability to detect a cause of failure; failure prognostics, which is responsible for detecting a failure that may occur in the future [35]. The prognosis of the health status of a piece of equipment has become an unavoidable concept in the context of today's industry, where intelligent manufacturing and industrial big data provide input solutions for maintenance [36,37]. Then, through a CBM strategy, it is also possible to make a prognosis of the state of the equipment, having the indication of possible problems and/or failures that could happen in the future [38]. Effective prognostic techniques that can anticipate future conditions are integrated into an advanced predictive maintenance framework that is incorporated into the condition monitoring system [1]. We then speak of predictive maintenance (PdM), whose aim is to predict the condition diagnosis of the equipment.

### 1.4. Predictive Maintenance PdM

PdM is a CBM strategy, but instead of diagnosing the equipment, it performs a real-time prognosis [5]. Thus, as the authors pointed out, PdM is more efficient at planning maintenance actions than a CBM policy. PdM improves productivity, product quality and overall manufacturing efficiency [24]. According to Harald et al. [39], predictive maintenance can reduce machine downtime by 30% to 50% and extend machine life by 20% to 40%. Therefore, it is possible to think of predictive maintenance as being more accurate than condition-based maintenance [40]. The aim of CBM is to identify deviations or substantial modifications that are usually signs of a development failure. Consequently, condition-based monitoring serves as a crucial pillar of predictive maintenance [40].

By monitoring a machine's operating conditions in real time, it is possible to detect key patterns for future failure prediction, thus predicting maintenance actions. According to Yam et al. [38], an "alarm" can be given when values outside the normal operating range are predicted, allowing system operators to take appropriate action to check machine conditions and repair faults before a more critical failure, hence the need for companies to acquire an equipment-condition-based maintenance strategy with a focus on prediction. Then, a predictive maintenance strategy is also based on equipment condition monitoring but integrates prognostics to effectively conclude on the state of equipment health in the future [1]. According to Oakley et al. [17], predictions based on information extracted from values collected by sensors have helped to improve decision-making regarding the maintenance of equipment.

### 1.5. HMM-GRU to Perform Maintenance

Normally, the phenomena of the degradation evolution of equipment components have a stochastic nature and can be described by stochastic modelling processes [19,34,41]. These stochastic models are modelled based on probability and statistical theories and can be related to component degradation and failure occurrences [41]. In this paper, we use a doubly stochastic process, the hidden Markov model (HMM), to characterise the state of equipment degradation. The objective of the HMM is to classify the health state of the equipment. To do this, we use the data collected by the sensors responsible for monitoring the equipment. These types of data are characterized by time series data, since they are collected over time in a continuous manner. Time series data are the most collected type of data in this new era of industry 4.0. These observations collected over time can be analysed and used as a tool that prevents unexpected equipment failures [42]. Thus, it is essential to use machine learning (ML) tools, which is a field of AI that extracts key patterns from collected temporal data through different paradigms such as: supervised learning, semisupervised learning, unsupervised learning and reinforcement learning. According to Kiangala and Wang [42], deep learning is a branch of ML consisting of different methods such as: artificial neural network (ANN), convolution neural networks (CNN), long short-term memory (LSTM), etc. In this paper, we will use the Gated Recurrent Units (GRU) tool, which is like an LSTM network with a forgetting gate. This is currently a well-recognized

prognostic tool and is the tool responsible for making the prediction of future data later classified by the HMM to obtain the health status of the equipment, thus performing a prognosis and supporting a PdM methodology.

### *1.6. Related Work*

There are several examples of papers supporting PdM, which use DNN to make predictions of the future, as well as other papers that use HMM to classify the state of a system. Mateus et al. [43] made a comparison about a predictive system using an LSTM network or a closed recurrent unit (GRU) for a multivariate data set. Antunes et al. [44] proposed in their paper a variation of the exponential smoothing technique for short-term forecasting and an artificial neural network for long-term forecasting. Mateus et al. [45] presented in his article predictive models using an LSTM network to predict future equipment status based on data from an industrial paper press. Zhang et al. [46] proposed an approach to perform a prognosis of rotating equipment's health using wavelet transform (WT), a principal component analysis (PCA) and artificial neural networks (ANN) to classify the failure and predict the condition of components, equipment and machines. Martins et al. [47] showed how it was possible to classify the health condition of equipment via an HMM with multivariate analysis. Yu [48] proposed an adaptive HMM method that evolved over time to detect equipment failures and component degradation monitoring. Arpaia et al. [49] presented a fault detection method using HMM for fluid machinery without a priori information about its failure conditions. Mateus et al. [50] presented a case study where they determined the future behaviour of the data collected by sensors coupled to industrial equipment; the authors used time series models and deep learning.

### *1.7. Contributions of the Paper*

In our case, the article stands out for integrating a diagnosis methodology that uses an HMM with a forecasting tool, the GRU. In this way, the methodology explained in this article demonstrates how a diagnosis and prognosis of the production equipment's health status can be made in an online mode. The added value of the methodology is that it is generic and can be used in any equipment with different sensors. Furthermore, it is a methodology that detects unusual patterns in the equipment's operation without prior information using unsupervised ML tools. Furthermore, the GRU predicts optimised observable states after going through a process of feature generation, PCA and clustering. That is, the GRU does not make the prediction directly on the data collected by the sensors but on a data set optimized by ML processes. Moreover, a future prediction is made with the GRU algorithm directly on the hidden states classified by the HMM. Thus, the methodology does not only make a future prediction about the observable data that need to be classified but also a prediction of the classification itself performed by the HMM.

### *1.8. Paper Structure*

This article is divided as follows: Section 1 gives an introduction with the objective to explain the context of the article, where maintenance concepts are described and where the tools used to accomplish the diagnosis and prognosis of equipment are introduced, as well as related works. In Section 2, a theoretical framework of each of the tools used in this article is presented. Section 3 talks about the methodology used and explains how it can be performed both to make a diagnosis of the equipment using the HMM, as well as a prognosis using the GRU. Section 4 describes the case study carried out in a production equipment used in the paper industry, where the objective was to realize 3its diagnosis and prognosis. In Section 5, a discussion of the results is provided, where an evaluation of the methodology is performed to understand if it works. Finally, in Section 6, the conclusion of the work is drawn.

## 2. Background

### 2.1. Principal Component Analysis (PCA)

A principal component analysis (PCA) is an unsupervised learning method of feature extraction and dimensional reduction (moving p-dimensional data to a lower-dimensional m-dimensional linear subspace), retaining the original features of the data and selecting their key properties [42,51–53]. It analyses a data table in which observations are described by several intercorrelated quantitative dependent variables and is widely used due to its ability to extract interpretable information by efficiently removing redundancies [54,55]. It is typically used to perform the dimensional reduction of large sets of time series observations [56], moving from representing possibly correlated variables to a new set of orthogonal, uncorrelated variables and preserving the highest percentage of information [40,46,55]. In this way, it allows a rapid assessment of any relationships between variables [54]. In other words, it is a method of projecting large dimensional measurements towards a minimum dimensional space and preserving maximum variance [57] by compressing sensory data according to their spatial and temporal correlations [58]. Then, the PCA produces linear combinations of the original variables to generate new axes, known as principal components (PCs), with the first PC having as high a variance as possible, possessing the greatest variability in the data, and each subsequent component in turn having as high a variance as possible under the constraint that it is not correlated with the previous components [46,54]. In other words, the PCA is a linear transformation that rearranges the data into a new coordinate system, in which the first PC is defined as the coordinate that shows the greatest variation in the data when projected in that direction; the second PC is the coordinate that presents the second largest variation, and so on for the other components [54].

### 2.2. K-Means Algorithm

The K-means algorithm is a clustering, nonhierarchical, unsupervised learning method in ML, from the branch of multivariate statistical analysis, where the number of clusters  $K$  is determined, and the observations closest to the cluster centre are included in that same cluster [59–63]. Clusters are formed so that the distribution of samples among clusters maximises intragroup cohesion, i.e., the distance of observations from their centroid, increasing similarity within the same cluster, and dissimilarity between different clusters [63–66]. K-means clustering is used to perform the classification of unlabelled data in which the specific response variables are unknown [67]. In this process, the similarity between observations in the same cluster increases and the similarity with data from other clusters decreases [67,68].

The K-means algorithm uses the distance between data points as the standard measure of data similarity, usually using the Euclidean distance [60,69] (which is the one used in this paper). That is, it minimises the square sum of the distances from each data point to its assigned grouping centre [70]. The Euclidean distance equation is represented below in Equation (1) [60,61,69,71].

$$\text{dist}(x_i, x_j) = \sqrt{\sum_{d=1}^D (x_{i,d} - x_{j,d})^2} \quad (1)$$

where  $x_i, x_j$ —are two sequence points and  $D$  represents the dimension.

The K-means approach can be quite sensitive to the initial value chosen for the number of clusters ( $k$ ), which, if improperly defined, can significantly affect the result of the clustering process, the number of iterations needed to reach algorithm convergence, as well as the accuracy and complexity of the clustering algorithm [59,60,68]. Therefore, to minimise the influence of the initial choice of  $k$  and define the optimal number of clusters, it is common to use decision support methods. The Elbow method is a technique used for this purpose, in which the cost function is calculated for different values of  $k$  during the clustering process [61]. Usually, it is plotted a graph of the cost function as a function of

different values of  $k$  is plotted and, from this graph, it is possible to identify the “elbow” point that indicates the optimal number of clusters.

$$E_k = \sum_{r=1}^k \frac{1}{nr} D_r \quad (2)$$

where  $k$  denotes the size of the cluster,  $nr$  represents the number of data points in the cluster and  $D_r$  is the sum of the distances between all points within the cluster.

### 2.3. Gated Recurrent Unit (GRU)

The GRU was introduced by Cho et al. [72] and is based on LSTM, with a relatively simpler structure requiring fewer parameters for its formation, having only two gates (Figure 1) [73–79]: the reset gate is a mechanism that can be used to help with model encryption, allowing one to determine the amount of past information that can be forgotten; the update gate, on the other hand, is responsible for combining the entry and forgetting gates in an LSTM model, allowing one to determine the degree of the previous hidden state that will be used to update the current state. Both the reset gate and the update gate are mechanisms used to solve the leak gradient problem in neural networks, since they allow the manipulation of the information in intermediate layers without losing relevant information for future predictions [76]. According to the authors, what makes these mechanisms special is their ability to maintain long-term memories, without removing relevant information for future predictions, which enables a better model performance.

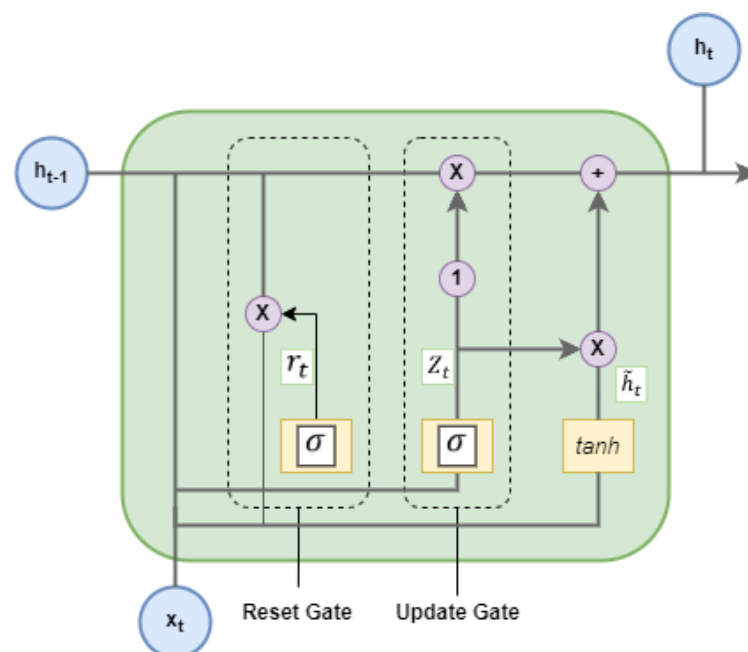


Figure 1. Recurrent Gated Neural Network.

