IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 August 2023, accepted 10 September 2023, date of publication 15 September 2023,
date of current version 22 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315842

RESEARCH ARTICLE

Exploration of Production Data for Predictive Maintenance of Industrial Equipment: A Case Study

NANNA BURMEISTER[1], RASMUS DOVNBORG FREDERIKSEN[1], ESBEN HØG[2], AND PETER NIELSEN[1]

[1]Department of Materials and Production, Aalborg University, 9000 Aalborg, Denmark
[2]Department of Mathematical Sciences, Aalborg University, 9000 Aalborg, Denmark

Corresponding author: Nanna Burmeister (nannaburmeister@outlook.dk)

ABSTRACT Data-driven predictive maintenance is typically based on collected data from multiple sensors or industrial systems over a period of time, where historical and real-time data are combined as input to black-box machine learning models. In the current study we provide a case study of a major manufacturing company of large industrial equipment. We investigate the opportunity to utilize the manufacturing state of the equipment alone to predict future conditions. The production data contain information about the errors or defects in the equipment found in production. The defects are potentially repaired before the equipment is installed. We present a proactive approach based on interpretable machine learning models, where the production data are used to predict maintenance, which creates the opportunity to prevent future maintenance. The solution is easily translated into a simple set of rules that can be used to separate critical production errors from non-critical production errors. Identifying critical production errors potentially prevents future and more expensive repairs of errors detected in inspections after the equipment has been installed. Our paper contributes to the literature on predictive maintenance in two ways. Firstly, we show the viability of a more proactive approach utilizing production data to prevent future maintenance. Secondly, we demonstrate the applicability of interpretable machine learning models to understand the relationship between the features of the production errors and the later inspection errors.

INDEX TERMS Case study, explainability, industrial equipment, interpretability, machine learning, predictive maintenance, preventive maintenance.

I. INTRODUCTION

Data is the key to the current information generation in Industry 4.0 and is used to anticipate or contribute to making decisions based on predictions of the future state of the system [1]. Predictive maintenance is a central topic in this field and is usually based on historical data-based models and domain knowledge [2], [3]. The impact of maintenance in manufacturing companies represents a total of 15 to 60% of the total costs of operating all manufacturing [4], [5]. The purpose of predictive maintenance is to anticipate pending failures in advance to improve the decision-making process for the maintenance activity mainly by avoiding the downtime [4], [5], [6]. Predictive maintenance can be sub-divided into three different approaches for the prediction process: physical model-based, knowledge-based, and data-driven. The latter accounts for the models most often found in current predictive maintenance solutions, which are based on statistics, pattern recognition, or artificial intelligence and machine learning models [4]. Central for data-driven predictive maintenance solutions are the models predicting the future conditions of the equipment.

The predictive maintenance research field is large, and the data used for data-driven predictive maintenance are typically collected from multiple sensors or industrial systems over a period of time, where historical and real-time data

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du.

are combined as input. The data-driven approach targets machine learning models for the purpose of predicting the maintenance or repair of industrial equipment. In this sense, the dominant approach to predictive maintenance is proactive as it is based on performance changes in industrial equipment. Maintenance is primarily required because of wear, however in some cases, it might be influenced by the manufactured state of the industrial equipment. If so, preemptively correcting the production lacks and flaws, maintenance might not be needed at all. Maintenance of industrial equipment is in general expensive, and the cost can be significantly reduced if the maintenance is prevented during production of the industrial equipment. The reduced costs are due to ease of access, no lost production, and the errors being smaller (assuming that the errors potentially develop after the industrial equipment is put into service).

The literature proves the effectiveness of a wide range of different machine learning implementations in predictive maintenance. However, most research uses so-called black-box methods focused on predictive performance without providing insights about root cause analysis and explainability. Despite the increased predictive power compared to simple interpretable approaches, their prediction logic is difficult or even impossible to fully explain. Thus the ability to build a predictive model with high accuracy comes at the cost of not being able to explain fully the main causes of impending failures. In such applications, the potential for prevention of the occurrence of maintenance is unexploited.

The current paper contributes a case study of a major manufacturing company of large industrial equipment in the predictive maintenance research field. The case study shows the possibility to utilize explainable machine learning models together with data collected from the manufacture state of the industrial equipment alone to predict future maintenance. By understanding the factors leading to equipment degradation or failure, it opens up the possibility to rectify the underlying phenomena instead of only addressing the symptoms, thereby avoiding maintenance altogether. Consequently, the impact of the variables can be translated into concrete actions on the production side.

The remainder of the paper is structured as follows. Section II provides a literature review of related research. Section III describes the intrepretable models used in the paper. Section IV gives a description of the case study, the data set, the data preprocessing, the training pipelines and the experimental setup. Section VI presents the results of the case study and discusses the results against the existing research. Finally, Section VII concludes the work.

