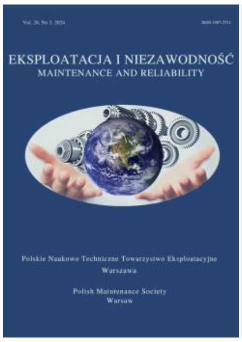

Eksploatacja i Niezawodnosc – Maintenance and Reliability

Volume 26 (2024), Issue 3

journal homepage: http://www.ein.org.pl

Article citation info:
Henriques L, Farinha T, Mendes M, Fault Detection and Prediction for a Wood Chip Screw Conveyor, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2024: 26(3) http://doi.org/10.17531/ein/189323

Fault Detection and Prediction for a Wood Chip Screw Conveyor

Indexed by:

Lucas Henriques[a], Torres Farinha[a,b], Mateus Mendes[a,b,*]

[a] Polytechnic University of Coimbra, Portugal
[b] RCM2+ - Research Centre for Asset Management and Systems Engineering

Highlights

- Detection of faults in a screw conveyor, using different classifiers.

Abstract

Equipment maintenance is a key aspect to maximize its availability. Nowadays, much of the functioning and condition monitoring data from industrial machines is collected through sensors and stored, for off-line analysis. The present work focuses on data analysis of a screw conveyor of a biomass industry. The screw velocity and load were monitored and analysed, in order to detect and predict possible faults. A machine learning approach was used to detect anomalies, where different algorithms were tested with the data available, in order to train an anomaly classifier. The anomaly classifier was able to accurately identify most anomalies, based on historical data, temporal patterns and information of the maintenance interventions performed. The research carried out allowed to conclude that the Extra Trees Classifier algorithm achieved the best performance, among all algorithms tested, with 0.7974 F-score in the test set. The anomaly classifier has been shown to achieve remarkable accuracy in identifying anomalies. This research not only improves understanding of the performance of screw conveyors in biomass industries, but also highlights the practical utility of employing machine learning for proactive fault detection.

Keywords
industrial equipment, machine learning, predictive maintenance, screw conveyor

1. Introduction

The biomass industry relies on heavy equipment and requires adequate maintenance for reliability and safety.

1.1. Biomass Energy

Biomass is one of the most important and sustainable forms of renewable energy. A dry ton of wood can generate up to 1 MWh of electricity in an efficient power plant. Modern biomass power plants are complex and mostly automated systems, where the raw material is moved and transformed along the plant with minimum human intervention. The process is efficient and safe. However, it poses important challenges in terms of automation, monitoring and equipment maintenance, for smooth and safe continuous operation. The plant that is object of the present study aims to operate continuously, with just one stop for maintenance every year.

1.2. Screw conveyor maintenance

The present paper is about fault detection and prediction of a screw conveyor. This screw conveyor is one of two large screws feeding wood chips to a large plant that produces paper

(*) Corresponding author.
E-mail addresses: L. Henriques, a21260214@isec.pt, T. Farinha (ORCID: 0000-0002-9694-8079) tfarinha@isec.pt, M. Mendes (ORCID: 0000-0003-4313-7966) mmendes@isec.pt

pulp and electricity. The screws operate in a silo with variable load, and they feed the wood chips for further processing in the plant. They are a critical component of the plant, as their failure causes interruption in the supply of all of the systems ahead.

Sensors were installed to monitor the load that is deposited on the screw conveyor and its velocity. Through the data coming from those sensors, the behavior and condition of the screw conveyor are screened and potential failures may be predicted before they happen.

The main objective of this paper is to develop a model for identifying anomalies in the screw conveyor operation, using Machine Learning to extract patterns from past data. An anomaly classifier must be able to accurately identify anomalies in the industrial machine based on historical data, temporal patterns and maintenance procedures. The objective is to understand whether the models developed are capable of identifying anomalies and whether they bring benefits to the screw conveyor predictive maintenance industry. The models developed could be very useful to avoid serious damage to the screw conveyor and production losses. The objective is to improve and optimize production efficiency, reducing resource consumption and increasing productivity. This article innovates through the introduction of temporal window analysis and fault prediction for screw conveyors. This involves a detailed examination of time-based patterns in operational data, enabling a consistent and near real-time understanding of system behavior. By incorporating the extraction and classification of these temporal windows, our model aims to provide a comprehensive view of the dynamics of screw conveyor operation and condition.

1.3. Paper structure

The article consists of the following sections. In Section 2 the different types of maintenance are described and predictive maintenance is discussed in more detail to contextualize the topic of the work. Section 3 presents the state of the art, addresses various approaches to solving problems in other similar industrial areas and presents a comparison of results from the articles analyzed. In Section 4, the screw conveyor is presented and a detailed description is given. Section 5 presents the dataset and the type of variables that it contains. Section 6 explains the entire preprocessing process applied and describes in detail all the techniques applied to clean and transform the data set. Section 8 shows the types of maintenance and explains how the extraction of time windows for classifying anomalies works. In section 9, the training process of various machine learning models, and their performance, are explained.

2. Predictive Maintenance

The industrial environment is constantly changing over the years. The aim is always to solve problems by applying new technologies which can improve safety or quality, or just lower the costs [35]. An area with great influence in recent years has been artificial intelligence, namely through the application of machine learning techniques, as it can imitate intelligent human behavior for decision making.

Unexpected failures in the industrial context can endanger property or even people. They can also cause production faults for several hours, days, or even weeks. That can, therefore, cause incalculable losses. In order to avoid or at least minimize such losses, predictive maintenance aims to achieve good forecasts of future equipment behaviours, based on historical data of the industrial equipment.

Maintenance strategies can be divided into Corrective, Preventive and Predictive [31,24,17]. Corrective maintenance is reactive, the corrective repair per- formed after the machine fails. This approach is not acceptable for critical equipment, such as the screw conveyor in the present study.

In preventive maintenance, there is a maintenance schedule for each equipment. This schedule can have different time intervals, depending on the manufacturers and the equipment in question. This type of maintenance tends to be more economical because it prevents equipment from reaching total wear and tear and deteriorating surrounding items of an industrial machine. However, it often leads to situations where interventions may be too soon or too late, because the conditions of operation of the equipment do vary over time.

Predictive maintenance uses data analysis methods in order to predict the behavior of the equipment. There are numerous variables that can be monitored through sensors, and the data collected must be analyzed for prediction purposes. These sensors can be, for example, pressure, humidity, vibration, proximity, gas, temperature, oil levels or velocity, among many others [32]. In a maintenance focused on prediction, it is not intended that the components reach maximum wear, but a wear that is cost-effective and still safe to keep the machine in production as long as possible, without unnecessary maintenance interventions that could cause downtime and early waste of materials.

3. State of the Art

Predictive maintenance through sensor data analysis has proven to be an innovative and efficient approach to ensure the operational integrity of industrial equipment. In this state of the art, we have surveyed and studied the most outstanding scientific and technological contributions, carefully choosing articles that stand out in solving predictive maintenance problems and sensor analysis. The selection of these articles was based on the relevance that each one presents to solve various problems, focusing on solutions that address specific challenges found in manufacturing environments.

Kanawaday *et al.* [33] collected sensory data from production cycles of a slitting machine to be used in machine learning algorithms. Sensory data involves vari- ables such as: timestamp, voltage, pressure, width and diameter. Pre-processing was carried out, outliers were detected through clustering and graphical visualizations were made to help separate good and bad production cycles. They use a predictive model called ARIMA (Autoregressive Integrated Moving Average) in order to generate forecast data for new production cycles. The predicted data is used in supervised models to find fault corrections. Among the supervised models, the one that obtained the best result was Deep Neural Network with 98.69% accuracy.

Becherer *et al.* [34] show an approach capable of predicting machine main- tenance. They identify that the quality of the data and the chosen sensors are essential for prediction. They also say that data from sensors with information such as temperature, engine velocity and acoustic signals are very important for finding anomalies. Fault records and other relevant data, such as quality assur- ance, must be well identified to help predict maintenance. The article shows an approach capable of predicting machine maintenance through Decision Trees and some important features. The most important feature focuses on a “lifetime”. Over time, if the “lifetime” feature reaches an alarming value, the model is able to decide the need for maintenance. If the features do not return to normal standards, the equipment is halted and maintenance is triggered.

Geca *et al.* [15] used 24 hour time windows to predict failures and labeled the data to solve a multi-class classification problem. The different classes are divided into four components for failure and another for normal operation of the machine. The article identifies a complete table of attributes and their values, component change days, age, errors and model type. Gradient Boosting, Decision Tree and Artificial Neural Networks (ANN) obtained the three best prediction results. Confusion matrices were developed to evaluate predicted false alarms, situations in which a failure is predicted but actually does not happen. ROC curves help to understand which component failures are detected effectively. Of the three algorithms chosen, all obtained very attractive results, but gradient boosting was the one that obtained the best results. It achieved an accuracy of 99.92%, gave a false alarm 34 times and failed to detect 39 failures in over 122,000 test data.

Leukel *et al.* [37] present a literature review. They analyze which supervised machine learning algorithms are most used, which ones obtained better results, which are the reliable metrics and best practices to follow to obtain results in maintenance predictions. In a total of 34 articles analyzed, the most adopted models were Random Forest, Support Vector Machine, Artificial Neural Net- work and Decision Trees. It was found that many of the articles used metrics that can be unbalanced when analyzing predictions. This is due to the small datasets in which maintenance failures occurred, making it difficult to predict these events, since they happened few times and the dataset is not balanced. Another important aspect suggested by the author was the use of adjustable cross-validation for each dataset size.

Strauss *et al.* [26] report a case study where predictive maintenance is carried out on a BMW group electric elevator system. The authors use low-cost sensors, due to the fact that the study finds that 75% of the companies would not be willing to invest more than 500 euros to digitize their machines. They used temperature and vibration sensors, configured a 5 GHz wireless connection with a connection to an IoT cloud. They found that 1.2 GB of data was generated per hour in a set of 480 sensors. They identified some anomalies which facilitated the identification of outliers. Short time slots were created to facilitate the extraction of features such as minima, maxima, mean, standard deviation, variance and gradients. They used supervised and unsupervised algorithms, but the best results were obtained with supervised algorithms. In the supervised algorithms, the Random Forest had the best performance, reaching an F1-score of 98.1%. Decision Trees and k-Nearest Neighbors achieved very similar performance. Decision Trees, Random Forests and SVMs processed the score very quickly, while the Logistic Regression Algorithm and k-Nearest Neighbors were relatively slow.