Then, the GRU has an update gate  $z$  and a reset gate  $r$  to simplify the memory block structure of the original LSTM network. The input of the GRU network being  $x_t$ , the formula to calculate the next output and state value in GRU is [73,75–77,80,81]:

1. The algorithm starts with the calculation of the update gate  $z_t$  for time step  $t$  (Equation (3)):

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]); \quad (3)$$

when connecting the network unit, the value  $x_t$  is multiplied by its respective weight  $W_z$ , as well as the value  $h_{t-1}$ , which contains the information of the previous units in  $t - 1$ . The results of these multiplications are then summed, and an activation function *sigmoid* is applied to normalise the result between 0 and 1. In this way, the relevant information is kept and irrelevant information is filtered out.

2. The reset gate allows the model to determine how much past information should be forgotten, thus controlling how much information is retained (Equation (4)):

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (4)$$

3. In this step, new memory content is introduced that uses the reset gate to store the most important information from the past. After obtaining the toggle signal, the toggle reset activation function is used to obtain the reset data and combine them with the *tanh* activation function, resulting in  $\tilde{h}_t$ .

$$\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t]) \quad (5)$$

*tanh* is capable of controlling the range of output values between  $-1$  and  $1$ . It is possible to observe that the input data are incorporated and the hidden information is regulated by the *tanh* activation function.

4. Finally, the vector  $h_t$  is calculated to contain the relevant information of the current unit and transmit it to the next stage of the network, determining what should be kept from the previous stages  $h_{t-1}$ . The end result,  $h_t$ , contains the current unit and previous step information that is relevant to the final output (Equation (6)):

$$h_t = (1 - z_t) \times h_{t-1} + z_t * \tilde{h}_t \quad (6)$$

Through direct training, it can be inferred that the state of the reset gate  $r_t$  controls the combination between the input  $x_t$  and the previous state  $h_{t-1}$ . On the other hand, the  $z_t$  update gate determines the use of current and previous-time information.

#### 2.4. Hidden Markov Models (HMMs)

An HMM is a statistical modelling technique widely used to model sequential data such as time series. Its dynamic Bayesian network structure is relatively simple, but it can capture complex patterns of temporal dependence between observable variables and latent (unobservable) variables [82]. It is developed on the basis of the Markov chain, which is a discrete memoryless random process responsible for describing the relationship between the sequence of states of the next moment with the current one [83–85]. An HMM is an evolution of a Markov chain that requires two stochastic processes, adding a random relationship between the sequence of states and the observation vector, and where the sequence of states cannot be directly observed [83,84,86–89]. Then, an HMM is a probabilistic time series model, doubly stochastic, which includes the transition of hidden states and emitting observations [90]. The hidden state transition, which follows Markov chains, is the actual state within the system, mapped by observable states, which are directly observed and have a correlation with the hidden states [90–93].

A typical HMM can be expressed by  $\lambda = (N, M, \pi, A, B)$  [82–84,90,91,93,94], where:

- $N$  represents the number of hidden states, where a certain state  $q_t \in S, S = S_1, S_2, \dots, S_N$ ;
- $M$  is the number of observable states, where the observation at time  $t$  corresponds to the hidden state  $q_t$  and is represented by  $O_t \in V, V = V_1, V_2, \dots, V_M$ ;
- $\pi$  is the initial distribution of the hidden states, where  $\pi_i = P(q_1 = S_i)$ ;
- $A$  is the matrix of transition probabilities between hidden states, where  $A = a_{ij_{N \times M}}$  and where  $a_{ij}$  equals:

$$a_{ij} = P(q_{i+1} = S_j | q_i = S_i), 1 \leq i, j \leq N \quad (7)$$

- $B$  is the emission matrix, where the probability that the  $j$ th hidden state generates the  $i$ th observable state is represented, where  $B = b_{jk_{N \times M}}$ .  $b_{jk}$  is represented by:

$$b_{jk} = P(O_t = V_k | q_t = S_j), 1 \leq j \leq N, 1 \leq k \leq M \quad (8)$$

There are three problems that need to be solved to use an HMM in an integrity assessment and health-status prediction of equipment [83,93]:

- Evaluation

Using the model  $\lambda = (\pi, A, B)$  and an observation sequence  $O = O_1, O_2, \dots, O_T$ , the probability of the observation sequence  $O$  is determined by the forward–backward algorithm. This algorithm evaluates the model with the most suitable observation.

- Training

It consists of training the model with the HMM, where given a sequence of observations, the model parameters are re-evaluated to maximise the likelihood. The Baum–Welch algorithm is a learning technique for tuning the parameters of an HMM model based on observed data, establishing the relationship between the parameters of the old and new HMM model, and continuing to iterate until convergence is achieved. Specifically, the HMM elements,  $\lambda = (\pi, A, B)$ , are identified such that the likelihood,  $P(O|\lambda)$ , of the observation sequence  $O$  given the  $\lambda$  model is maximised [87,92].

- Decoding

The ideal state sequence is created in order to ensure that the likelihood will reach its highest value given the model,  $\lambda$ , and the observation sequence,  $O$ . Viterbi's method solve the HMM decoding problem, inferring the HMM's most likely series of hidden states,  $S$  [91].

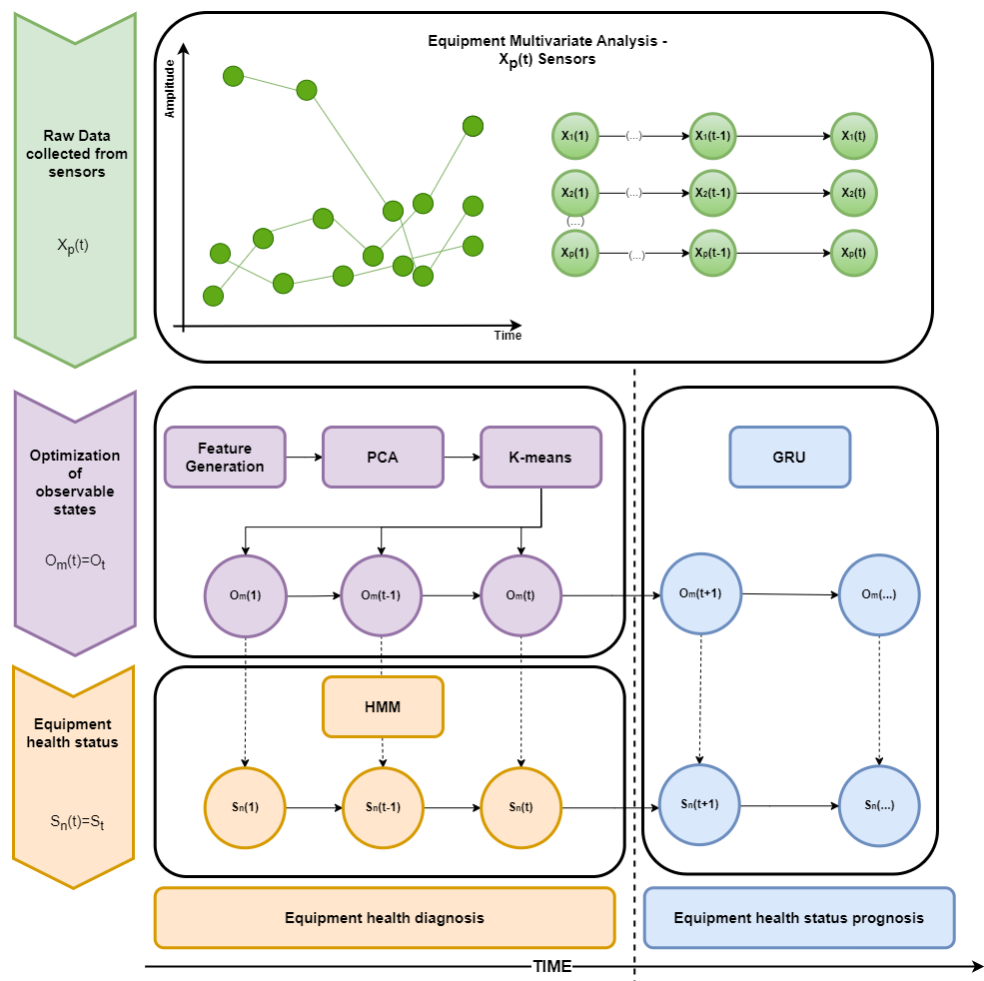
### 3. Methodology

In order to explain the procedure and methodology used in the case study presented in this article, we can follow Figure 2, where the whole process used is schematized.

First, several data were collected over time through several sensors,  $X_p(t)$ , attached to the equipment. These data subsequently went through a data cleaning phase whose goal was to eliminate/replace everything that decreased the integrity of the set to analyse. After that, they passed to an optimization phase of observable states, with the objective of improving the observations that mapped the hidden states of the HMM. In this way, the optimization of observable states started with a feature creation of the data set, to obtain more information about the data collected from the equipment, since continuous and variable features over time can provide a prediction of possible failures [51]. By comparing the extracted features to the original signal, the feature extraction sought to gather more precise data that increased the accuracy of the performance assessment [1]. In order to generate features and consequently reduce the dimension of the data, a time-window processing method [51] was used. Time windows with different intervals can be created depending on the study to be performed. A feature generation method was applied in the time domain, where several features were created in each of the time windows. The features chosen for the characterization of the equipment failure behaviour were chosen



according to the articles [47,95] as they showed that they related well to the detection of deviations in equipment behaviour.



**Figure 2.** Methodology used to make a diagnosis and prognosis of the equipment through HMM-GRU.

After creating the time-domain features, a feature selection and dimensional reduction process, namely PCA, was used. Here, the aim was to work only with important features, the principal components, as well as to solve the curse of dimensionality by reducing the number of features and thus increasing the speed. From the various existing data reduction methods, PCA-based data compression is the most widely used technique and has superior performance in terms of reconstruction error [58]. From the point of view of equipment health diagnosis, the reduction of class representation speeds up the decision phase [57]. Moreover, as [96] explained, highly correlated features lead to overfitting, and the PCA technique is applied to remove the highly correlated features based on the correlation matrix, thus increasing the prediction accuracy. Without increasing the computational complexity, it considers potential correlations between the answer variables [97], transforming the data into uncorrelated features that assist in converting the data from a high-dimensional to a low-dimensional space, retaining the maximum amount of information [98]. As there were large differences between the ranges of the variables that were provided as inputs to the PCA, a normalisation of the data was first performed. Through normalisation, the amplitude of the initial continuous variables contributes equally to the analysis. Features with larger amplitudes do not overlap features with smaller amplitudes, thus not leading to one-sided results [54]. Therefore, the data were first transformed into comparable scales. This was performed using a *Zscore* normalisation (Equation (9)), where the mean was

subtracted and divided by the standard deviation for each value of each feature, causing all features to be standardized to a zero mean and one standard deviation.

$$Z_{Score} = \frac{x_i - \bar{X}}{std_{dev}(X)} \quad (9)$$

After the normalisation, the data were fed to the PCA that was responsible for transforming the orthogonal orientations moving them in the direction of the bigger variability of the data. These new vectors, known as principal components, can be thought of as new axes that offer the best perspective for visualizing and analysing data in order to make the differences between findings more obvious. The original answer variables are converted into uncorrelated principal components after the original data are mapped to a new vector [97]. It is also a widely used technique when, as is the case with the methodology in this paper, it is necessary to check the clustering trend of the data. PCA, in an effort to preserve all pertinent information, employs projection methods from high-dimensional spaces to lower-dimensional subspaces [99].