II. BACKGROUND

The predictive maintenance research field is characterized by black-box machine learning models targeting high forecast performance in terms of accuracy. In [4], the authors cover the most commonly found predictive models in a literature review regarding predictive maintenance. These are Random Forest [7], [8], Deep Learning [8], [9], and other strategies linked to Artificial Neural Networks and Machine Learning [4]. The range of algorithms used is not very large and there are specific patterns for each type of need. The solutions are complex and it requires physical knowledge to adjust features, filters, and parameterization of prediction functions [4]. These tendencies are also found in [10], another recent literature review regarding predictive maintenance. Most recent studies use non-interpretable models with predictive performance as the main focus for the potential of avoiding unnecessary replacement of equipment, saving costs, or improving the safety, availability, and efficiency of processes [10], [11], [12].

Table 1 is a summary of relatively recent studies regarding predictive maintenance on real-life data [4], [10]. Black-box approaches (see the caption to Table 1) clearly dominate.

Even though it is sufficient to know about prediction accuracy in a low-risk environment, there are cases where it is essential to explain and interpret the model to understand how it arrived at the predictions. However, this ability often comes at the cost of quality in the predictive performance. Reference [13] argues that the use of explainable machine learning models creates solutions with an adequate explanation so that the end-user can understand the overall behavior, weakness, and strength of the system. Recent work [14] challenges the use of black-box methods within predictive maintenance. Thereby leveraging modern optimization techniques to construct interpretable methods with performance rivaling the black-box methods while enabling the use of the insights and the confidence that interpretability brings. In [15], the authors compares remaining useful life prediction provided by different explainable machine learning methods using different metrics. They conclude, the framework to be really useful for both local and global explanations. Similarly, [16] considers predicting hard drive failures in a data center using interpretable machine learning. The paper demonstrates the ability to provide meaningful insights about short- and long-term hard drive health while also maintaining high predictive performance.

The data used for predictive maintenance are typically collected from multiple sensors or industrial systems over a period of time, where historical and real-time data are combined as the input. The data contains information about the process, events, and alarms that occur along the industrial production line [4], [10].

The paper aims to contribute to the existing research on predictive maintenance with two contributions.

1) Demonstrating the potential of utilization of the manufacturing state of the industrial equipment alone to predict future maintenance. This approach has the advantage of creating the opportunity to utilize the results to change the production error repair process in real time and reduce costs drastically by repairing the production errors while the industrial equipment is still in production.

TABLE 1. A summary of recent papers on predictive maintenance. In column 3 a mark of (bb) indicates a so-called black-box method.

Reference	Year	ML method(s)	Equipment
[17]	2017	ARIMA	Slitting Machine
[8]	2018	**Random Forest**	Industrial Pumps
[7]	2018	**Random Forest**	Extruders
[18]	2018	Decision Trees, **Random Forest**, Bernoulli Naive Bayes, Gaussian Naive Bayes, **Artificial Neural Networks**	Industrial equipment for anode
[19]	2018	Generalized linear model, **Random Forest**, **Gradient Boosting**, **Deep Learning**.	Semiconductor
[20]	2018	**Random Forest**	Supermarket refrigeration systems
[21]	2019	Decision trees, **Random Forest**	Wind turbines
[22]	2019	**Deep Learning**	Computer numerical control machine
[23]	2021	**Gradient Boosting**, **Long short-term memory**	Wind turbines
[15]	2022	Decision Tree	Hard disk drivers

2) A framework obtaining interpretable machine learning models which identify and explain the features that predict the critical conditions on the equipment that indicates root causes. To do this, we propose to use variations of Classification Trees and Bayesian Networks for the prediction and root cause detection of critical errors that are expected to require repair within the lifetime of the industrial equipment. The framework contributes to a solution easily translated into a systematic tool identifying what production errors are critical. Furthermore, it ensures confidence that such a system continues to perform well if deployed in production.

III. THE INTERPRETABLE MODELS

This section describes the selected interpretable statistical models and the different approaches regarding the training pipeline modeling setup.

A. BAYESIAN NETWORKS

Bayesian Networks (*BNs*) are highly suitable for handling uncertainties and providing the means to decompose complex problems into simpler ones through their use of conditional probabilities. A basic feature of BN is inference from the probabilistic graphical models, which give specified probabilistic dependencies underlying a particular model using a directed acyclic graph (*DAG*). Consider now a probabilistic graphical model given as a graph in which nodes represent random variables and the arcs represent conditional dependence assumptions. The graph gives a compact representation of the joint probability distribution. In this formulation, the undirected graphical model is a Markov network, while a directed graphical model is called a Bayesian network or a Belief Network.