Rodrigues *et al.* [18] describe a case study where supervised and unsupervised learning methods were used to create models for predicting the condition of an industrial paper press. The main objective was to determine when the asset was either in its nominal operation or working outside this zone, thus being at risk of failure or sub-optimal operation. The article explains that the equipment's states of operation were determined using a clustering algorithm, such as k-means, and that a lubricant classification algorithm was developed using neural networks. The lubricant classifier results were 98% accurate compared to human expert classifications.

A comparison of the results of the analyzed articles can be observed in Table 1. Leukel *et al.* [37] analyze several articles and have a different notation to show the results. For each algorithm, the minimum and maximum accuracy obtained in the analyzed articles are shown. For this reason, the results are also not shown in the table, but it is important to emphasize that the study is important to understand which algorithms are most used in this type of problem. Becherer *et al.* [34] do not indicate results—that is a paper describing an approach capable of solving anomaly detection problems without testing, and for this reason the results are not in the Table.

Table 1. Summary of relevant works for fault detection.

Article	Author	Year	Model	Results
[33]	Kanawaday *et al.*	2017	Naive Bayes	Accuracy 96.61%
			Suport Vector Machine	Accuracy 95.52%
			CART	Accuracy 94.46 %
			Deep Neural Network	Accuracy 98.69%
[15]	Geca *et al.*	2020	Gradient Boosting	Accuracy 99.92 % F1-Score 97.76 % Precision 98.02 % Recall 97.52 %
			Random Forest	Accuracy 99.93 % F1-Score 97.92 % Precision 98.74 % Recall 97.14 %
			Artificial Neural Network	Accuracy 99.88 % F1-Score 96.82 % Precision 96.04 % Recall 97.14 %
[26]	Strauss *et al.*	2018	Random Forest	F1-Score 98.1%
			Decision Tree	F1-Score 97.3%
			K-Nearest Neighbors	F1-Score 97.4%
[18]	Rodrigues *et al.*	2022	Neural Network	Accuracy 98%

From the articles analyzed, there are very good results for fault detection and prediction. However, the results are always specific of each equipment and situation. Overall, many good results were obtained and in the articles references were made to the use of competent metrics on imbalanced data sets.

In the present work, the objective is to apply the best algorithms for classification, such as Support Vector Machine, Gradient Boosting, Random Forest and Decision Tree.

The scientific contributions highlighted in this review provide a valuable starting point, offering insights, methodologies and approaches that can be adapted to the screw conveyors industrial context.

Despite the efforts of the academic and industrial community to integrate predictive maintenance with sensor analysis, there is a pressing need for more in-depth investigations into implementation strategies in specific industrial environments, such as screw conveyors. Many of the studies analyzed offer valuable information, but direct application in specific industrial contexts still lacks a more detailed approach. The central motivation for this research lies in the lack of screw conveyor research in the areas of predictive maintenance and critical industrial processes. This study aims to contribute to a solid approach about predictive maintenance strategies, thus optimizing the reliability and operational efficiency of screw conveyors in diverse industrial environments, namely the biomass industry where screw conveyors are critical components whose failure may lead to halt the power plant.

4. Screw Conveyor

The screw conveyor was invented by Archimedes in the years 235–240 B.C. It is a helical blade, usually in a tube, to move powdered, granulated products or even liquids [29]. Screw conveyors are made of a wide range of different types of materials that can be steel, stainless steel variants, plastic or other materials.

Screw Conveyors have as main objective to supply silos, carry out loading and unloading of materials or even supply other types of industrial machines. As a rule, they are widely used in the food, chemical and pharmaceutical industries [30,12], among others.

The product to be transported is placed on the feeding side and the movement of the helical blade is responsible for making the material to reach its destination. Screw conveyors have the advantage of ease of maintenance, adaptability to each location, resistance to corrosion or heat in some situations and can even be sealed to prevent the release of dust or gases [7,9].

Screw conveyors take up little space, have a firm structure and are easy to install, allow for high transport speeds and high safety performance. Using these conveyors, it is possible to avoid moving material manually, increasing the productivity of industries, the fundamental role in production systems, minimizing human error, reducing risks in the workplace and reducing associated costs, among many other benefits.

Figure 1 shows a picture of a screw conveyor like the one monitored for the dataset used in this article. The screw conveyor is connected to an apparatus with an electric a motor, which can be seen in Figure 2. There is also a Shaftless Helical Conveyor in order for the material to flow through the auger as well as around it. This type of conveyor is widely used in mining, wastewater and chemical processing and has the advantage of increasing efficiency, flexibility and ease of transporting complex products.

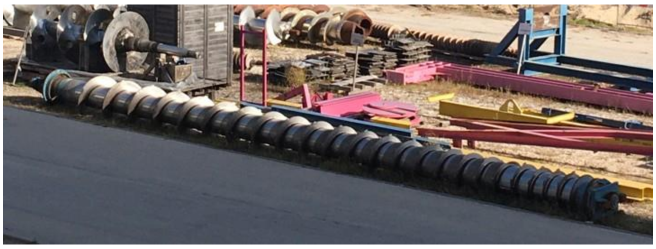

Fig. 1. Picture of a screw Conveyor like the one used for the present paper.

4.1. Screw Conveyor Capacity

The capacity of a screw conveyor can be calculated from the speed, width and angle of the belt, together with the density of the material to be conveyed.

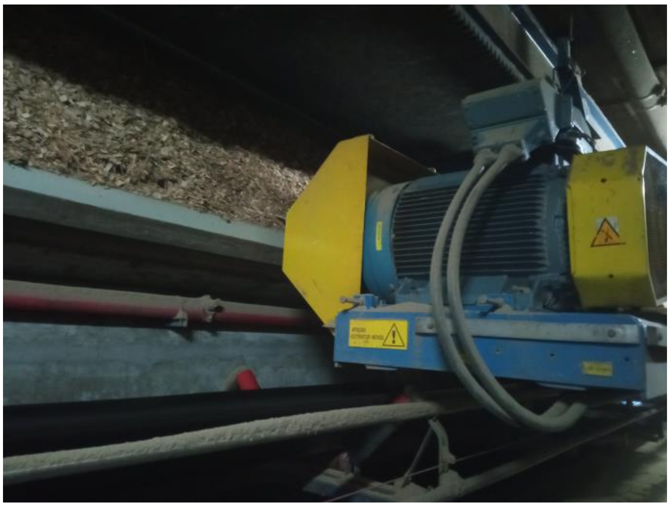

Fig. 2. Apparatus where the screw Conveyor is operating.

The actual output of a screw conveyor is considerably less than the theoretical output. The volumetric efficiency is defined as η_v [25] and is calculated as in Equation 1.

$$\eta_v = \frac{Q_a}{T_t} \quad (1)$$

In the equation, Q_a is the actual volumetric output in m^3/min and Q_t is theoretical volumetric output, also in m^3/min. Q_t is calculated using Equation 2.

$$Q_t = \frac{\pi}{4}(d_{sf}^2 - d_{ss}^2)l_p n \quad (2)$$

where d_{sf} is screw flight diameter (m), d_{ss} is screw shaft diameter (m), lp is the pitch length (m) and n is the screw rotational speed in rotations per minute ($r.p.m.$).

4.2. Screw Conveyor Placements

The screw conveyors can have up to three possible placements, which can be horizontal, inclined and vertical [28][27]. Horizontal positioning requires less effort on the part of the engine to get the product to its destination, which is not the case with inclined and vertical positioning. The larger the degree of inclination, the lower the efficiency of the screw conveyor and the amount of material that flows through it. Therefore, to maximize efficiency and reduce engine power, the angle of inclination must be as low as possible.

4.3. Common Problems

There are some common problems with screw conveyors [14,6,5]. If the input material is too large, it can result in a very high engine load, causing the engine to heat up and burn out. Another common case is the existence of friction between the screw blade and the tube wall. If the distance is not adequate, it can make a lot of noise and the engine can burn out. The material to be transported can overflow and that usually happens when there is water surrounding it. It depends on the product to be transported, but it happens in dry and refined products, which mixed with water makes a thick mixture with the possibility of blocking the bearings . Other problems can be the placement of scrapers, redoing the lubrication tubes, retainers, repairing welds, replacing belts, chains, rubbers and screws.

5. Dataset

Temporal values of one screw conveyor with 1-minute period records and the respective sensory variables were the object of study for the present paper. The sensors installed on the screw conveyor make it possible to measure the load of material to be introduced into the screw conveyor, in percentage of maximum load, and the velocity of the screw conveyor in rotations per minute.

The dataset consists of samples recorded between May 2015 and March 2022. Through an also associated report, it is possible to identify the respective scheduled maintenance and faults that occurred over time. The data were received in spreadsheets and were loaded into Python dataframes for data analysis.

Table 2 shows the maintenance interventions whose need is aimed to be predicted. The maintenance events in Table 2 were identified through a maintenance log report provided by the maintenance team. This report provides the name of the maintenance, date and time from the start of the maintenance to its completion. The data set is limited in some of the anomalies (Table 2). One of the cases is the maintenance operation "Repair of Moving Spheres" as they only have one record during the entire period of operation monitored. Due to this limitation and as there are other classes with few examples, it is necessary to evaluate the classification of machine learning models. Due to the limited number of records, this article approach aims to detect just the need to have a maintenance operation in general, rather than predicting specific maintenance events.

Table 2. Different types of maintenance and the number of events present in the dataset.

Procedure	# Events
Scraper Repair	7
Screw Repair	3
Belt Repair	2
Repair of Moving Spheres	1

6. Preprocessing

Preprocessing refers to the various steps required to clean and transform raw data into a format suitable for further analysis [23]. In Python, preprocessing typically involves importing and loading data, cleaning and organizing the data to remove missing or incorrect values, and transforming the data into a format suitable for analysis. It may involve converting data into numeric formats, creating new variables, normalizing the data and handling outliers. It is an important step in the data analysis process, as it helps ensure that the data are in a consistent and usable format, for subsequent algorithm submissions.

Missing values, null values, and even duplicate values were checked. No duplicate values were found in the dataset used, while missing and discrepant data were noticed.

The total dataset has 5,325,192 records. Some of them were found to be null or missing values, as can be seen in Fig 3. In the load variable, there are some records of null values, while in the velocity variable, there are no null values. Missing values and null values are around 5% of the total dataset.