Once the PCA phase was over, the new variables, the principal components, were input into the clustering process, which is a multivariate analysis technique to judge the degree of similarity between objects in order to classify them [59,70]. This was done over time in order to understand the clusters that were forming. The clusters were sorted in descending order with cluster 1 having the most data, cluster 2 having the second most data and so on. Cluster 1 was the cluster that appeared most often over time since it was the one with the most data and the last cluster was the rarest. To perform this step, K-means clustering was used. K-means clustering is an unsupervised learning algorithm, used to highlight the intrinsic properties and laws of the data [69]. Among the various existing clustering types, K-means clustering was chosen because it is [65,69] relatively simpler with an easy implementation and fast convergence, it has a strong interpretation ability and it can handle a large number of observations efficiently. According to [100], K-means clustering is the most useful tool for data mining, summarization, probability density estimation and many other essential tasks. Due to the findings' clarity and high scalability, it has recently become one of the most widely used algorithms in data analysis [101]. It is suitable for the reduction of large-scale original failure scenarios [66]. To cluster data, first, it defines the number of clusters,  $k$ , and then the data sets are assigned to each of the  $k$  clusters. The goal is to minimise the square sum of the distances from each data point to its assigned cluster centre. The distance between the data points and the centre of the clusters is calculated in order to assign the data to the nearest clusters. This is repeated for several iterations until convergence is reached [70]. With this analysis, we only had observations optimized by clusters where each cluster represented one observation. These new observations served as the HMM's inputs to train the model and conduct equipment diagnosis. The HMM fitted well with the detection method whose spectrum states were unknown but the receiver could be determined [93]. In this case, the collected and processed observations came from the sensors and the hidden states that represented the health state of the equipment. The objective of the HMM was, through the observable states, to determine which hidden states best applied to each observation over time. To this end, three hidden states were defined that represented the diagnosis of the health state of the equipment: hidden state 1 represents the "State of good functioning"; state 2 represents the "Alert state"; state 3 represents the "State of bad functioning". To use the HMM as a classifier, the HMM parameters whose sizes are determined in advance must be determined by training [90]. Thus, first, the Baum–Welch algorithm was used to train and update the initial parameters of the HMM,  $\lambda$ , that could explain the observation sequence. That is, the HMM parameters were identified such that the probability,  $P(O|\lambda)$ , of the observation sequence  $O$  given the  $\lambda$  model was maximised [92]. Based on the observed state sequence, the HMM training first calculated the maximum probability of the model parameters [91]. After obtaining the parameters of the HMM,  $\lambda$ , finally, the diagnosis of the equipment was performed through the Viterbi algorithm

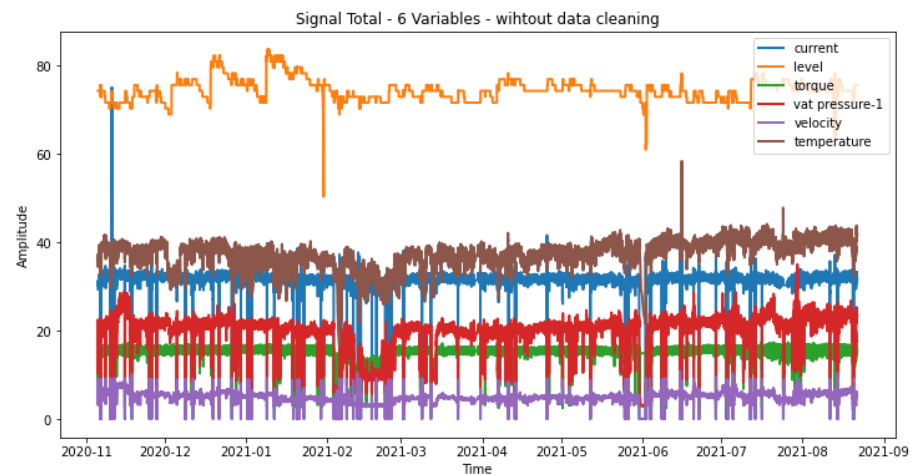
that indicated which hidden states best applied over time. That is, the Viterbi algorithm used dynamic programming to find the maximum likelihood path and finally performed a prediction on the HMM [91]. It is a dynamic programming algorithm to find the most likely sequence of hidden states called the Viterbi path that results in a sequence of observed events [85]. The HMM was used to make the diagnostic classification of the equipment, since it is a model suitable for continuous dynamic signal processing and its function is able to discover the hidden state with a higher probability, through a sequence of observations (in our case, coming from the clustering) [83]. According to the author, in statistical learning theory, an HMM is most efficient in pattern recognition processing. HMMs have often been used for recognising changing behaviours of dynamic features of a system [90], modelling time-series-based phenomena due to their computational efficiency and because they can be used to build data-driven models that provide characteristic indicators [92] and modelling nonstationary and complex random physical processes of machine condition deterioration; the hidden Markov model (HMM) is able to perform both monitoring and diagnosis [102].

Once the diagnosis of the equipment was made, it was possible to move on to a prognosis with the aim of predicting the condition of the equipment in the future, using a DNN algorithm for the time series prediction. According to the literature, deep learning has become an active and promising area of research, and the most widely used deep learning algorithms are the recurrent neural network (RNN), the long short-term memory (LSTM), the convolution neural network (CNN), and the gated recurrent unit (GRU) [81]. In this paper, the GRU recurrent neural network was used, which is a simplification of the LSTM architecture, being able to train more quickly because there are fewer parameters to modify. Furthermore, it can also avoid gradient leakage issues [103]. According to recent research, recurrent units are simpler, necessitating the use of RNNs with smaller memory requirements and less demanding training algorithms. GRU uses the so-called update and reset gate, which are the two vectors that determine what information should be passed to the output, addressing the gradient leakage issue of a standard RNN [81]. According to the author, the GRU's unique quality is that they can be taught to retain information over the long term without forgetting it or deleting information that is unrelated to the prediction. The goal here was to use this neural network to make a prediction of a few days both on the optimised observable states of the K-means clustering and directly on the hidden states of the HMM. In this way, it was possible to see in which of the cases we obtained better values from the GRU model. Since the predictions were made on smaller optimized states, the GRU network was chosen since it showed better results on smaller and less frequent datasets [104,105]. The advantage of GRU cells is that they are as powerful as LSTM cells, even for small datasets [105].

## 4. Case of Study

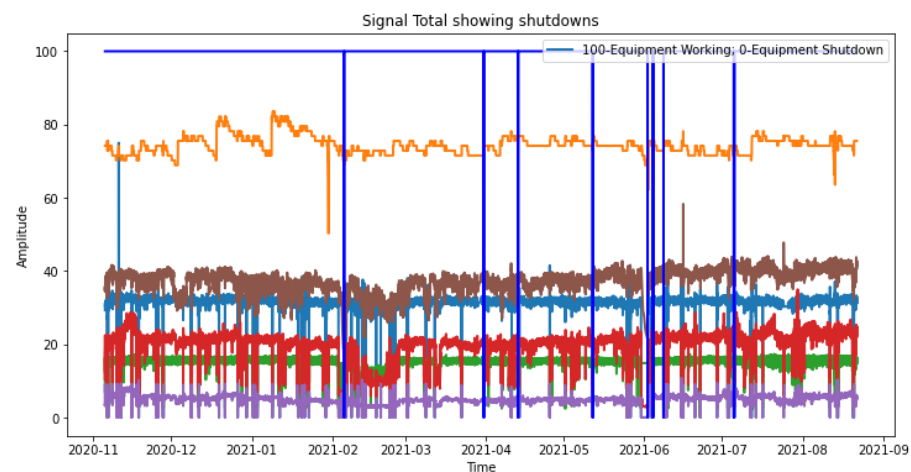
### 4.1. Data Preparation

For this study, we used data collected from a company in the paper industry. More specifically, data were acquired from a pulp-drying press, whose objective is to remove moisture from the pulp. This is a very important process in the production flow, so this equipment needs a type of PdM maintenance. For this, six sensors attached to the equipment were used, collecting observations continuously (every 5 min) over time. Each of the sensors collected values of different magnitudes, as follows: current, hydraulic level, torque, pressure, rotation speed and temperature. With these magnitudes, it was possible to obtain a better picture of the state of the health of the equipment. The data had three years of history, and we used 83,329 data points collected for each of the six sensors (Figure 3).



**Figure 3.** Amplitude of each of the variables under study over time without data preparation.

The quality of the data collected is an aspect to be taken into account, since it may present some flaws in automated data collection processes. This causes inaccurate or incorrect data to be collected. To find meaningful information from big data, it is essential to perform a preprocessing of the data [106]. This step is of utmost importance to ensure reasonable results, whether it is analysis with exploratory data mining, classification or building a good and robust predictive model. Thus, data cleaning was performed to remove data incoherence and increase data integrity. To do this, a program was created that replaced duplicate data, nonexistent data and zeros with the average of their sign. It was chosen not to remove what could be outliers, since they could represent a real malfunction of the equipment and therefore add value to the prediction. We also replaced the equipment stoppages (by the respective average of each signal), since these could be confused with malfunctioning and could reduce the effectiveness of the prediction. A program was created that, when detecting that current, torque, pressure and speed were below a certain threshold at the same time, would be seen as a shutdown of the equipment (Figure 4). The data used for the study are represented in Figure 5.



**Figure 4.** Equipment stoppages over time.

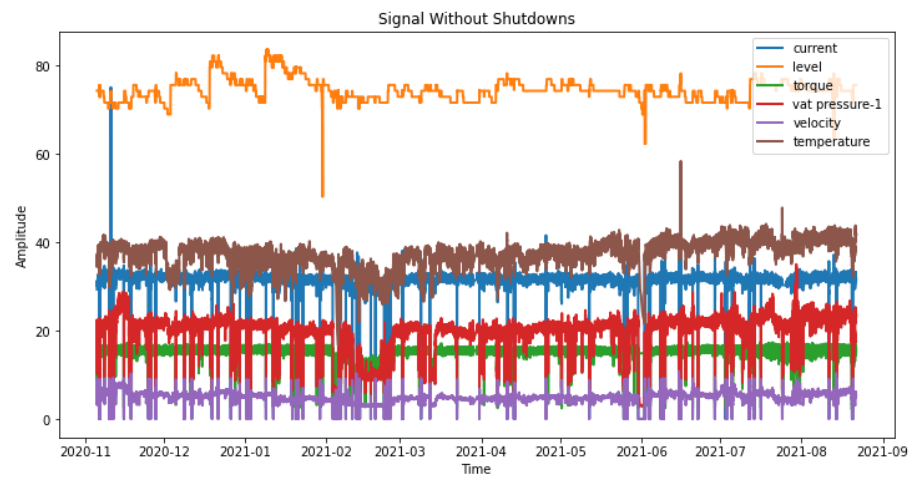


Figure 5. Final signal used for the equipment health status study.

#### 4.2. Feature Generation

After increasing the integrity of the data, they were divided into temporal windows with the aim of creating various time domain features (Table 1).

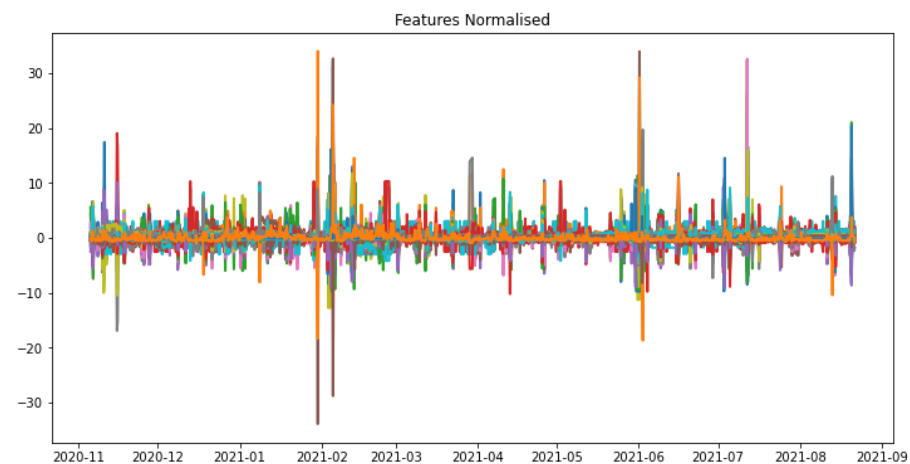
Table 1. Mathematical equations for time domain-based statistical features.