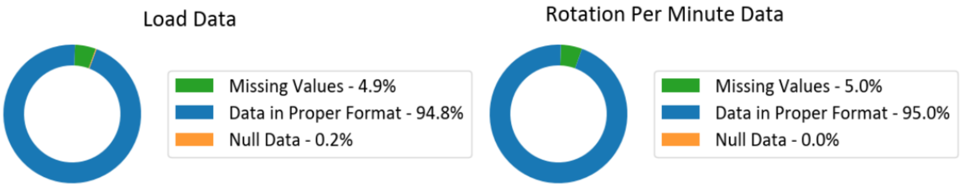

Fig. 3. Amount of missing data in variables Load and Velocity.

After these steps, each variable was converted to the closest data type to facilitate a better perception of the data. In this case, the date was converted to a year-month-day hour:minute:second format, and the remaining variables were converted to floating point numbers, since they represent continuous values. The indicated date format corresponds to the ISO standard and commonly used approaches to specifying date formats.

6.1. Discrepant Data

When performing this conversion operation to float, a conversion error occurred because there were data samples corrupted with non-ASCII characters in some temporal events. In order to deal with these non-ASCII character values, the value "NaN" was assigned, which means Not A Number. We chose to assign "NaN" because in a next step an interpolation method was used and there is a python function that already works with this type of data.

Before moving on to the next stage, it was necessary to understand the reason for the presence of these discrepant data and it was concluded that the screw converters could suffer from jamming or even some stops for lubrication of the machines by the operators. These stop events mean that the sensors do not record values at certain points in time.

With the data in the right formats, it is possible to have a better understanding of the signals involved. It is also possible to analyze the signal of the variables over a period of a week, month, or even over a year. Some graphs were created and they show that at some periods of short minutes, the sensory data exhibit some sudden interruptions. These interruption events can be caused by voltage spikes and they are recognized as discrepant samples, as shown in Figure 4.

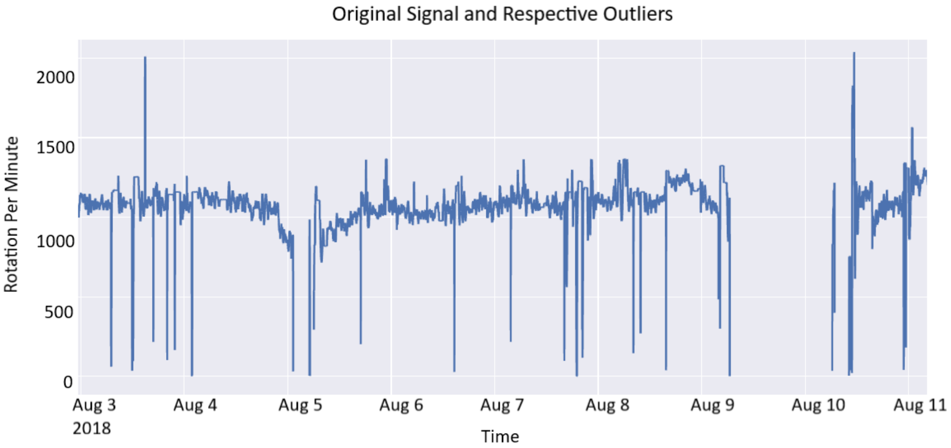

Fig. 4. Plot of the original velocity signal, showing some missing and discrepant data.

6.2. Processing Discrepant Samples

As temporal and sequential events are important for analyzing machine behavior and predicting the need of maintenance, it is important to treat missing data and discrepant samples. Discrepant data were treated using an approach with the z score statistical measure. In this approach, z can be positive, negative, or zero, depending on how far the x_i (variable instant value) is from the mean [3]. The z score represents the number of standard deviations x_i is away from the mean of a distribution. It is used to identify how typical or unusual a given value is, within a given data set. A positive z score indicates that x_i is above the mean, a negative z score indicates that x_i is below the mean, and a zero z score indicates that x_i is equal to the mean. After getting the mean and standard deviation, Equation 3 is used to calculate the z score for a given x_i value:

$$z_i = \frac{x_i - \mu}{\sigma} \qquad (3)$$

where x_i is the value being evaluated, μ is the mean of the distribution and σ is the standard deviation.

The z score interval is used to find the discrepant samples and those samples are assigned "NaN." As previously mentioned, the interpolation method used is done through a function that replaces "NaN" values.

Figure 5 shows a plot of the original data, where the discrepant data (outliers) were eliminated, using the z score method with z = 2, and the interpolation effect. The z score is visible through the red lines that represent the lower and upper limits.

Interpolation is the process of estimating a value within the range of a set of known data points. It is often used in computer graphics to smooth out or fill in gaps in data. There are many different interpolation methods that can be used, such as linear interpolation, polynomial interpolation and spline interpolation [22].

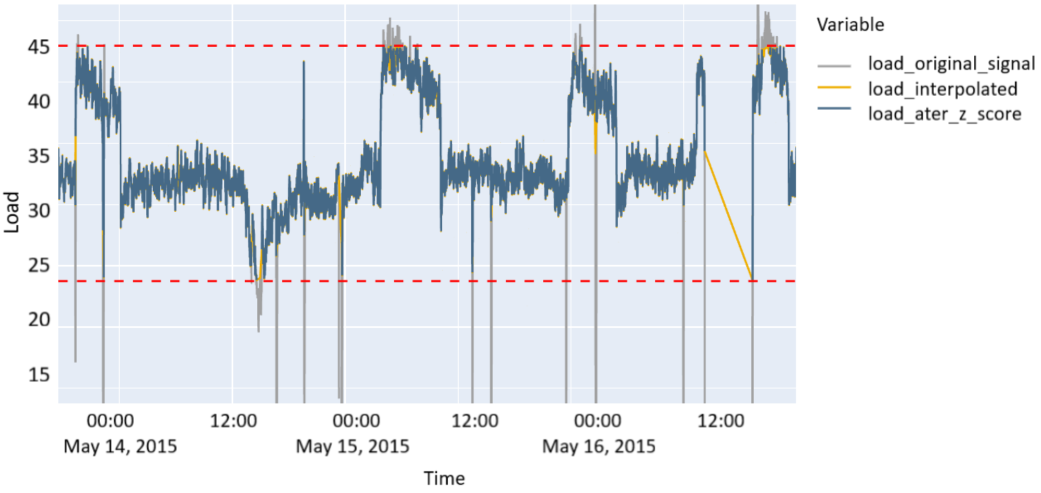

Fig. 5. Variable Load after removal of discrepant samples and interpolation.

Figures 6 and 7 show the number of dataset values that were out of the z score interval and were, therefore, considered discrepant data. All values out of the interval were assigned "NaN" because they are too far from the mean and are assumed to be outliers.

The number of samples to be estimated by interpolation increased after using the z score, as shown in Fig 6. Filtration through the z score limits added 324,667 "NaN" to the load variable, while in the velocity variable, 298,468 were added. The number of samples that were missing or were cut out and therefore had to be interpolated in the load variable was 343,967 and in the velocity variable, it was 1,976,133. It is important to mention that the z score limits process were established month by month with a z score equal to 2. The choice of z_score with z = 2 was made with the aim of finding a balance between sensitivity and robustness. If the z value was too low, it would eliminate many points, including some points that would be valid for pattern analysis, and if the z value was too high, it might not adequately remove outliers as shown in Figure 5.

Table 3 shows that after cleaning of discrepant data and interpolation, the standard deviation of the dataset decreased. This reduction is verified because z score limits were imposed, which makes the data more condensed and not so dispersed due to the effect of discrepant samples. The mean slightly increases, and this happens because in the original dataset there are several values tending.

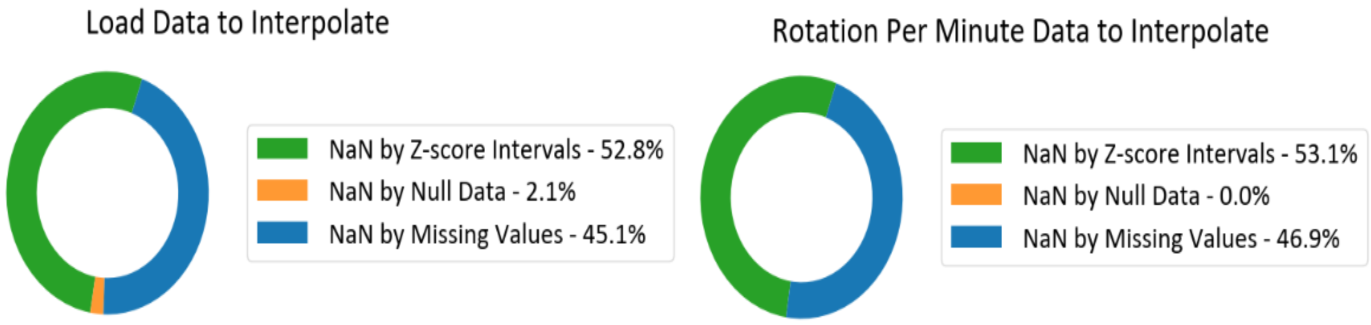

Fig. 6. Number of missing samples in the original data set.

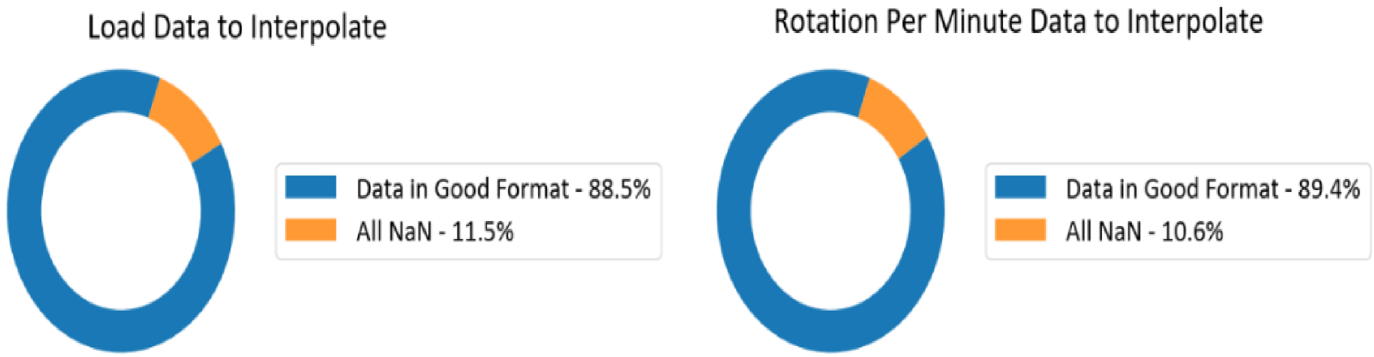

Fig. 7. Quality of data after cleaning discrepant data.

to 0. That was probably when the screw conveyor was stopped, but the sensors were still recording. That causes the average to be initially lower. As we later apply the z score thresholds, the lower values disappear and the mean value increased. Clearing points near zero is not a problem, since that means the equipment was possibly turned off or the sensors were not logging data. In either case the data are not relevant for the machine learning models.