Parameter	Mathematical Equation	Parameter	Mathematical Equation
Mean	$T_1 = \frac{\sum_{n=1}^N x(n)}{N}$	A Factor	$T_{12} = \frac{T_5}{T_2 \cdot T_3}$
Standard deviation	$T_2 = \sqrt{\frac{\sum_{n=1}^N (x(n) - T_1)^2}{N - 1}}$	B factor	$T_{13} = \frac{T_7 \cdot T_8}{T_2}$
Variance	$T_3 = \frac{\sum_{n=1}^N (x(n) - T_1)^2}{N - 1}$	SRM	$T_{14} = \left( \frac{\sum_{n=1}^N \sqrt{ x(n) }}{N} \right)^2$
RMS	$T_4 = \sqrt{\frac{\sum_{n=1}^N (x(n))^2}{N - 1}}$	SRM shape factor	$T_{15} = \frac{T_{14}}{T_1}$
Absolute maximum	$T_5 = \max x(n) $	Latitude factor	$T_{16} = \frac{T_5}{T_{14}}$
Coefficient of skewness	$T_6 = \sqrt{\frac{\sum_{n=1}^N (x(n) - T_1)^3}{(N - 1) \cdot T_2^3}}$	Fifth moment	$T_{17} = \sqrt{\frac{\sum_{n=1}^N (x(n) - T_1)^5}{(N - 1) \cdot T_2^5}}$
Kurtosis	$T_7 = \sqrt{\frac{\sum_{n=1}^N (x(n) - T_1)^4}{(N - 1) \cdot T_2^4}}$	Sixth moment	$T_{18} = \sqrt{\frac{\sum_{n=1}^N (x(n) - T_1)^6}{(N - 1) \cdot T_2^6}}$
Crest factor	$T_8 = \frac{T_5}{T_4}$	Median	$T_{19} = \text{median}x(n)$
Margin factor	$T_9 = \frac{t_5}{T_3}$	Mode	$T_{20} = \text{modex}(n)$
RMS shape factor	$T_{10} = \frac{T_4}{\frac{1}{N} \sum_{n=1}^N  x(n) }$	Minimum	$T_{21} = \text{min}x(n)$
Impulse factor	$T_{11} = \frac{T_5}{\frac{1}{N} \sum_{n=1}^N  x(n) }$		

The data were then divided into 6 h windows, in order to cover four daily operating shifts. In total, 1158 temporal windows were created, where in each one there was a set of 72 data. Then, for each window, 21 characteristics were taken for each one of the six sensors, producing a matrix of  $1158 \times 132$ .

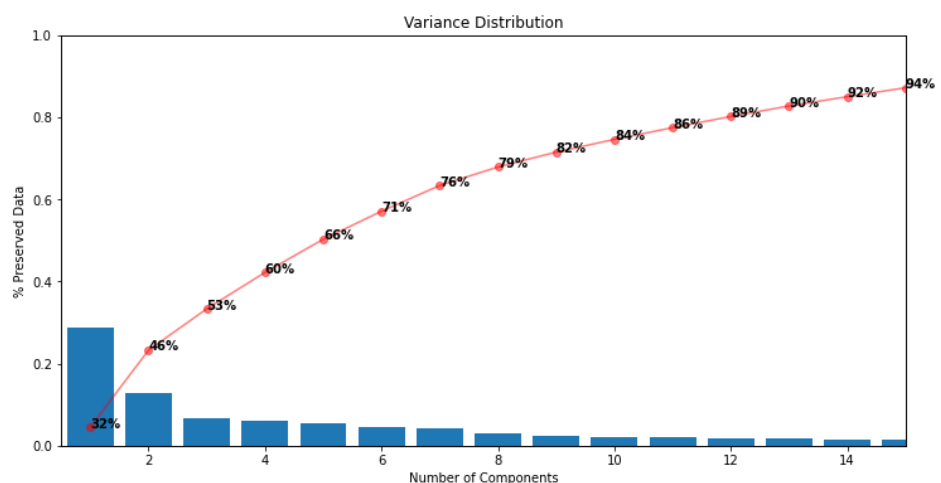
#### 4.3. PCA

The created features went through the PCA method in order to reduce the dimension and also generate new features, the PCs, to use later in the K-means algorithm. In this way, we increased computational speed and worked only with characteristics that were really important for the study, which, although few in number, preserved most of the information. To do this, a *z-score* normalisation of the data was first performed, since they had different amplitudes (Figure 6).



**Figure 6.** All features normalised over time.

The normalised features were then fed into the PCA. Through the study of eigenvectors and eigenvalues, we verified that 10 PCs preserved about 85% of the data variance, as we can see in the Pareto chart represented in Figure 7.



**Figure 7.** Pareto graph with the percentage of preserved information vs. the number of PCs.

#### 4.4. K-Means Clustering

The matrix with the new characteristics, the PCs, resulting from the PCA were sent to the clustering process where the objective was to group the data points that most resembled each other in the same group. To do so, K-means clustering was used. The K-means clustering algorithm followed the following procedure [33,59,65,71,101,107]:

Step 1. Determine the initial number of clusters  $k$ .

Step 2. Randomly select the initial  $k$  centroids  $c_j, j = 1, 2, \dots, k$  in the observations.

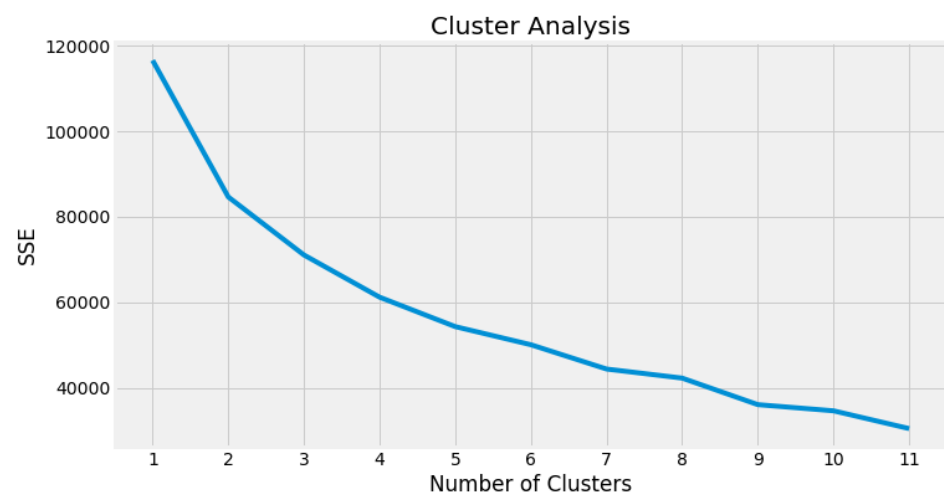
Step 3. Calculate the distance between observation and the initial centroid, and assign the observed object to the cluster closest to the result, using Equation (1)

Step 4. Define a new centroid based on the average of the cluster variables (Equation (10)).

$$c_j = \frac{1}{N_j} \sum_{x_i \in S_j} x_i \quad (10)$$

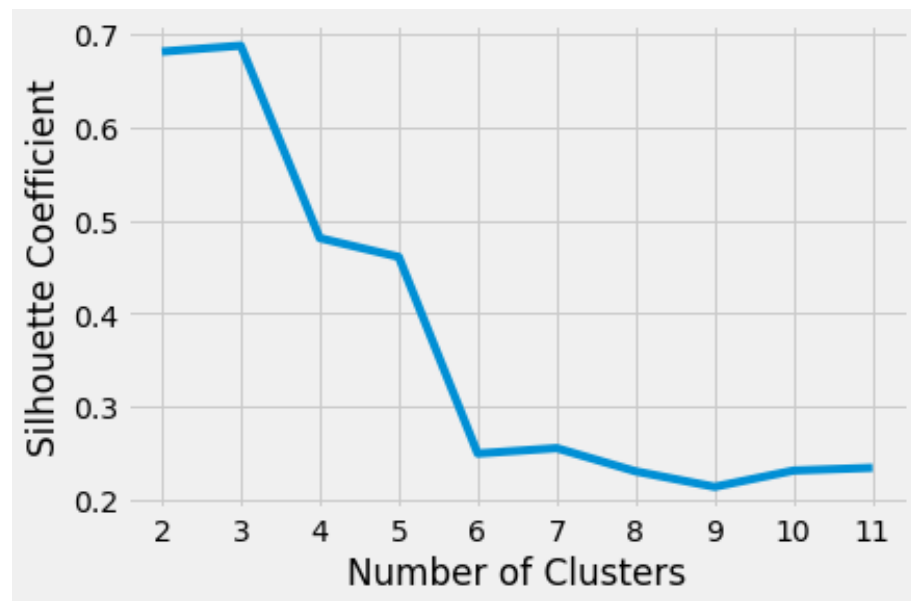
Step 5. Repeat Step 3 using the new centroid until the observed objects are not relocated to another cluster.

This is an iterated process, which iteratively moves the centroids to minimise the total variance within the cluster [67], with two conditions for terminating the iteration [61,65,69]: the specified number of iterations is reached; the cluster centre no longer changes. This paper used the second form. As found in the theoretical framework, to start the K-means method, it is necessary to indicate the number of clusters,  $k$ . This was done using the elbow method, one of the most used methods to select the number of clusters, using for this the error sum of squares (SSE) vs. the number of clusters (Figure 8).



**Figure 8.** Elbow SSE method vs. no. of clusters.

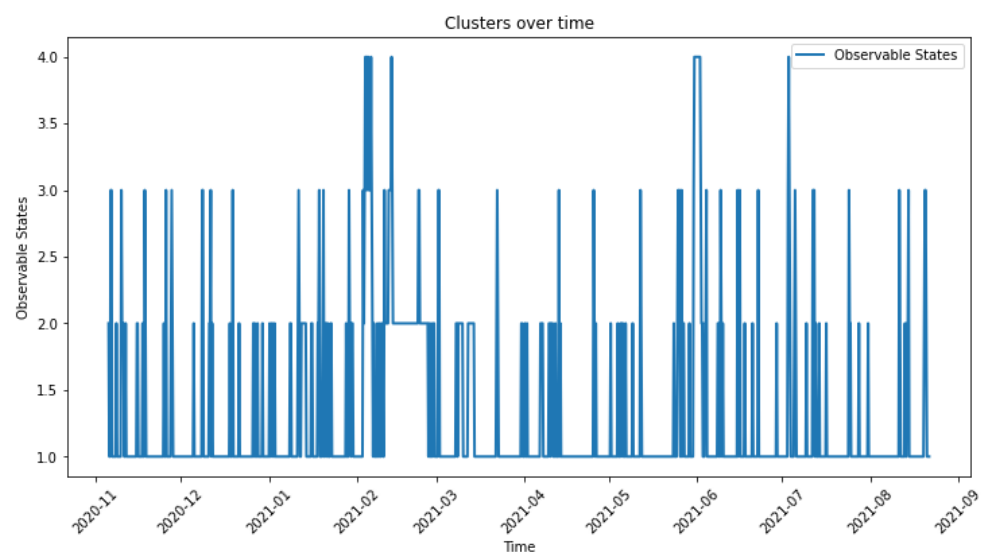
Through the graph of the elbow method, we can see that from  $k = 4$  to  $k = 6$ , an elbow started to be created, where the SSE values decreased more slowly. This showed that the number of clusters increased without significantly improving the SSE value. As the elbow method graph alone was inconclusive, we also conducted a silhouette study (Figure 9) to support the decision of the number of clusters.



**Figure 9.** Silhouette method with silhouette coefficient vs. no. of clusters.

Starting from the elbow graph, where it was unclear which of  $k = 4$ ,  $k = 5$  or  $k = 6$  to choose, through the silhouette graph, we concluded that  $k = 4$  was the optimal choice for this dataset. In this research, the silhouette index was used to evaluate the clustering algorithm and choose the number  $k$  of clusters. The silhouette index is applied in cases of exclusive partitioned clustering and takes into consideration measures of coherence and separation of events in a cluster [108]. The silhouette function calculates the average silhouette coefficient of all samples based on the average intracluster distance and the average distance from the nearest cluster for each sample. Its index ranges from  $[-1, 1]$ , with high silhouette values reflecting good solutions for clustering processes.

Knowing the number of clusters,  $k = 4$ , the k-means clustering was performed and the distribution of the clusters is shown in temporal form (Figure 10). The clusters are sorted in descending order of the number of points, with the first cluster having the most values.



**Figure 10.** Cluster-optimised observations over time.



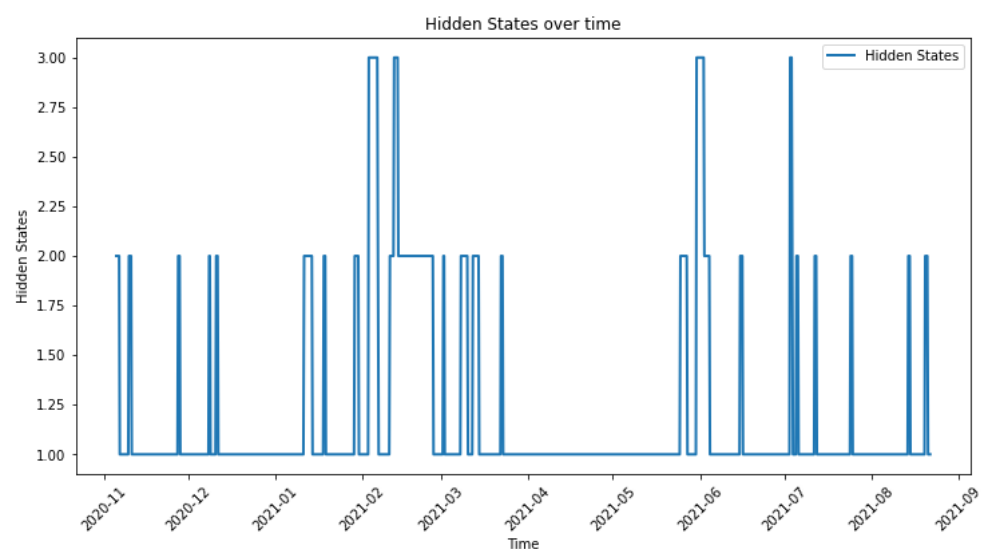
#### 4.5. HMM

After the clustering phase, we applied a classification phase, where the diagnosis of the equipment was made through three hidden states: hidden state 1 represented the “good working state”, state 2 represented the “alert state”, and state 3 was the “malfunctioning state”. Thus, for the HMM, a doubly stochastic method, the observable states were represented by the clusters defined by the K-means clustering and the hidden states were the health state of the equipment. We started by using the observable states to train the model and obtain its parameters. This was performed using the *Baum–Welch* algorithm, which employs a special case of the expectation–maximization algorithm to find local maxima of  $P(O|\lambda)$  [92]. The HMM training first estimated the maximum likelihood of the model parameters based on the observed state sequence. The *Baum–Welch* algorithm was then used to calculate the transition probability  $A = a_{ij}$ , the observation probability  $B = b_{jk}$ , and the initial observation probability  $\pi_i$ , from which the updated algorithm was used to predict the hidden states, and the final algorithm resulted in the prediction of the equipment’s health state. The accuracy was used to evaluate the model. For this, the observations were divided in a temporal manner into training and test data, where 70% were used for training and 30% for testing. The division was made in a temporal manner since the diagnosis reported by the *Viterbi* algorithm represented the hidden states that best fitted the observations over time. After the model was trained with 70% of the data, it generated observable states with the same number of samples as the test data, in order to determine the accuracy (Equation (11)).

$$Accuracy = \frac{\sum (HMM_{GeneratedObservations} = Data_{Test})}{n_{Samples}} * 100 \quad (11)$$

As the parameters of the HMM were based on probabilities, observable states were generated 10,000 times and the accuracy was taken as the average accuracy over the 10,000 runs. For this specific case, an accuracy of approximately 72% was obtained.

Once a good value for the accuracy of the model was obtained, the corresponding parameters were used to determine the sequence of hidden states. For this purpose, the *Viterbi* algorithm was used. The *Viterbi* algorithm finds the most probable sequence of hidden states resulting from the sequence of clusters. Through Figure 11, we can verify the evolution of the health status of the equipment throughout the study time.



**Figure 11.** Hidden states/diagnosis of equipment over time.

#### 4.6. Prognostic with GRU Model

Once the equipment diagnosis was made, the object in this phase was to make a prediction, using the GRU network. The network was prepared to predict 7 days in the future both the observable states that were input into the HMM, as well as the hidden states of the classifier model.

Using a recurrent neural network with an encoder and decoder structure, we made a prediction of the observable states over a period of 7 days into the future, corresponding to one week. Figure 12 presents the prediction of the 7 days with a five-unit GRU recurrent neural network, with a delay window of 3 days. The structure of the network featured a *relu* activation function in the first layer and a *relu* function in the second layer. Figure 13 represents the prediction after processing the data using the function of Equation (12) for a better visualization of the model prediction results. The purpose of that operation was to scale the predicted data,  $x_n$ , to the same values as those of the observable states.

$$ScaleValues = round([x_n - \min(x_n)] * [\frac{n_{States} - 1}{\max(x_n) - \min(x_n)}]) + 1 \quad (12)$$

To validate the accuracy of the model, we used the errors *MAPE* (Equation (13)), *RMSE* (Equation (14)), *MAE* (Equation (15)) and  $R^2$  (Equation (16)), which showed, respectively, the following values of 5.49, 0.33, 0.086 and 0.68, showing a good evaluation of the model.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|x_i - y_i|}{x_i} \quad (13)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (14)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (x_i - y_i)^2}{\sum_{i=1}^n (x_i - \bar{y})^2} \quad (16)$$

where  $x_i$  is the actual value,  $y_i$  is the value predicted by the model, and  $n$  is the total number of observations.

Using the same neural network with the same architecture, changing only the function to a sigmoid in the last layer, with the same delay window, it was possible to perform a prediction of the hidden states for the next 7 days as shown in Figure 14, which, after scaling, persisted, as shown in Figure 15. The *MAPE*, *RMSE*, *MAE* and  $R^2$  errors of the model evaluation had the values 2.79, 0.22, 0.035 and 0.71, respectively.

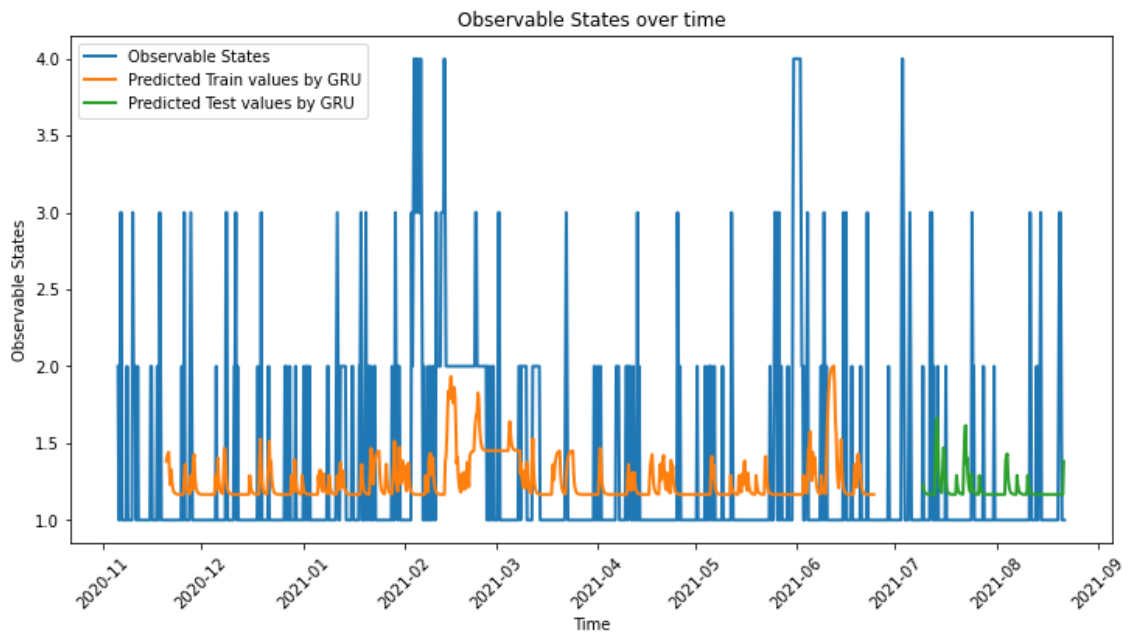


Figure 12. Predicted training and test values for the observable states.

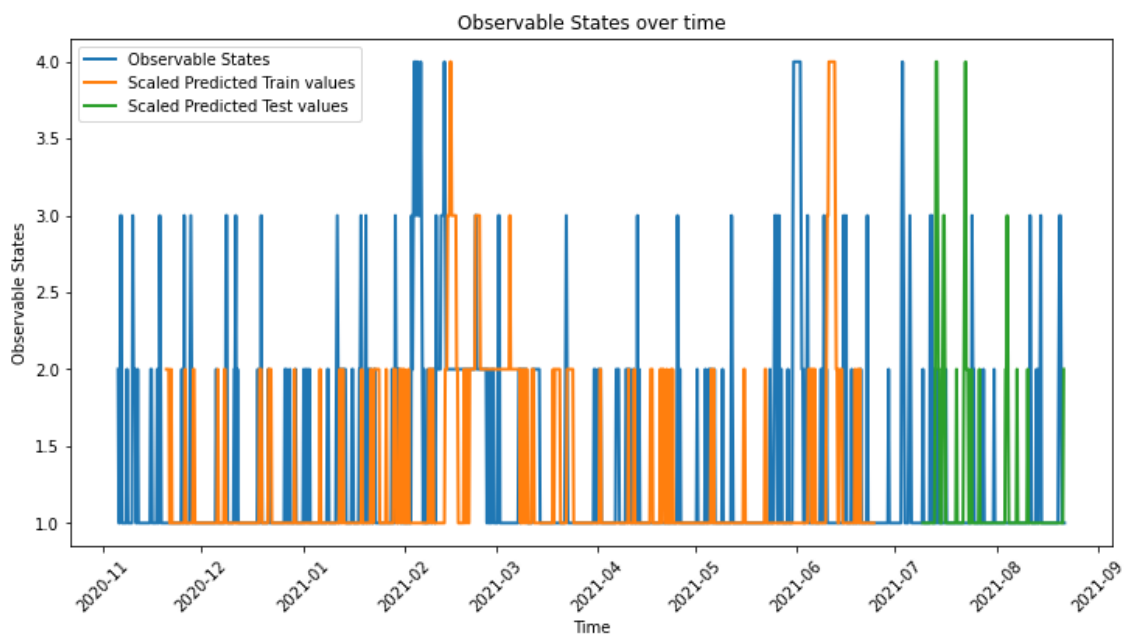


Figure 13. Scaled training and test predicted values for the observable states.

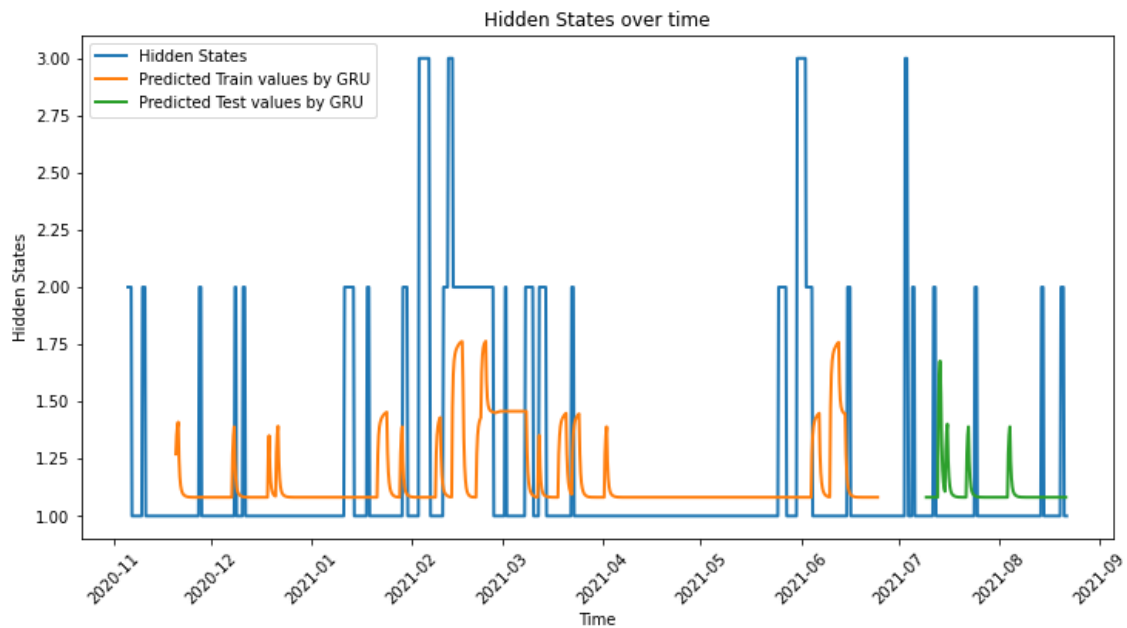


Figure 14. Scaled training and test predicted values for the observable states.