Table 3. Analysis of the mean and standard deviation in the original data versus the data after applying the z score and interpolation.

Mean Standard Deviation				
Load	Before		31.58	5.50
	After		31.84	3.92
Velocity	Before		1281.47	452.09
	After		1343.38	281.86

6.3. Correlation and Dispersion

Analyses of the correlation of variables load and velocity were carried out for each year. The results are shown in Table 4. It is possible to notice that there are some years in which there is a weak correlation, while in others it is the opposite.

The best correlation result obtained happened in 2022, as can be seen in Table 4, but it is a misleading result. In 2022 there are only three months registered and there are some long term maintenance interventions.

This long duration of maintenance causes the values of the two sensors to be turned off or to tend towards zero, increasing the correlation, as shown in the results for the year 2022. In 2021 we evaluated a total of 12 months, with the same interruptions similar to previous years and, in fact, we are able to verify a correlation between the recorded signal of the two variables.

For comparative purposes and because of the previous explanation, a deeper analysis will be made between the worst year, which is 2018, and the second best year, which is 2021.

Figure 8 shows the dispersion of velocity versus load, in the years 2018 and 2021. Through the dispersion graphs it is possible to perceive the correlation previously noticed in Table 4. It is possible to observe that the graph on the left has values more dispersed while the graph on the right has the data present in a smaller area. This comparison can be analyzed through the white areas in the graphs. In 2021, the variables are more correlated and therefore there are more empty areas. In 2018, as the variables are not correlated, the empty areas are almost non-existent and the data points are more dispersed.

It can also be seen that the values of velocity the 2018 scatter plot are lower than in 2021, while for load it is the opposite.

Table 4. Analysis of the correlation between load and rotation per minute vari- ables annually.

Years	Correlation
2015	-0.17
2016	0.02
2017	-0.22
2018	-0.28
2019	-0.14
2020	-0.27
2021	0.23
2022 (3 months only)	0.60

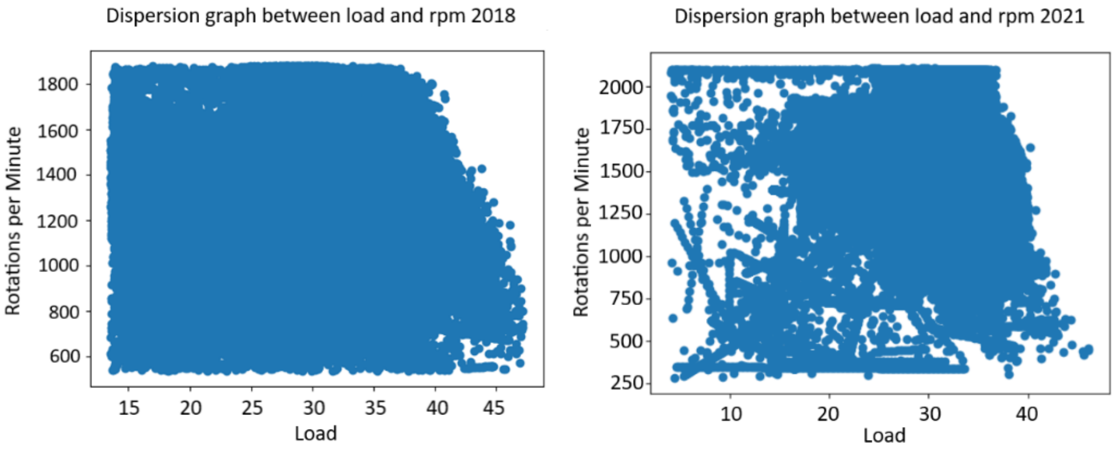

Fig. 8. 2018 and 2021 scatter plots.

6.4. Lag Plots

The lag plot also allows to analyze the presence of autocorrelation in time series data at different time intervals. This is a measure of how related a data point is to an earlier data point in time. By analyzing the autocorrelation through this approach, we can see if the data are scattered randomly in a lag plot. If they are spread out, it would mean little or no autocorrelation in the data.

However, if there is a recognizable pattern or structure such as a diagonal line, this suggests that there is autocorrelation present in the data. This auto- correlation is important because certain patterns and trends can clearly indicate the need for maintenance.

An analysis was made for the two dataset variables and it was concluded that they establish a direction, that is, a clear trend of correlation over time, as can be seen in Figure 9. It is important to mention that the figures present a lag equal to one and with the increase of these lags there is more dispersion. It is possible to notice that the velocity has more dispersion than the load variable, but in both cases it is possible to notice that there is the correlation that was being sought, even though with a considerable amount of dispersion, specially in velocity

An analysis of the lags plots was also carried out, but with the data of the original signal, without any pre-processing (no discrepant data removed and no interpolation). The result can be seen in Figure 10. Variables go from 0 to higher values very quickly and back again. These events can be reading, recording or communication errors, which is just noise in the data and not a realistic behaviour of the screw.

While in Figure 9 the data are not so dispersed because they were submitted to the pre-processing already explained previously, in Figure 10 the opposite happens. Through this example it is possible to visualize the importance and necessity of treating outliers.

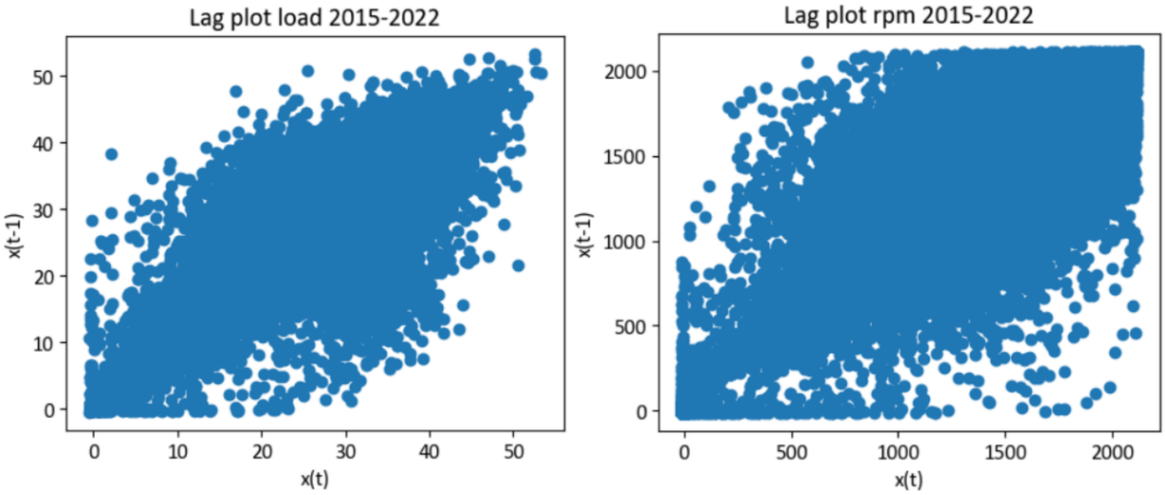

Fig. 9. Lag plots for load and rotation, for all sample data in the period 2015- 2022, with lag=1.

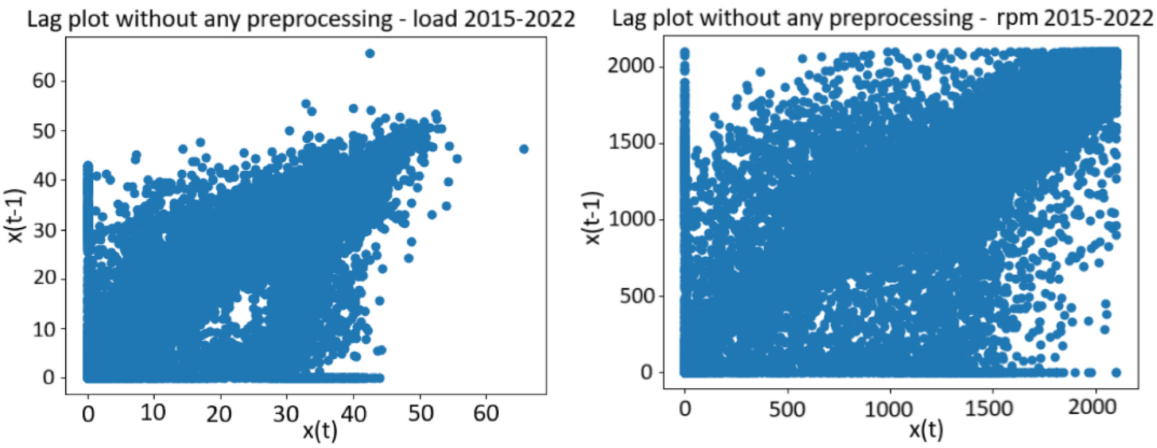

Fig. 10. Lag plot without any preprocessing 2015-2022, with lag=1.

6.5. Seasonality

As the data are temporal, they may carry evidence of temporal phenomena such as working shifts, holidays, vacations, periods of material loading on the screw conveyor, specific hours, among many others. A possible approach to better understand some of these patterns is seasonality, which is a repeating pattern. Seasonality is an important aspect to consider when analyzing time series data, as it can have a significant impact on the fault detection and prediction results.

Seasonal Decomposition

Seasonal signals can be decomposed into different components, such as trend, seasonal component and random noise [2,1,8].

Trend refers to the general direction in which the series is moving over time. The trend can be increasing, decreasing or stable over time. An increasing trend means that the time series is increasing over time, while a decreasing trend means that the series is decreasing. A stable trend means that the series is remaining constant over time.

Seasonality refers to the regular variations that occur over a specific time interval. Seasonality can be described as additive or multiplicative. Additive seasonality occurs when seasonal variation is constant across all periods and is added to a constant mean value. Multiplicative seasonality occurs when the seasonal variation increases or decreases as the average value increases or decreases. Random noise is identical to an associated error. This is a time series component that represents random or unpredictable variation. This random noise can be caused by factors such as measurement or sampling errors, unpredictable events or changes in the environment and usually has no clear standard meaning

associated with it, which can make data analysis difficult.