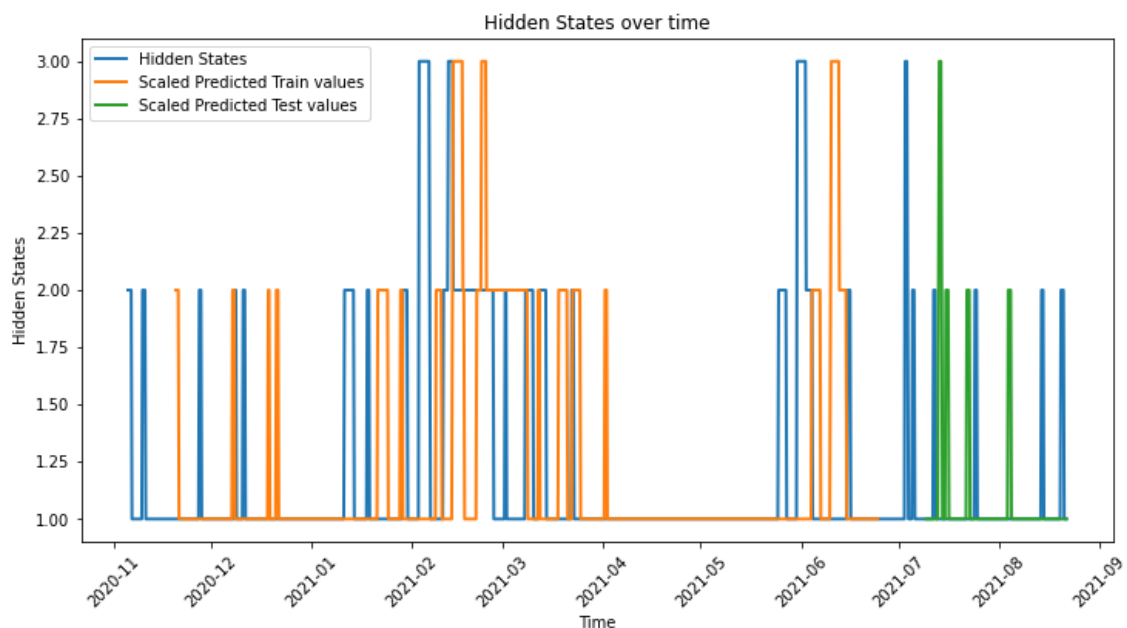


Figure 15. Scaled training and test predicted values for the observable states.

## 5. Discussion

Starting with the diagnostic analysis of the health state of the production equipment, we verified through Figure 11 that the HMM was able to detect with an accuracy close to 72% several states 2 and 3 throughout the study time. We can see in Figure that failure state 3 appeared four times over time, indicating equipment failure. The well-functioning state, state 1, was almost constant, showing that the equipment was mostly in good condition. Several alert states appeared that were quickly solved in order not to reach the equipment failure state.

We also verified through Figures 5 and 11, that state 3, as predicted by the HMM, happened when the values collected by the sensors escaped from the normal pattern of behaviour. This was verified mainly in the periods February 2021 and March 2021, where

we saw that the temperature and pressure sensors had a big drop in their measured values. This indicated that the HMM was able to classify the health status of the equipment without having any previous information about its operation or anomalies. Furthermore, this is a non-supervised methodology, capable of being used in any type of equipment with different types of sensors. It is also capable of performing fault detection in real time, and through the application of a DNN, the GRU, it is also capable of performing equipment prognostics.

The recurrent GRU neural network according to the literature review presents advantages in predicting data of reduced size. This paper showed that the proposed model, a five-unit GRU network, presented a good accuracy of the observed and hidden states of the machine. In this way, it was possible to anticipate a fault detection 7 days ahead with a high degree of accuracy. Normally, the future prediction is done on data collected by sensors, and then it is necessary to run the whole fault-detection methodology. It should be noted that, in our methodology, the prediction was made directly on clusters, optimized observable states, and also directly on hidden states classified by the HMM. In the prediction made on the clusters, it was necessary to use the HMM classifier to obtain the operating states of the equipment. When the prediction was made on the HMM states, we obtained directly the health status of the equipment 7 days ahead. The GRU's capacity to prevent information overlap allows it to perform better with smaller quantities of training data despite having a less complex architecture [103]. In terms of faster computation times and superior results, the GRU also demonstrates the ability to outperform LSTM. Thus, through our methodology, it is possible to directly detect the states of the equipment, through 7-day prognostics, in a faster way, to obtain information in real time and be able to act more quickly with regard to unexpected breakdowns.

## 6. Conclusions

This paper showed a methodology capable of making a diagnosis and prognosis of the state of health of production equipment. First, data cleaning was conducted, followed by the generation of statistical features in the time domain. After the feature generation, a data normalisation and dimensional reduction were performed, through a PCA, to obtain new features with more information and improve compactness. Then, K-means clustering was used to group similar data into groups and create the observable states that were input to the HMM. The HMM, in turn, was responsible for classifying the observable states into hidden states that represented the state of health of the equipment, numbered from one to three, where state 1 represented "Good Operation", state 2 represented "Alert State" and state 3 "Failure State". After the detection of health states over time, it was possible to make a prognosis directly on these states, or on the observable states obtained from the clustering, 7 days ahead. For that, a GRU was used, which obtained good results with these types of data. This is a generic methodology, which can be used in different types of equipment, with different types of sensors. It works without prior information about the behaviour of the equipment and can work in real time.

We conclude that through this methodology, it is possible to improve the quality of CBM and PdM maintenance, thus improving the production flow and consequently, the company's profits.

In future work, other types of models will be applied to each of the steps in order to check which combination obtains the best results. That is, in dimensional reduction, other algorithms will be used, such as linear discriminant analysis (LDA), independent component analysis (ICA), etc.; in the clustering, the Gaussian mixture model (GMM), density-based spatial clustering of applications with noise (DBSCAN), among others. Moreover, the realization of the prediction will use other DNNs that can get better results as well as a prediction in bits through one-hot encode. Other types of statistics that can detect the behaviour of the equipment will also be added. Finally, a classification algorithm will also be implemented to input new data, collected by the sensors, in order to understand in which cluster they best fit. This will lead to the implementation of a metric that will call for

a new training session for the entire methodology if the data disperse significantly from the previously created clusters.

**Author Contributions:** Conceptualization, A.M., J.T.F., I.F. and A.M.C.; methodology, A.M., I.F. and J.T.F.; software, A.M., B.M. and I.F.; validation, A.M., J.T.F., B.M., I.F., M.M., J.R. and A.M.C.; formal analysis, A.M., J.T.F. and I.F.; investigation, A.M., B.M. and I.F.; resources, A.M., J.T.F. and A.M.C.; writing—original draft preparation, A.M.; writing—review and editing, J.T.F., B.M. and I.F.; project administration, J.T.F. and A.M.C.; funding acquisition, J.T.F. and A.M.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** Our research, leading to these results, has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement 871284 project SSHARE and the European Regional Development Fund (ERDF) through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020), under project POCI-01-0145-FEDER-029494, and by national funds through the FCT—Portuguese Foundation for Science and Technology, under projects PTDC/EEI-EEE/29494/2017, UIDB/04131/2020, and UIDP/04131/2020.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Restrictions apply to the availability of these data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

(ANN)	Artificial neural network
(CBM)	Condition-based maintenance
(CM)	Corrective maintenance
(CNN)	Convolution neural networks
(CPS)	Cyberphysics systems
(DBSCAN)	Density-based spatial clustering of applications with noise
(DNN)	Deep neural network
(GMM)	Gaussian mixture model
(GRU)	Gated recurrent unit
(HMM)	Hidden Markov model
(ICA)	Independent component analysis
(IoT)	Internet of things
(LDA)	Linear discriminant analysis
(LSTM)	Long short-term memory
(MAE)	Mean absolute error
(MAPE)	Mean absolute percentage error
(ML)	Machine learning
(PCA)	Principal components analysis
(PCs)	Principal components
(PdM)	Predictive maintenance
(RMSE)	Root-mean-square error
(RUL)	Remaining useful life
(TBM)	Time-based maintenance
(WT)	Wavelet transform