Seasonal Decomposition Results

The results of the seasonal decomposition were performed using the python statsmodels library. The library provides a seasonal decompose function that identifies the seasonal, trend, and residual temporal components.

Seasonal decomposition has a period parameter that represents the number of observations in a seasonal cycle. This configurable parameter can be adjusted for periods of a few hours, days, weeks or even shorter or longer periods.

Different time periods, correspond to different observations and patterns, but for our study, we found that 1-2 day windows can be ideal for finding patterns on an annual basis. In order to understand the ideal windows, it was necessary to analyze several time periods, as shown in Table 11. It started with a time period of half a day and it was noticed that in the window of 1 day the residue is smaller. Seasonality differs in both and it is clear that the longer the time period, the greater its scale. In a second analysis, between the periods of 1 and 2 days, the residue is maintained but the seasonality scale increases as in the previous analysis. Comparing the 2-day and 4-day window, it can be concluded that the values in the seasonal variation and in the residue variation were maintained. Seasonal variation values indicate a periodic variation compared to the average or trend values of the data, as the analysis is for a period of one year. Generally, a seasonal variation for this type of equipment is related to periodic workloads in the factory, but it may also be a sign of periodic wearing out of some part.

Figure 11 shows an example of application of the seasonal decomposition, more specifically with a period of 2 days as previously mentioned. Table 5 shows the seasonal variation and residue obtained with different sliding window sizes.

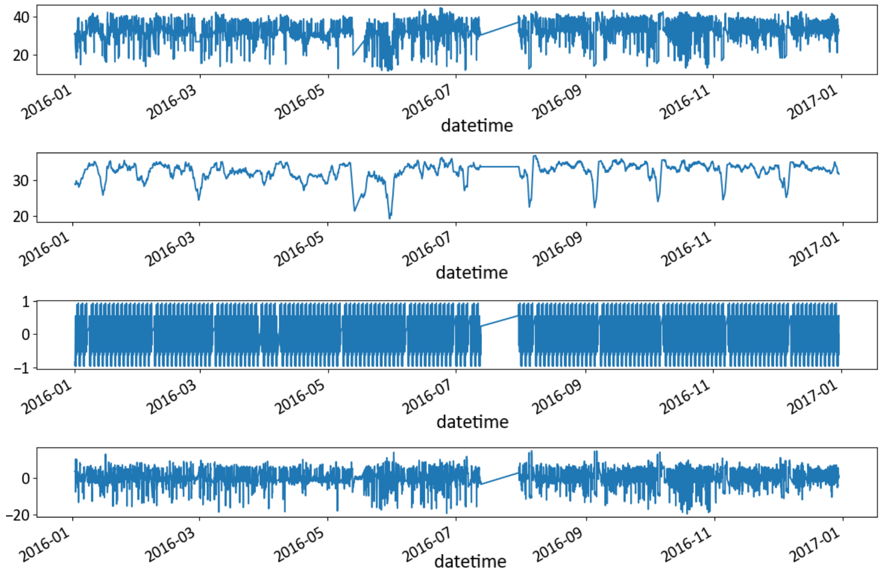

Fig. 11. Example of application of seasonal decomposition, with a period of 2 day.

Table 5. Seasonal decomposition results obtained for different sliding window sizes.

Window	Seasonal Variation	Residue
1/2 day	-0.25 to 0.25	-25 to 25
1 day	-0.50 to 0.50	-20 to 20
2 days	-1 to 1	-20 to 20
4 days	-1 to 1	-20 to 20

6.6. Stationarity

A series is considered stationary if it does not change its statistical properties over time. Mean, variance, and autocorrelation are some of the characteristics that are analyzed to identify whether or not a series is stationary [8]. Nonstationary time series often exhibit trends and other patterns that change over time, such as stock prices, economic indicators, and weather patterns[19].

Types of Stationarity There are two types of stationarity [4]: strong stationarity and weak stationarity. Strong stationarity implies that all statistical properties of the time series are constant with respect to time. In a strong stationary time series, the autocorrelation is constant over time, which implies that the patterns repeat regularly.

Weak stationarity in a time series means that statistical properties such as mean and variance can vary over time, but this variation occurs around a con- stant mean value or around a regular pattern. This means that, on average or in general, the time series maintains a constant and predictable behavior over time.

Stationarity Test Stationarity is a fundamental assumption in many time series models, as it allows us to make accurate predictions based on past observations. An analysis of the Dickey-Fuller test can be performed, in order to evaluate whether a data set of a time series is stationary or not [36]. In the Dickey-Fuller test, the α was set to 0.05, that is, a 95% confidence test was per- formed to analyze whether our series are stationary or not [20]. Going a little deeper into Dickey-Fuller, it returns interesting variables that are also part of the analysis. It returns a value representing the test statistic and a p-value.

The test statistic is a number that measures how non-stationary the time series is. The more negative the number, the stronger the evidence for stationarity. The p-value is a measure of probability that analyzes whether or not a null hypothesis is true. If the p-value is less than the α confidence level, then the null hypothesis is considered unlikely to be true and the time series is concluded to be stationary. In a more simplified way, if there is a low value of p, in principle the Dickey-Fuller test will return the time series as stationary.

An attempt was made to submit the data in full, but due to excessive computation, annual tests were chosen. In all annual tests, the stationarity result was successfully obtained and can be seen in Table 6.

If a time series is not stationary, it can be transformed into a stationary series through techniques such as differencing, which involves calculating the differences between the values observed at different points in the time series [21]. Differentiation can remove non-stationary trends and patterns from the time series, making it stationary and easier to model.

Table 6. Stationarity Results by Year.

Year	Variable	Test statistics	α	P-value
2015	load	-16.587	2.861	$1.812 \times e^{-29}$
	rpm	-9.675	2.861	$1.246 \times e^{-16}$
2016	load	-17.064	2.861	$7.901 \times e^{-30}$
	rpm	-15.178	2.861	$6.242 \times e^{-28}$
2017	load	-22.310	2.861	0.0
	rpm	-13.643	2.861	$1.635 \times e^{-25}$
2018	load	-19.887	2.861	0.0
	rpm	-15.761	2.861	$1.190 \times e^{-28}$
2019	load	-20.356	2.861	0.0
	rpm	-10.817	2.861	$1.843 \times e^{-19}$
2020	load	-19.171	2.861	0.0
	rpm	-10.706	2.861	$3.411 \times e^{-19}$
2021	load	-11.297	2.861	$1.331 \times e^{-20}$
	rpm	-9.767	2.861	$7.278 \times e^{-17}$
2022	load	-8.079	2.861	$1.475 \times e^{-12}$
	rpm	-7.726	2.861	$1.151 \times e^{-11}$

6.7. Autocorrelation

Autocorrelation can also help understand whether the data are stationary or not, and how they behave over time. Through autocorrelation graphs, if they have a sinusoidal shape, then it means the data are stationary and there is no trend in the data. In the graphical analysis of the autocorrelation, the values tend to be close to zero, as shown in Figure 12. The example on the figure represents a period of one year, more specifically the year 2021 in the load variable.

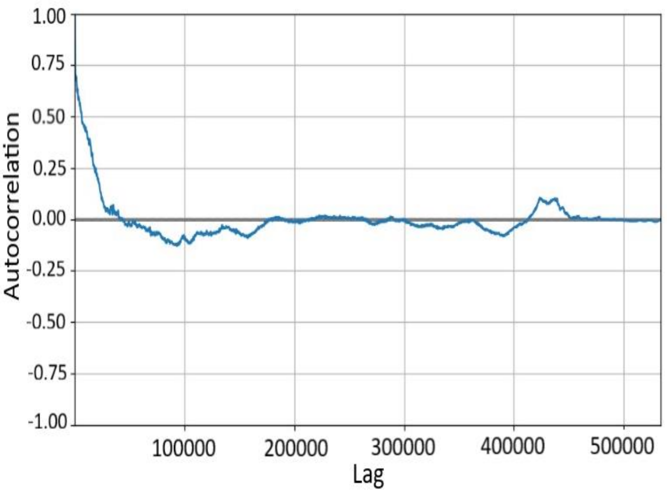

Fig. 12. Autocorrelation of variable load during year 2021.

For the prediction of data and events, it is necessary to ensure that the data are stationary. If, at that moment, it was verified that the autocorrelation was non-stationarity, one could apply the first or second-order differentiation to make the signal stationary as previously mentioned.

6.8. Smoothing the data

Filters can also be applied to clarify or smooth trends, detect outliers, or just reduce noise. In this case, smoothing is particularly interesting, because the signals exhibit a lot of noise. The presence of noise difficults interpretation and also the process of training machine learning models. After smoothing, discrepant data and noise are minimized, or removed, and the trends and patterns become clearer to observe.

LOWESS (Locally Weighted Scatterplot Smoothing) is a method widely used to graphically smooth data [16]. With the application of LOWESS the data will be visually less dispersed and it is possible to remove some of the random variations in the signal [11]. The LOWESS method works by fitting a curve through the data using an weighted moving average. At each point on the scatterplot, a weighted average is taken and data weights are determined on the window points.

In practice, LOWESS assigns greater weights to nearby points and smaller weights to more distant points. This filter is a simple and effective method of smoothing out noisy data and can also help identify patterns in scatterplots. It is often used in exploratory data analysis to identify trends and relationships in data and to detect outliers and unusual patterns. Examples were chosen to visualize the result of the LOWESS filter, which can be seen in Figures 13 and 14.