## References

1. Zhang, M.; Amaitik, N.; Wang, Z.; Xu, Y.; Maisuradze, A.; Peschl, M.; Tzovaras, D. Predictive Maintenance for Remanufacturing Based on Hybrid-Driven Remaining Useful Life Prediction. *Appl. Sci.* **2022**, *12*, 3218. [\[CrossRef\]](#)
2. Hu, J.; Sun, Q.; Ye, Z.S. Condition-Based Maintenance Planning for Systems Subject to Dependent Soft and Hard Failures. *IEEE Trans. Reliab.* **2021**, *70*, 1468–1480. [\[CrossRef\]](#)
3. Kumar, P.; Sushil, C.; Basu, K.; Chandra, M. Quantified Risk Ranking Model for Condition-Based Risk and Reliability Centered Maintenance. *J. Inst. Eng. Ser. C* **2017**, *98*, 325–333. [\[CrossRef\]](#)
4. Pais, J.; Raposo, H.D.; Farinha, J.; Cardoso, A.J.M.; Marques, P.A. Optimizing the life cycle of physical assets through an integrated life cycle assessment method. *Energies* **2021**, *14*, 6128. [\[CrossRef\]](#)
5. Huynh, K.T.; Grall, A.; Berenguer, C. A Parametric Predictive Maintenance Decision-Making Framework Considering Improved System Health Prognosis Precision. *IEEE Trans. Reliab.* **2019**, *68*, 375–396. [\[CrossRef\]](#)
6. Chuang, C.; Ningyun, L.U.; Bin, J.; Yin, X. Condition-based maintenance optimization for continuously monitored degrading systems under imperfect maintenance actions. *J. Syst. Eng. Electron.* **2020**, *31*, 841–851. [\[CrossRef\]](#)
7. Zhu, Z.; Xiang, Y. Condition-based maintenance for multi-component systems: Modeling, structural properties, and algorithms. *IIEE Trans.* **2021**, *53*, 88–100. [\[CrossRef\]](#)
8. Koochaki, J.; Bokhorst, J.A.C.; Wortmann, H.; Klingenberg, W. The influence of condition-based maintenance on workforce planning and maintenance scheduling. *Int. J. Prod. Res.* **2013**, *51*, 2339–2351. [\[CrossRef\]](#)
9. Leoni, L.; Carlo, F.D.; Abaei, M.M.; Bahootoroody, A. A hierarchical Bayesian regression framework for enabling online reliability estimation and condition-based maintenance through accelerated testing. *Comput. Ind.* **2022**, *139*, 103645. [\[CrossRef\]](#)
10. Lee, S.M.; Lee, D.; Kim, Y.S. The quality management ecosystem for predictive maintenance in the Industry 4.0 era. *Int. J. Qual. Innov.* **2019**, *5*, 1–11. [\[CrossRef\]](#)
11. Liu, B.; Do, P.; Iung, B.; Xie, M. Stochastic Filtering Approach for Condition-Based Maintenance Considering Sensor Degradation. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 177–190. [\[CrossRef\]](#)
12. Kolhatkar, A.; Pandey, A. Predictive maintenance methodology in sheet metal progressive tooling: A case study. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–10. [\[CrossRef\]](#)
13. Raposo, H.; Farinha, J.T.; Pais, E.; Galar, D. An integrated model for dimensioning the reserve fleet based on the maintenance policy. *WSEAS Trans. Syst. Control.* **2021**, *16*, 43–65. [\[CrossRef\]](#)
14. Zhang, J.; Zhao, X.; Song, Y.; Qiu, Q. Joint optimization of condition-based maintenance and spares inventory for a series—Parallel system with two failure modes. *Comput. Ind. Eng.* **2022**, *168*, 108094. [\[CrossRef\]](#)
15. Soltanali, H.; Khojastehpour, M.; Farinha, T.; Pais, J.E. An Integrated Fuzzy Fault Tree Model with Bayesian Network-Based Maintenance Optimization of Complex Equipment in Automotive Manufacturing. *Energies* **2021**, *14*, 7758. [\[CrossRef\]](#)
16. Li, Y.; Peng, S.; Li, Y.; Jiang, W. A review of condition-based maintenance: Its prognostic and operational aspects. *Front. Eng. Manag.* **2020**, *7*, 323–334. [\[CrossRef\]](#)
17. Oakley, J.L.; Wilson, K.J.; Philipson, P. A condition-based maintenance policy for continuously monitored multi-component systems with economic and stochastic dependence. *Reliab. Eng. Syst. Saf.* **2022**, *222*, 108321. [\[CrossRef\]](#)
18. Shin, M.K.; Jo, W.J.; Cha, H.M.; Lee, S.H. A study on the condition based maintenance evaluation system of smart plant device using convolutional neural network. *J. Mech. Sci. Technol.* **2020**, *34*, 2507–2514. [\[CrossRef\]](#)
19. Peng, S.; Feng, Q.M. Reinforcement learning with Gaussian processes for condition-based maintenance. *Comput. Ind. Eng.* **2021**, *158*, 107321. [\[CrossRef\]](#)
20. Hsu, J.Y.; Wang, Y.F.; Lin, K.C.; Chen, M.Y.; Hsu, J.H.Y. Wind Turbine Fault Diagnosis and Predictive Maintenance Through Statistical Process Control and Machine Learning. *IEEE Access* **2020**, *8*, 23427–23439. [\[CrossRef\]](#)
21. Staden, H.E.V.; Boute, R.N. The effect of multi-sensor data on condition-based maintenance policies. *Eur. J. Oper. Res.* **2021**, *290*, 585–600. [\[CrossRef\]](#)
22. Kenda, M.; Klob, D.; Bra, D. Condition based maintenance of the two-beam laser welding in high volume manufacturing of piezoelectric pressure sensor. *J. Manuf. Syst.* **2021**, *59*, 117–126. [\[CrossRef\]](#)
23. Kumar, G.; Jain, V.; Gandhi, O.P. Availability analysis of mechanical systems with condition-based maintenance using semi-Markov and evaluation of optimal condition monitoring interval. *J. Ind. Eng. Int.* **2018**, *14*, 119–131. [\[CrossRef\]](#)
24. Adu-amankwa, K.; Attia, A.K.A.; Janardhanan, M.N.; Patel, I. A predictive maintenance cost model for CNC SMEs in the era of industry 4.0. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 3567–3587. [\[CrossRef\]](#)
25. Tsuji, K.; Imai, S.; Takao, R.; Kimura, T.; Kondo, H. A machine sound monitoring for predictive maintenance focusing on very low frequency band. *SICE J. Control. Meas. Syst. Integr.* **2021**, *14*, 27–38. [\[CrossRef\]](#)
26. Rodrigues, J.; Martins, A.; Mendes, M.; Farinha, T.; Mateus, R.; Cardoso, A.J. Automatic Risk Assessment for an Industrial Asset Using. *Energies* **2022**, *15*, 9387. [\[CrossRef\]](#)
27. Ingemarsdotter, E.; Lena, M.; Jamsin, E.; Sakao, T.; Balkenende, R. Challenges and solutions in condition-based maintenance implementation—A multiple case study. *J. Clean. Prod.* **2021**, *296*, 126420. [\[CrossRef\]](#)
28. Liu, Y.; Yu, W.; Dillon, T.; Fellow, L.; Rahayu, W.; Li, M. Empowering IoT Predictive Maintenance Solutions With AI: A Distributed System for Manufacturing Plant-Wide Monitoring. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1345–1354. [\[CrossRef\]](#)
29. Lin, C.Y.; Hsieh, Y.M.; Cheng, F.T.; Huang, H.C.; Adnan, M. Time Series Prediction Algorithm for Intelligent Predictive Maintenance. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2807–2814. [\[CrossRef\]](#)

30. Soltanali, H.; Khojastehpour, M.; Pais, J.E.; Farinha, J. Sustainable Food Production: An Intelligent Fault Diagnosis Framework for Analyzing the Risk of Critical Processes. *Sustainability* **2022**, *14*, 1083. [[CrossRef](#)]
31. Ghasemi, A.; Yacout, S.; Ouali, M.S. Optimal condition based maintenance with imperfect information and the proportional hazards model. *Int. J. Prod. Res.* **2007**, *45*, 989–1012. [[CrossRef](#)]
32. Martins, A.B.; Farinha, J.T.; Cardoso, A.M. Calibration and Certification of Industrial Sensors—A Global Review. *WSEAS Trans. Syst. Control.* **2020**, *15*, 394–416. [[CrossRef](#)]
33. Martins, A.; Fonseca, I.; Farinha, T.; Reis, J.; Cardoso, A. Online Monitoring of Sensor Calibration Status to Support Condition-Based Maintenance. *Sensors* **2023**, *23*, 2402. [[CrossRef](#)]
34. Wu, Z.; Guo, B.I.N.; Tian, X.; Zhang, L. A Dynamic Condition-Based Maintenance Model Using Inverse Gaussian Process. *IEEE Access* **2020**, *8*, 104–117. [[CrossRef](#)]
35. Nguyen, K.A.; Do, P.; Grall, A. Condition-based maintenance for multi- component systems using importance measure and predictive information. *Int. J. Syst. Sci. Oper. Logist.* **2014**, *1*, 228–245. [[CrossRef](#)]
36. Zhang, W.; Yang, D.; Wang, H. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. *IEEE Syst. J.* **2019**, *13*, 2213–2227. [[CrossRef](#)]
37. Pais, E.; Farinha, J.T.; Cardoso, A.J.; Raposo, H. Optimizing the life cycle of physical assets—A review. *WSEAS Trans. Syst. Control.* **2020**, *15*, 417–430. [[CrossRef](#)]
38. Yam, R.C.M.; Tse, P.W.; Li, L.; Tu, P. Intelligent Predictive Decision Support System for Condition-Based Maintenance. *Adv. Manuf. Technol.* **2001**, *17*, 383–391. [[CrossRef](#)]
39. Harald, R.; Schj, P.; Wabner, M.; Frie, U. Predictive Maintenance for Synchronizing Maintenance Planning with Production. *Adv. Manuf. Autom. VII* **2018**, *451*, 439–446. [[CrossRef](#)]
40. Popescu, T.D.; Aiordachioaie, D.; Culea-Florescu, A. Basic tools for vibration analysis with applications to predictive maintenance of rotating machines: An overview. *Int. J. Adv. Manuf. Technol.* **2022**, *118*, 2883–2899. [[CrossRef](#)]
41. Chen, C.; Zhu, Z.H.; Member, S.; Shi, J. Dynamic Predictive Maintenance Scheduling Using Deep Learning Ensemble for System Health Prognostics. *IEEE Sens. J.* **2021**, *21*, 26878–26891. [[CrossRef](#)]
42. Kiangala, K.S.; Wang, Z. An Effective Predictive Maintenance Framework for Conveyor Motors Using Dual Time-Series Imaging and Convolutional Neural Network in an Industry 4.0 Environment. *IEEE Access* **2020**, *8*, 121033–121049. [[CrossRef](#)]
43. Mateus, B.C.; Mendes, M.; Farinha, J.T.; Assis, R.; Cardoso, A.M. Comparing LSTM and GRU Models to Predict the Condition of a Pulp Paper Press. *Energies* **2021**, *14*, 6958. [[CrossRef](#)]
44. Antunes, J.; Torres, J.; Mendes, M.; Mateus, R.; Cardoso, A. Short and long forecast to implement predictive maintenance in a pulp industry. *Ekspluat. Niezawodn. Maint. Reliab.* **2022**, *24*, 33–41. [[CrossRef](#)]
45. Mateus, B.C.; Mendes, M.; Farinha, J.T.; Cardoso, A.M. Anticipating future behavior of an industrial press using lstm networks. *Appl. Sci.* **2021**, *11*, 6101. [[CrossRef](#)]
46. Zhang, Z.; Wang, Y.; Wang, K. Intelligent fault diagnosis and prognosis approach for rotating machinery integrating wavelet transform, principal component analysis, and artificial neural networks. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 763–773. [[CrossRef](#)]
47. Martins, A.; Fonseca, I.; Farinha, J.; Reis, J.; Cardoso, A. Maintenance prediction through sensing using hidden markov models—A case study. *Appl. Sci.* **2021**, *11*, 7685. [[CrossRef](#)]
48. Yu, J. Adaptive hidden Markov model-based online learning framework for bearing faulty detection and performance degradation monitoring. *Mech. Syst. Signal Process.* **2017**, *83*, 149–162. [[CrossRef](#)]
49. Arpaia, P.; Cesaro, U.; Chadli, M.; Coppier, H.; De Vito, L.; Esposito, A.; Gargiulo, F.; Pezzetti, M. Fault detection on fluid machinery using Hidden Markov Models. *Measurement* **2020**, *151*, 107126. [[CrossRef](#)]
50. Mateus, B.; Mendes, M.; Farinha, J.; Martins, A.; Cardoso, A. Data Analysis for Predictive Maintenance Using Time Series and Deep Learning Models—A Case Study in a Pulp Paper Industry. In *Proceedings of IncoME-VI and TEPEN 2021*; Springer International Publishing: Cham, Switzerland, 2023; pp. 11–25. [[CrossRef](#)]
51. Chen, H.; Hsu, J.Y.; Hsieh, J.Y.; Hsu, H.Y.; Chang, C.H.; Lin, Y.J. Predictive maintenance of abnormal wind turbine events by using machine learning based on condition monitoring for anomaly detection. *J. Mech. Sci. Technol.* **2021**, *35*, 5323–5333. [[CrossRef](#)]
52. Tipler, S.; Alessio, G.D.; Haute, Q.V.; Parente, A.; Contino, F.; Coussement, A. Predicting octane numbers relying on principal component analysis and artificial neural network. *Comput. Chem. Eng.* **2022**, *161*, 107784. [[CrossRef](#)]
53. Gu, Y.K.; Zhou, X.Q.; Yu, D.P.; Shen, Y.J. Fault diagnosis method of rolling bearing using principal component analysis and support vector machine. *J. Mech. Sci. Technol.* **2018**, *32*, 5079–5088. [[CrossRef](#)]
54. Booker, N.K.; Knights, P.; Gates, J.D.; Clegg, R.E. Applying principal component analysis (PCA) to the selection of forensic analysis methodologies. *Eng. Fail. Anal.* **2022**, *132*, 105937. [[CrossRef](#)]
55. Kamari, A.; Schultz, C. A combined principal component analysis and clustering approach for exploring enormous renovation design spaces. *J. Build. Eng.* **2022**, *48*, 103971. [[CrossRef](#)]
56. Lim, Y.; Kwon, J.; Oh, H.S. Principal component analysis in the wavelet domain. *Pattern Recognit.* **2021**, *119*, 108096. [[CrossRef](#)]
57. Babouri, M.K.; Djebala, A.; Ouelaa, N.; Oudjani, B.; Younes, R. Rolling bearing faults severity classification using a combined approach based on multi-scales principal component analysis and fuzzy technique. *Int. J. Adv. Manuf. Technol.* **2020**, *107*, 4301–4316. [[CrossRef](#)]
58. Zhu, T.; Cheng, X.; Cheng, W.; Tian, Z.; Li, Y. Principal component analysis based data collection for sustainable internet of things enabled Cyber—Physical Systems. *Microprocess. Microsystems* **2022**, *88*, 104032. [[CrossRef](#)]