LOWESS can be configured to smooth the data with sliding windows of different periods of time, but it was decided to apply this smoothing with a sliding window of 24 h. LOWESS has a fraction parameter that performs the smoothing. This fraction must contain the period to be smoothed (p), divided by the total amount of samples in the data set (t), as presented in Equation 4. In Figures 13 and 14, the fraction corresponds to the number of samples of 24 hours

$$fraction = \frac{p}{t} \tag{4}$$

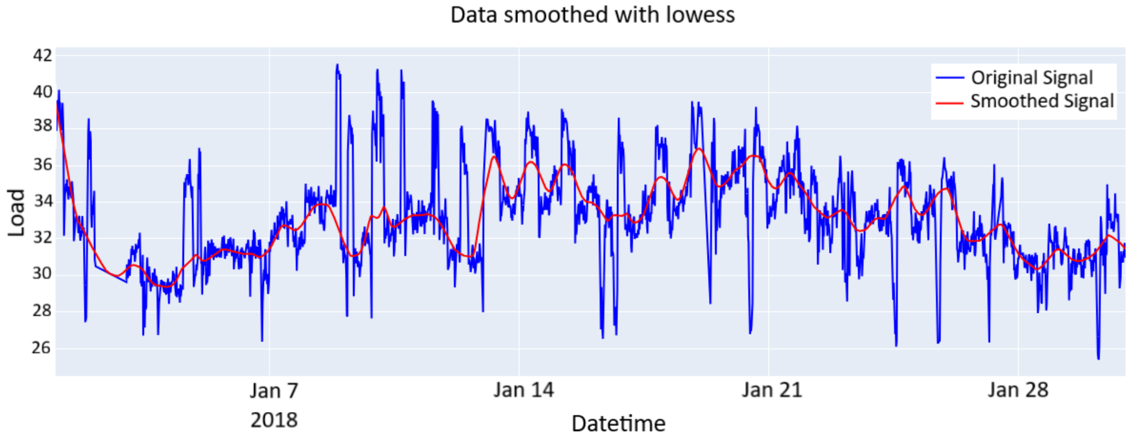

Fig. 13. Original signal and signal smoothed with LOWESS filter for variable Load.

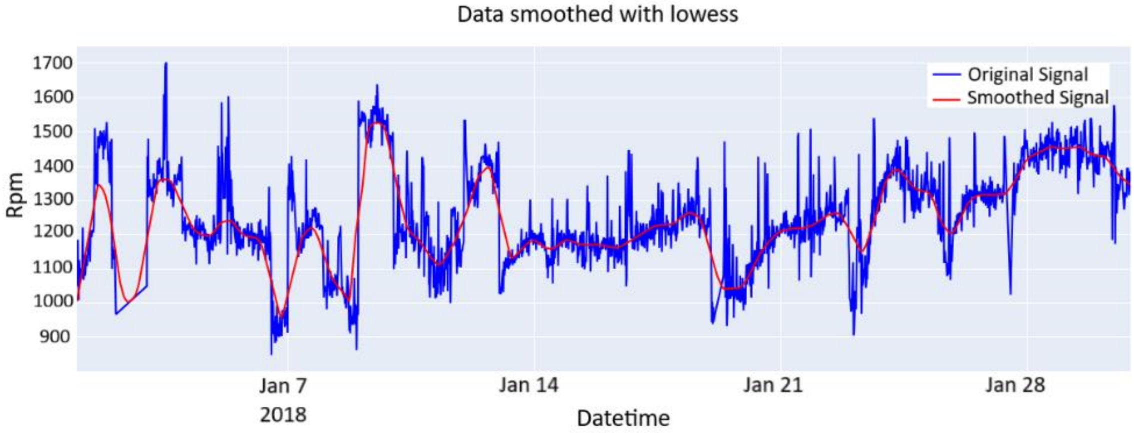

Fig. 14. Original signal and signal smoothed with LOWESS filter for variable Velocity.

7. Data preparation

In order to incorporate temporal information in each sample, more variables were created and added to the dataset. The sampling rate was also decreased to one sample per hour instead of one per minute.

7.1. Incorporating temporal information

In general, the more variables the better for the algorithms to be able to draw more conclusions. The original dataset only had three time series variables (time, load and velocity), but it was decided to expand it with the first and second-order differentiation for both sensors.

The first-order differentiation, if positive means that the initial function in- creases, and if negative means that the initial function decreases.

In second-order differentiation, if the derivative of the function is positive, the original function is concave upwards, while if it is negative, the original function is concave downwards. In second-order differentiation when the derivative is equal to 0 it means the change of concavity.

Adding the first and second-order differentiation to sensor data provides benefits such as improved trend analysis, better understanding of acceleration patterns, and enhanced capabilities for data analysis and modeling.

The datetime variable was also transformed through one-hot encoding to extract the days of the week and representative weeks in an entire year. This one-hot encoding process resulted in seven more new features for the days of the week and 52 more features for the weeks throughout the year. The objective of doing this one-hot encoding of the time is to incorporate temporal information into the features. This, if there is maintenance on specific days or specific weeks, that can be encoded into the input for machine learning models. If it is common to have large scale maintenance in a specific month, then the representative weeks of that month can be a determining factor for classifying anomalies.

So each sample X_i of the dataset is now represented as in Equation 5

$$X_i = (L, L_{D1}, L_{D2}, \Omega, \Omega_{D1}, \Omega_{D2}, W_1, \dots W_{52}, D_1, \dots D_7) \qquad (5)$$

where L is the load variable and Ω is the velocity variable. L_{D1} and Ω_{D1} represent the first order differentiation. L_{D2} and Ω_{D2} represent the second order applied to the original variables. W_j, for $0 < j \leq 52$, is the week of the year in one-hot encoding. The day of the week in one-hot encoding is D_k, for $0 < k \leq 7$.

7.2. Downsampling

Time windows can be extracted at different time periods. The dataset has records of information with period of 1 minute, but for the industrial context, it is sufficient to classify in longer periods to make mid-term or long-term decisions. For this purpose, downsampling techniques were applied, incorporating the average of the downsampling window.

To better contextualize, without downsampling, a 24 hour extraction with a period of 1 minute would result in 1,440 records (equivalent to 1,440 minutes in a day) as shown in the example in Figure 15.

In this case, a downsampling window of 60 samples was applied and the 24 hour extraction would have only 24 records, where sample represents the average of 60 sensor readings, as shown in the in Figure 16.

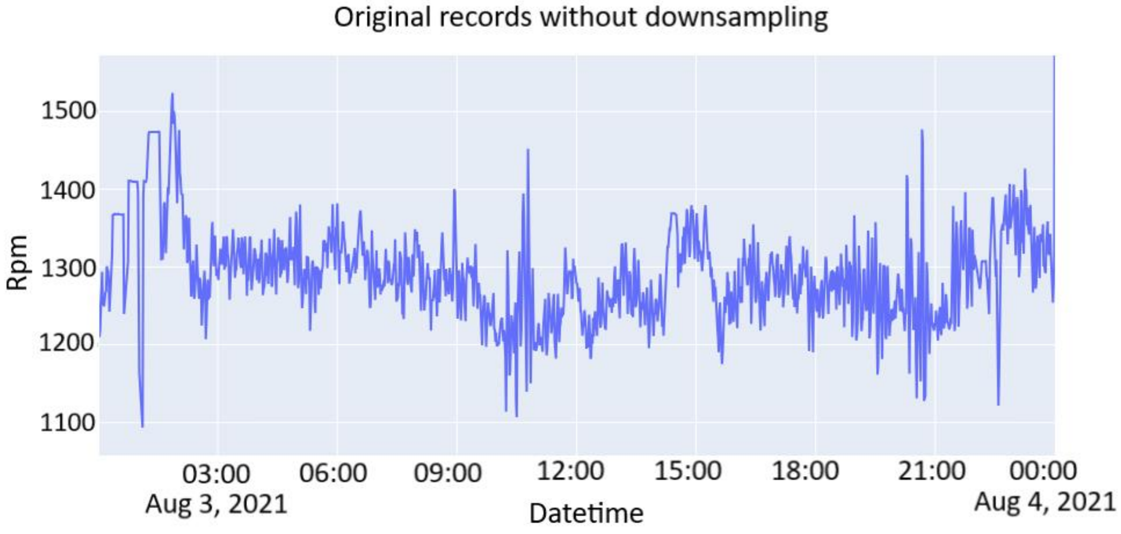

Fig. 15. Velocity during a 24 hours time window, before downsampling.

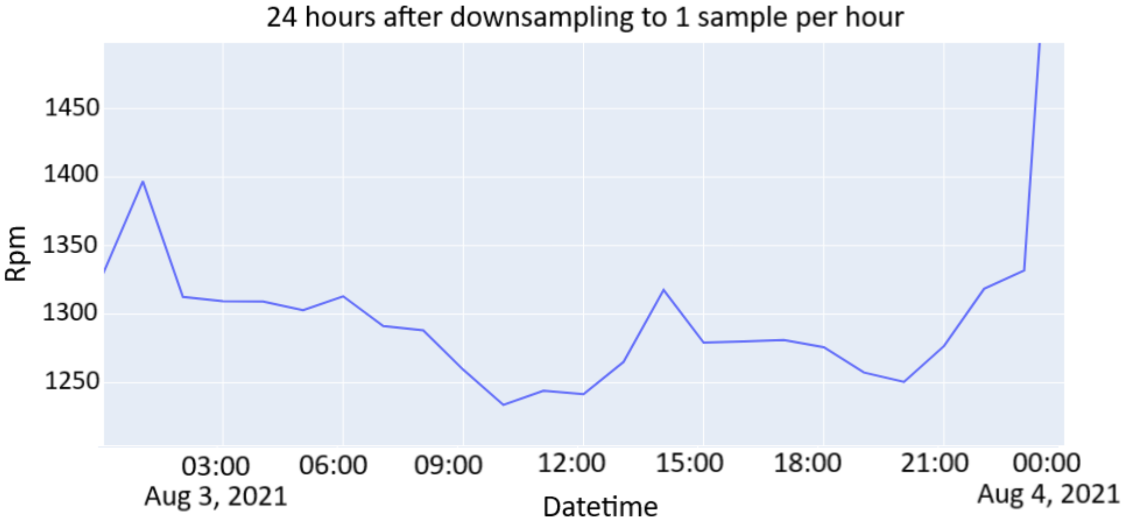

Fig. 16. Velocity during 24 hours after downsampling to 1 sample per hour.

8. Types of Maintenance and Respective Extraction of Windows

Table 2 shows the maintenance interventions whose need is aimed to be predicted. If the need for those interventions is predicted in advance, the procedure can be planned and optimized in order to minimize downtime and other costs. The task is very difficult, specially considering that there are few examples of maintenance interventions available.

The maintenance events in Table 2 were identified through a maintenance log report provided by the maintenance team. This report provides the name of the maintenance, date and time from the start of the maintenance to its completion. Data were extracted from the windows of three days before until the date scheduled for the start of maintenance. This approach guarantees analyzing variable patterns three days before the screw conveyor needs maintenance. If each day extracted represents 24 samples, then each three day window has 72 samples. Through Figure 17 it is possible to see how the explained extraction is carried out.

After extracting the windows and pre-processing the data, as explained in the previous section, it is possible to associate each window with each type of maintenance, or no maintenance, intervention. As there are few maintenance examples for each type, the problem is solved through binary classification. The two possible states are normal operation of the screw conveyor or maintenance events in general.

After the data preparation steps described above, the dataset has 2,160 records (30 windows extraction with 72 hours each) not leading to maintenance events and 936 records (13 windows extraction with 72 hours each) leading to maintenance events. For the maintenance and non-maintenance examples, 3,096 (2,160 + 936) were the total number of records.

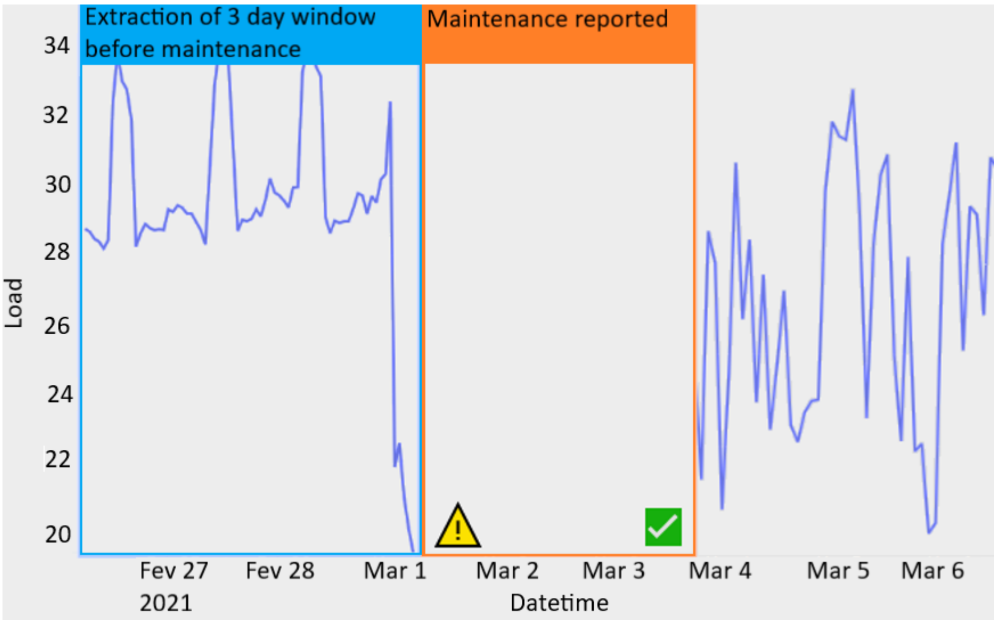

Fig. 17. Example of extraction of a maintenance window.

9. Classification

9.1. Training and testing sets

Random division was tested and it was noticed that points from the test set would be side by side with those from the training sets. In random division, the model would understand the similarities of temporal features and assign the target correctly in all events, which would cause overfitting.

To solve this overfitting problem, some complete windows were chosen for the different sets. For the training set, 22 complete normal operation windows and 10 complete maintenance windows were added, resulting in 2,304 records ((22 + 10) 72 records for each window). For the test set, 8 complete normal operation windows and 3 complete maintenance windows were added resulting in 792 records ((8 + 3) 72 records). The records mentioned guarantee a division of around 75% of the windows for the training set and 25% for the test set.

9.2. Classification experiments

The application of smoothing using LOWESS was tested with different granular- ity fractions but the results in the confusion matrix were the same. If smoothing had improved the classification, it would have been added, but in the present case it was effective for visualization purposes only.

The data set is unbalanced, as previously seen, but we could have increased the data from the minority class. SMOTE (Synthetic Minority Over-sampling Technique) was tested on the training set to balance the number of samples on the minority class, but the impact on the results was not relevant, so it was abandoned.

The Pycaret tool was used to evaluate several classification models at the same time [13]. The hyperparameters used are the default ones for each algorithm.

The Extra Trees Classifier (ETC) has obtained the best result with 0.9911 F1-Score and the results obtained through Pycaret can be seen in Table 7. ETC is a variation of Random Forest. It chooses random splits for each node, while Random Forest tries to find the best split for each node in the tree [10]. The randomness of ETC can result in more diverse trees and in some cases it can achieve better performance compared to Random Forest. However, the Random Forest algorithm managed to obtain very approximate and similar results. The Extra Trees Classifier algorithm usually obtains better results when there are many attributes or when computational efficiency is taken into consideration.

Results using a feedforward neural network model were added to Table 7. The results of this experiment were not obtained through Pycaret, but were added to the table for comparative purposes.

Of the 216 maintenance examples tested, the best algorithm failed in 29 of the results, while in examples without maintenance, it also failed 66 times out of 576 records, as shown in Figure 18 with the results of the confusion matrix.

Table 7. Classification results obtained through Pycaret, for different algorithms, and feedforward Artificial Neural Network.

	Model	Accuracy	AUC Recall Precision	F1
et	Extra Trees Classifier	0.9870	0.9994 0.9919 0.9909	0.9911
rf	Random Forest Classifier	0.9752	0.9974 0.9866 0.9798	0.9831
xgboost	Extreme Gradient Boosting	0.9726	0.9948 0.9848 0.9781	0.9813
lightgbm	Light Gradient Boosting Machine	0.9648	0.9934 0.9830 0.9692	0.9760
ada	Ada Boost Classifier	0.9609	0.9924 0.9731 0.9732	0.9731
gbc	Gradient Boosting Classifier	0.9596	0.9947 0.9740 0.9706	0.9722
dt	Decision Tree Classifier	0.9530	0.9431 0.9651 0.9705	0.9676
annf	Artificial Neural Network Feedforward	0.9455	0.9908 0.9693 0.9569	0.9630
lr	Logistic Regression	0.9139	0.9842 0.9461 0.9366	0.9410
lda	Linear Discriminant Analysis	0.9113	0.9807 0.9202 0.9567	0.9377
ridge	Ridge Classifier	0.9100	0.0000 0.9228 0.9525	0.9370
nb	Naive Bayes	0.8969	0.9803 0.8583 1.0000	0.9234
qda	Quadratic Discriminant Analysis	0.8943	0.8998 0.8879 0.9645	0.9242
dummy	Dummy Classifier	0.7273	0.5000 1.0000 0.7273	0.8421
svm	SVM - Linear Kernel	0.5371	0.0000 0.5784 0.4432	0.5003

It is also important to understand the results through graphical visualizations. Figure 19 shows an example of a maintenance window in the test set. The points in green mean correspond to true positives and the points in red are false positives.

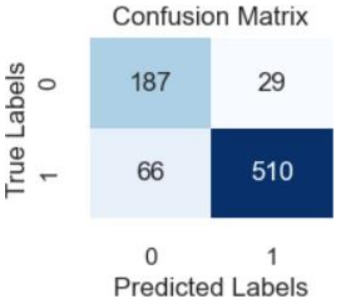

Fig. 18.Confusion matrix of the Extra Trees classifier, where 0 means maintenance intervention and 1 no maintenance.

Through the confusion matrix it is possible to see that the model failed to classify more information than in the results of the training set. However, through the test set, it is possible to observe that the majority of events are correctly predicted by the models.

Through the results of the confusion matrix for the test set with the Extra Trees, the result obtained for F1 is 0.7974.

It must also be noticed that the exact times of the beginning and end of a maintenance intervention may be blurry, and that can also impact on the results. When the maintenance team has more maintenance windows, the model can be retrained to incorporate more information and therefore improve the results. This technique ensures that the machine learning model can analyze behaviors similar to those that occurred before the screw conveyor failed. It is already clear from the results that the model is able to learn, so in the future more data can improve the results or lead to more sophisticated feature engineering or models to be applied.

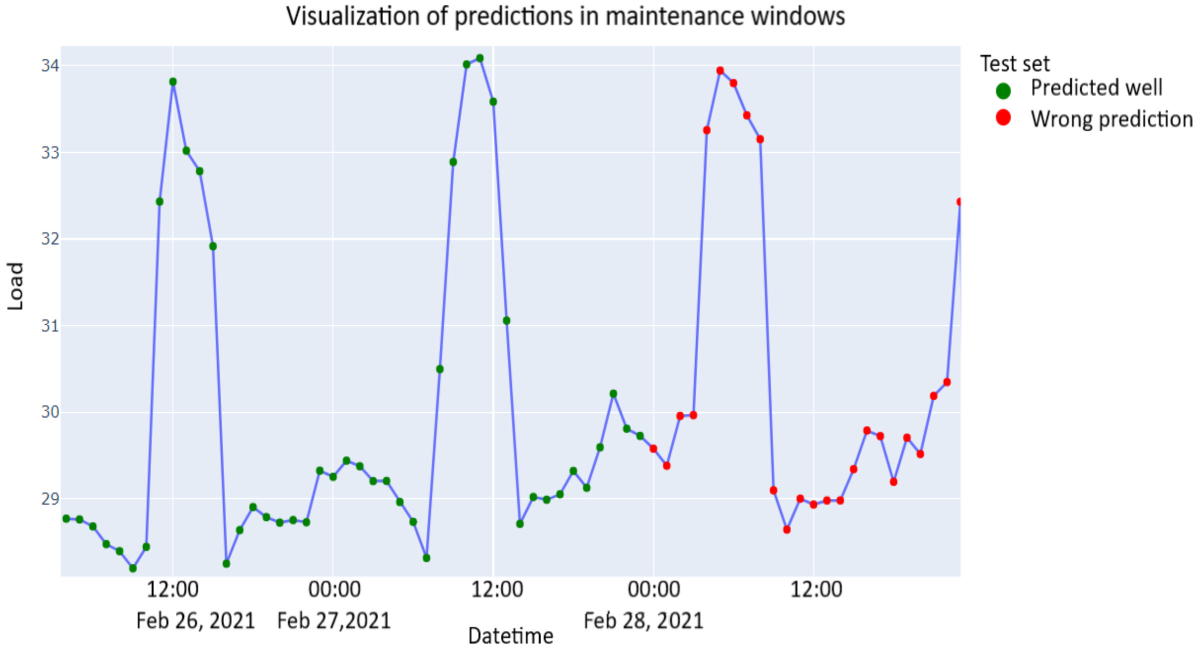

Fig. 19. Visualization of classification results in part of the test set. The image shows correct and incorrect predictions in a test window.