59. Park, J.H.; Kang, Y.J. Evaluation index for sporty engine sound reflecting evaluators' tastes, developed using K-means cluster analysis. *Int. J. Automot. Technol.* **2020**, *21*, 1379–1389. [[CrossRef](#)]
60. Yang, Y.; Liao, Q.; Wang, J.; Wang, Y. Application of multi-objective particle swarm optimization based on short-term memory and K-means clustering in multi-modal multi-objective optimization. *Eng. Appl. Artif. Intell.* **2022**, *112*, 104866. [[CrossRef](#)]
61. Voronova, L.I.; Voronov, V.; Mohammad, N. Modeling the Clustering of Wireless Sensor Networks Using the K-means Method. In Proceedings of the International Conference on Quality Management, Transport and Information Security, Information Technologies (IT QM IS), Yaroslavl, Russia, 6–10 September 2021; pp. 740–745. [[CrossRef](#)]
62. Sinaga, K.P.; Yang, M.S. Unsupervised K-Means Clustering Algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [[CrossRef](#)]
63. Sinaga, K.P.; Hussain, I.; Yang, M.S. Entropy K-Means Clustering With Feature Reduction Under Unknown Number of Clusters. *IEEE Access* **2021**, *9*, 67736–67751. [[CrossRef](#)]
64. Niño-adan, I.; Landa-torres, I.; Portillo, E.; Manjarres, D. Influence of statistical feature normalisation methods on K-Nearest Neighbours and K-Means in the context of industry 4.0. *Eng. Appl. Artif. Intell.* **2022**, *111*, 104807. [[CrossRef](#)]
65. Lakshmi, K.; Visalakshi, N.K.; Shanthi, S. Data clustering using K-Means based on Crow Search Algorithm. *Sādhanā* **2018**, *43*, 1–12. [[CrossRef](#)]
66. Ni, Y.; Zeng, X.; Liu, Z.; Yu, K.; Xu, P.; Wang, Z.; Zhuo, C.; Huang, Y. Faulty feeder detection of single phase-to-ground fault for distribution networks based on improved K-means power angle clustering analysis. *Int. J. Electr. Power Energy Syst.* **2022**, *142*, 108252. [[CrossRef](#)]
67. Liao, Q.Z.; Xue, L.; Lei, G.; Liu, X.; Sun, S.Y.; Patil, S. Statistical prediction of water flooding performance by K-means clustering and empirical modeling. *Pet. Sci.* **2022**, *19*, 1139–1152. [[CrossRef](#)]
68. Yoo, J.H.; Park, Y.K.; Han, S.S. Predictive Maintenance System for Wafer Transport Robot Using K-Means Algorithm and Neural Network Model. *Electronics* **2022**, *11*, 1324. [[CrossRef](#)]
69. Han, P.; Wang, W.; Shi, Q.; Yue, J. A combined online-learning model with K-means clustering and GRU neural networks for trajectory prediction. *AdHoc Netw.* **2021**, *117*, 102476. [[CrossRef](#)]
70. Chen, L.; Shan, W.; Liu, P. Identification of concrete aggregates using K-means clustering and level set method. *Structures* **2021**, *34*, 2069–2076. [[CrossRef](#)]
71. Visalaxi, S.; Sudalaimuthu, T. Endometrium Phase prediction using K-means Clustering through the link of Diagnosis and procedure. In Proceedings of the 8th International Conference on Signal Processing and Integrated Networks (SPIN), Amity University, Noida, India, 26–27 August 2021; pp. 1178–1181. [[CrossRef](#)]
72. Cho, K.; Merriënboer, V.; Gulcehre, B.; Bahdanau, C.; Bougares, D.; Schwwenk, F.; H. Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734. [[CrossRef](#)]
73. Ke, K.; Hongbin, S.; Chengkang, Z.; Brown, C. Short-term electrical load forecasting method based on stacked auto-encoding and GRU neural network. *Evol. Intell.* **2019**, *12*, 385–394. [[CrossRef](#)]
74. Lin, S.; Fan, R.; Feng, D.; Yang, C.; Wang, Q.; Gao, S. Condition-Based Maintenance for Traction Power Supply Equipment Based on Decision Process. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 175–189. [[CrossRef](#)]
75. Benhaddi, M.; Ouarzazi, J. Multivariate Time Series Forecasting with Dilated Residual Convolutional Neural Networks for Urban Air Quality Prediction. *Arab. J. Sci. Eng.* **2021**, *46*, 3423–3442. [[CrossRef](#)]
76. Liu, X.; Lin, Z.; Feng, Z. Short-term offshore wind speed forecast by seasonal ARIMA—A comparison against GRU and LSTM. *Energy* **2021**, *227*, 120492. [[CrossRef](#)]
77. Gugnani, V.; Kumar, R. Analysis of deep learning approaches for air pollution prediction. *Multimed. Tools Appl.* **2022**, *81*, 6031–6049. [[CrossRef](#)]
78. Veeramsetty, V.; Reddy, K.R.; Arjun, M.S.; Gaurav, M. Short-term electric power load forecasting using random forest and gated recurrent unit. *Electr. Eng.* **2022**, *104*, 307–329. [[CrossRef](#)]
79. Wang, Y.; Zou, R.; Liu, F.; Zhang, L.; Liu, Q. A review of wind speed and wind power forecasting with deep neural networks. *Appl. Energy* **2021**, *304*, 117766. [[CrossRef](#)]
80. Wang, S.; Chen, J.; Wang, H.; Zhang, D. Degradation evaluation of slewing bearing using HMM and improved GRU. *Measurement* **2019**, *146*, 385–395. [[CrossRef](#)]
81. Becerra-rico, J.; Aceves-fernández, M.A.; Esquivel-escalante, K.; Pedraza-ortega, J.C. Airborne particle pollution predictive model using Gated Recurrent Unit (GRU) deep neural networks. *Earth Sci. Inform.* **2020**, *13*, 821–834. [[CrossRef](#)]
82. Zhao, Z.; Wang, Y.; Feng, M.; Peng, G.; Liu, J.; Jason, B.; Tao, Y. Autoregressive State Prediction Model Based on Hidden Markov and the Application. *Wirel. Pers. Commun.* **2018**, *102*, 2403–2416. [[CrossRef](#)]
83. Sun, L.; Li, Y.; Du, H.; Liang, P.; Nian, F. Fault Diagnosis Method of Low Noise Amplifier Based on Support Vector Machine and Hidden Markov Model. *J. Electron. Test.* **2021**, *37*, 215–223. [[CrossRef](#)]
84. Li, Y.; Li, H.; Chen, Z.; Zhu, Y. An Improved Hidden Markov Model for Monitoring the Process with Autocorrelated Observations. *Energies* **2022**, *15*, 1685. [[CrossRef](#)]
85. Jandera, A.; Skovranek, T. Customer Behaviour Hidden Markov Model. *Mathematics* **2022**, *10*, 1230. [[CrossRef](#)]
86. Lin, T.; Wang, M.; Yang, M.; Yang, X. A Hidden Markov Ensemble Algorithm Design for Time Series Analysis. *Sensors* **2022**, *22*, 2950. [[CrossRef](#)] [[PubMed](#)]

87. Yao, Y.; Zhao, X.; Wu, Y.; Zhang, Y.; Rong, J. Clustering driver behavior using dynamic time warping and hidden Markov model. *J. Intell. Transp. Syst.* **2021**, *25*, 249–262. [[CrossRef](#)]
88. Liu, H.; Wang, K.; Li, Y. Hidden Markov Linear Regression Model and Its Parameter Estimation. *IEEE Access* **2020**, *8*, 187037–187042. [[CrossRef](#)]
89. Li, W.; Ji, Y.; Cao, X.; Qi, X. Trip Purpose Identification of Docked Bike-Sharing From IC Card Data Using a Continuous Hidden Markov Model. *IEEE Access* **2020**, *8*, 189598–189613. [[CrossRef](#)]
90. Park, S.; Lim, W.; Sunwoo, M. Robust lane-change recognition based on an adaptive hidden Markov model using measurement uncertainty. *Int. J. Automot. Technol.* **2019**, *20*, 255–263. [[CrossRef](#)]
91. Shang, Z.; Zhang, Y.; Zhang, X.; Zhao, Y.; Cao, Z.; Wang, X. Time Series Anomaly Detection for KPIs Based on Correlation Analysis and HMM. *Appl. Sci.* **2021**, *11*, 1353. [[CrossRef](#)]
92. López, C.; Naranjo, Á.; Lu, S.; Moore, K. Hidden Markov Model based Stochastic Resonance and its Application to Bearing Fault Diagnosis. *J. Sound Vib.* **2022**, *528*, 116890. [[CrossRef](#)]
93. Feng, Y.; Xu, W.; Zhang, Z.; Wang, F. Continuous Hidden Markov Model Based Spectrum Sensing with Estimated SNR for Cognitive UAV Networks. *Sensors* **2022**, *22*, 2620. [[CrossRef](#)]
94. Soleimani, M.; Campean, F.; Neagu, D. Integration of Hidden Markov Modelling and Bayesian Network for fault detection and prediction of complex engineered systems. *Reliab. Eng. Syst. Saf.* **2021**, *215*, 107808. [[CrossRef](#)]
95. Martins, A.; Fonseca, I.; Torres, F.J.; Reis, J.; Cardoso, A.J.M. Prediction Maintenance Based on Vibration Analysis and Deep Learning—A Case Study of a Drying Press Supported on Hidden Markov Model. SSRN. 2022. Available online: <https://ssrn.com/abstract=4194601> (accessed on 3 January 2023).
96. Gokilavani, N.; Bharathi, B. Test case prioritization to examine software for fault detection using PCA extraction and K-means clustering with ranking. *Soft Comput.* **2021**, *25*, 5163–5172. [[CrossRef](#)]
97. Fard, N.; Xu, H.; Fang, Y. A unique solution for principal component analysis-based multi-response optimization problems. *Int. J. Adv. Manuf. Technol.* **2016**, *82*, 697–709. [[CrossRef](#)]
98. Gang, A.; Bajwa, W.U. A linearly convergent algorithm for distributed principal component analysis. *Signal Process.* **2022**, *193*, 108408. [[CrossRef](#)]
99. Sancho, A.; Ribeiro, J.C.; Reis, M.S.; Martins, F.G. Cluster analysis of crude oils with k-means based on their physicochemical properties. *Comput. Chem. Eng.* **2022**, *157*, 107633. [[CrossRef](#)]
100. Reddy, S.; Nethra, S.; Sherer, E.A.; Amritphale, A. Prediction of the number of COVID-19 confirmed cases based on K-means-LSTM. *Array* **2021**, *11*, 100085. [[CrossRef](#)]
101. Li, Y.; Chu, X.; Tian, D.; Feng, J.; Mu, W. Customer segmentation using K-means clustering and the adaptive particle swarm optimization algorithm. *Appl. Soft Comput.* **2021**, *113*, 107924. [[CrossRef](#)]
102. Yang, W.; Chen, L. Machine condition recognition via hidden semi-Markov model. *Comput. Ind. Eng.* **2021**, *158*, 107430. [[CrossRef](#)]
103. Lee, Z.; Ean, W.; Najah, A.; Marlinda, A.; Malek, A. A systematic literature review of deep learning neural network for time series air quality forecasting. *Environ. Sci. Pollut. Res.* **2022**, *29*, 4958–4990. [[CrossRef](#)]
104. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. NIPS 2014 Deep Learning and Representation Learning Workshop. 2014. Available online: <http://xxx.lanl.gov/abs/1412.3555> (accessed on 3 January 2023).
105. Gruber, N.; Jockisch, A. Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text? *Front. Artif. Intell.* **2020**, *3*, 40. [[CrossRef](#)]
106. Cho, W.; Choi, E. Big data pre-processing methods with vehicle driving data using MapReduce techniques. *J. Supercomput.* **2017**, *73*, 3179–3195. [[CrossRef](#)]
107. Guo, J.; Wang, L.; Fukuda, I.; Ikago, K. Data-driven modeling of general damping systems by K-means clustering and two-stage regression. *Mech. Syst. Signal Process.* **2022**, *167*, 108572. [[CrossRef](#)]
108. Ferreira, V.; Pinho, A.; Souza, D.; Rodrigues, B. A New Clustering Approach for Automatic Oscillographic Records Segmentation. *Energies* **2021**, *14*, 6778. [[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.