10. Discussion

Classifying anomalies through machine learning is a crucial area of research for detecting unusual behaviors in predictive maintenance. In this article, we explored a possible approach to successfully perform anomaly detection and explained all the necessary steps from raw data to classification to the particular case of a screw conveyor in a biomass industry. The dataset provided by the industry contains timestamp, load and velocity with period of one minute. Records of maintenance interventions were also provided. The dataset is small for final conclusions, but a wealth of information was already extracted.

The analysis performed on the data showed how the screw conveyor behaves over time. Normal functioning points are slightly different in different years. Nonetheless, the time series are stationary on an yearly basis and they also exhibit some seasonal components.

The data were transformed in order to convey temporal information and be fed to machine learning models. The machine learning models were able to determine maintenance interventions with high accuracy. Extra Trees showed the best performance of all, with an F1 of 0.7974. Direct comparison with other state of the art approaches is not possible, due to the fact that no previous studies were found for screw conveyors. Nonetheless, the results obtained show that there is a great potential in the use of machine learning for anomaly detection and prevention in screw conveyors. The accuracy in detection of maintenance interventions is comparable to the results reviewed in the state of the art for other equipment.

As we advance in the field of anomaly detection, it becomes clear that all the information provided by a machine learning model is an asset to any automated business. That is specially important in equipment such as the screw conveyor under analysis, because of its role in feeding wood chips that supply all the production line ahead.

11. Conclusion

Screw conveyors are very important in industrial plants, as they feed the supplies to the equipment ahead of them. They are in general very reliable, but monitoring and timely repairs are crucial to avoid catastrophic downtimes.

The present paper analyses data from a screw conveyor. The analysis per- formed shows how the screw behaves in different years. The data were also transformed and fed to different classifiers, and the results showed that many of them are able to detect when a maintenance intervention is needed.

Future works include more experiments, in order to optimize the classification model, possibly incorporating more data from the power plant.

References

1. Wen, Q., Gao, J., Song, X., Sun, L., Xu, H. & Zhu, S. RobustSTL: A robust seasonal-trend decomposition algorithm for long time series. Proceedings Of The AAAI Conference On Artificial Intelligence. 33, 5409-5416 (2019) https://doi.org/10.1609/aaai.v33i01.33015409
2. Bandara, K., Bergmeir, C. & Hewamalage, H. LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. IEEE Transactions On Neural Networks And Learning Systems. 32, 1586-1599 (2020) https://doi.org/10.1109/TNNLS.2020.2985720
3. Curtis, A., Smith, T., Ziganshin, B. & Elefteriades, J. The mystery of the Z-score. Aorta. 4, 124-130 (2016) https://doi.org/10.12945/j.aorta.2016.16.014
4. Mudersbach, C. & Jensen, J. Nonstationary extreme value analysis of annual max- imum water levels for designing coastal structures on the German North Sea coast- line. Journal Of Flood Risk Management. 3, 52-62 (2010) https://doi.org/10.1111/j.1753-318X.2009.01054.x
5. Pezo, L., Jovanović, A., Pezo, M., Čolović, R. & Lončar, B. Modified screw conveyor-mixers-Discrete element modeling approach. Advanced Powder Technology. 26, 1391-1399 (2015) https://doi.org/10.1016/j.apt.2015.07.016
6. Sudarso, L. & Rachmat, R. Design of Belt Conveyor for Sandblasting Material Handling System. Jurnal Teknik Mesin Dan Mekatronika (Journal Of Mechanical Engineering And Mechatronics). 5, 28-37 (2020) https://doi.org/10.33021/jmem.v5i1.942
7. Karwat, B., Rubacha, P. & Stan'czyk, E. Optimization of a screw conveyor's ex- ploitation parameters. Eksploatacja I Niezawodno's'c. 23 (2021) https://doi.org/10.17531/ein.2021.2.8
8. Jebb, A., Tay, L., Wang, W. & Huang, Q. Time series analysis for psychological research: examining and forecasting change. Frontiers In Psychology. 6 pp. 727 (2015) https://doi.org/10.3389/fpsyg.2015.00727
9. Deulgaonkar, V., Walimbe, A., Mahurkar, S., Patil, S. & Kakade, S. Failure Anal- ysis of Industrial Screw Conveyors. Journal Of Failure Analysis And Prevention. 23, 1693-1700 (2023) https://doi.org/10.1007/s11668-023-01719-3
10. Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. Machine Learn- ing. 63, 3-42 (2006) https://doi.org/10.1007/s10994-006-6226-1
11. Jamlos, M., Jamlos, M., Mustafa, W., Othman, N., Rohani, M., Saidi, S., Sarip,M. & Nawi, M.Reduced graphene oxide uwb array sensor: high performance for brain tumor imaging and detection. Nanomaterials. 13, 27 (2022) https://doi.org/10.3390/nano13010027
12. Zaica, A., Visan, A., Paun, A., Gageanu, P., Bunduchi, G., Zaica, A., Stefan, V. & Manea, D. The coating process of corn grains using a treatment machine with brush screw conveyor. Proceedings Of The 44th International Symposium On Agricultural Engineering: Actual Tasks On Agricultural Engineering. pp. 23-26 (2016)
13. Ali, M. PyCaret: An open source, low-code machine learning library in Python. (2020,4), https://www.pycaret.org, PyCaret version 1.0.0
14. Hadi, S., Al Azis, A., Viyus, V., Puspitasari, E., Firdaus, A. & Setiawan, A. Plan- ning for Maintenance and Repair of Continuous Ship Unloader Using the IRRO Method. Logic: Jurnal Rancang Bangun Dan Teknologi. 21, 52-63 (2021) https://doi.org/10.31940/logic.v21i1.2383
15. Geca, J. Performance comparison of machine learning algorithms for predictive maintenance. Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie S'rodowiska. 10 pp. 32-35 (2020,9),https://www.mendeley.com/catalogue/32e41dca-742c-3e16-bd55-26f68a0da004, https://doi.org/10.35784/iapgos.1834
16. Wilcox, R. The regression smoother LOWESS: A confidence band that allows heteroscedasticity and has some specified simultaneous probability coverage. Journal Of Modern Applied Statistical Methods. 16, 3 (2017) https://doi.org/10.22237/jmasm/1509494580
17. Martins, A., Mateus, B., Fonseca, I., Farinha, J., Rodrigues, J., Mendes, M. & Cardoso, A. Predicting the Health Status of a Pulp Press Based on Deep Neural Networks and Hidden Markov Models. Energies. 16 (2023),https://www.mdpi.com/1996-1073/16/6/2651, https://doi.org/10.3390/en16062651
18. Rodrigues, J., Martins, A., Mendes, M., Farinha, J., Mateus, R. & Cardoso,A. Automatic Risk Assessment for an Industrial Asset Using Unsupervised and Supervised Learning. Energies. 15 (2022), https://www.mdpi.com/1996- 1073/15/24/9387 https://doi.org/10.3390/en15249387
19. Fukuda, K., Stanley, H. & Amaral, L. Heuristic segmentation of a nonstationary time series. Physical Review E. 69, 021108 (2004) https://doi.org/10.1103/PhysRevE.69.021108
20. Tatarintsev, M., Korchagin, S., Nikitin, P., Gorokhova, R., Bystrenina, I. & Serdechnyy, D. Analysis of the forecast price as a factor of

sustainable development of agriculture. Agronomy. 11, 1235 (2021) https://doi.org/10.3390/agronomy11061235

21. Hossain, Z., Rahman, A., Hossain, M. & Karami, J. Over-differencing and forecasting with non-stationary time series data. Dhaka University Journal Of Science. 67, 21-26 (2019) https://doi.org/10.3329/dujs.v67i1.54568
22. Bourke, P. Interpolation methods. Miscellaneous: Projection, Modelling, Render-ing. 1 (1999), doi: 10.1213/ANE.0000000000002864. PMID: 29481436.
23. Garcia, S., Luengo, J. & Herrera, F. Data preprocessing in data mining. (Springer, 2015) https://doi.org/10.1007/978-3-319-10247-4
24. Swanson, L. Linking maintenance strategies to performance. International Journal Of Production Economics. 70, 237-244 (2001), https://doi.org/10.1016/S0925-5273(00)00067-0
25. Zareiforoush, H., Komarizadeh, M., Alizadeh, M., Masoomi, M. & Tavakoli, H.Performance evaluation of screw augers in paddy grains handling. International Agrophysics. 24 pp. 389-396 (2010,1)
26. Strauß, P., Schmitz, M., Wostmann, R. & Deuse, J. Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning. 2018 IEEE International Conference On Big Data (Big Data). pp. 1474-1483 (2018) https://doi.org/10.1109/BigData.2018.8622076
27. Cavallo, C. All About Screw Conveyors - Types, Design, and Uses. (Thomasnet), (last access on: 2021-10-01) https://www.thomasnet.com/articles/materials- handling/all-about-screw-conveyors/
28. Evstratov, V., Rud, A. & Belousov, K. Process Modelling Vertical Screw Trans-port of Bulk Material Flow. Procedia Engineering. 129 pp. 397-402 (2015),https://www.sciencedirect.com/science/article/pii/S1877705815040187, International Conference on Industrial Engineering (ICIE-2015) https://doi.org/10.1016/j.proeng.2015.12.134
29. Tian, Y., Yuan, P., Yang, F., Gu, J., Chen, M., Tang, J., Su, Y., Ding, T., Zhang, K. & Cheng, Q. Research on the Principle of a New Flexible Screw Conveyor and Its Power Consumption. Applied Sciences. 8 (2018), https://doi.org/10.3390/app8071038
30. Waje, S., Thorat, B. & Mujumdar, A. Screw Conveyor Dryer: Process and Equip-ment Design. Drying Technology - DRY TECHNOL. 25 pp. 241-247 (2007,2) https://doi.org/10.1080/07373930601161112
31. Gobdc Planning For Every Eventuality: Instituting a Multi-Faceted Mainte- nance Program. (Gobdc), https://www.gobdc.com/planning-for-every-eventuality-instituting-a-multi-faceted-maintenance-program/
32. Sehrawat, D. & Gill, N. Smart sensors: Analysis of different types of IoT sen-sors. 2019 3rd International Conference On Trends In Electronics And Informatics (ICOEI). pp. 523-528 (2019) https://doi.org/10.1109/ICOEI.2019.8862778
33. Kanawaday, A. & Sane, A. Machine learning for predictive maintenance of in-dustrial machines using IoT sensor data. 2017 8th IEEE International Conference On Software Engineering And Service Science (ICSESS). pp. 87-90 (2017), https://doi.org/10.1109/ICSESS.2017.8342870