Consider a set of random variables $X = (X_1, X_2, \ldots, X_n)$ with $1 \leq i \leq n$. If there is an arc from node X_1 to X_2, then X_1 is the *parent* of X_2 and X_2 is the *child* [24], [25]. We define the set of parent nodes of X_i as $parents(X_i)$ and the BN can be specified as:

$$P(X) = \Pi_{i=1}^{n} P(X_i|parents(X_i)). \tag{1}$$

The features in the current paper represent different characteristics of the equipment found through inspection just after the equipment has been manufactured. The aim is to use the collected data to predict whether there is a fault or will be a fault at a future point in time. BNs are supervised classifiers and are also a reliable and interpretable model for discovering relationships among the variables in the system and their usage in areas of predictive analytics. We can apply Bayes' theorem (2) to the Bayesian classifier, which allows the probabilities of the model to be updated as new data are fed into the BN. In that way, Bayes' theorem is efficient to propagate information through the BN [25],

$$P(X|E) = \frac{P(E|X)P(X)}{P(E)} = \frac{P(E,X)}{\sum_X P(E,X)}. \tag{2}$$

Learning BNs involves two key stages; structural learning and parameter learning. There are many ways to learn the structure of the DAG. In the current paper, we used a score-based learning approach. This choice is based on the best performance in an out-of-sample cross validation evaluation. To do this, we first need to define the criterion to evaluate how well the BN fits the data, and then a search algorithm searches over the space of different DAGs for a structure achieving the best score. We used Hill-climbing as the search algorithm and BIC score as the scoring metric. Other approaches have been tested (e.g., Tabu Search, Grow-shrink, Naive Bayes), however Hill-climbing with BIC was the best performing algorithm based on the ROC AUC score in cross validation. Hill-climbing is a heuristic search approach, and it implements a greedy local search that starts from a disconnected DAG and then proceeds by iteratively performing single-edge manipulations that maximally increase the score. The search terminates once a local maximum is found. To avoid only local optima, the algorithm is restarted five times. After the structure is learned, the features not connected directly or indirectly to

the response variable are removed to obtain a simple, clear structure.

When the BN structure is completed, we fit the parameters of the local distributions based on MLE, which take the form of conditional probability tables [24], [25].

B. CLASSIFICATION TREES

Classification trees (*CTs*) are another simple and useful model for interpretation. As with BNs, CTs can be visualized.

First, the predictor space is divided into J distinct regions, $R_1, R_2, \ldots, R_J$. The approach corresponds to recursive binary splitting, which is a top-down approach, starting from the top of the tree, and then successively splitting the predictor space. A split is made between the values of a variable by selecting a threshold. The split is made with $\geq$ or $\leq$ for numeric variables and $=$ or $\neq$ for categorical variables. Each split is indicated via two new branches further down on the tree.

We use the Gini index as the criterion for making the binary splits. The Gini index is defined by:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}), \tag{3}$$

which is a measure of the total variance across the K classes. $\hat{p}_{mk}$ is the proportion of training observations in the mth region that are from the kth class. The Gini index dictates when a node split will occur in order to keep each node as pure as possible to reduce the total value of the Gini index. The Gini index can be weighted by multiplying the inner part of the sum with w_k, which is the selected weight for class k.

For every observation in region R_j, we gather the prediction by taking the mean or the mode of the response values for the training observation in R_j.

Parameter tuning of a cost complexity parameter is used to control the size of the tree and to prevent overfitting. The parameters are selected based on the best performance in cross validation. With cost complexity pruning, we consider a sequence of trees indexed by a non-negative tuning parameter α. For each value of α there corresponds a subtree $T \subset T_0$ such that:

$$\sum_{m=1}^{|T|} \sum_{\{i:x_i \in R_m\}} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \tag{4}$$

is as small as possible. Here $|T|$ is the number of terminal nodes of the tree T, R_m is the subset of the predictor space corresponding to the mth terminal node, and $\hat{y}_{R_m}$ is the predicted response associated with R_m [26].

IV. CASE COMPANY OVERVIEW AND DATA DESCRIPTION

Our research investigates a case study with a large-scale European manufacturer of industrial equipment. They manufacture large expensive products and also provide the majority of their customers with service contracts on the industrial equipment. All the products are rigorously investigated after production to ensure quality and integrity after they go into use. In this process, repairs are done to the equipment, while other issues are ignored as they pose no structural threat. All of this is collected and logged in a production database.

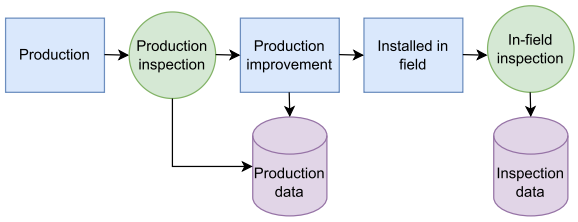


FIGURE 1. The inspection flow of the product's first five years and the stages of production and inspection data collection.

In the current paper, we only investigate the first inspection within the initial five-year period after manufacturing. Within the first five-year period, the equipment is inspected at least once. This process maps and describes all inspection errors on the equipment. An inspection error is anything on the equipment that needs to be noted; everything from cosmetic errors to structurally threatening severe inspection errors. All of this is collected and logged in an inspection database. The data-generating process is illustrated in Fig. 1.

The intended purpose is a clear insight into the relationship between production errors and errors identified during inspection that potentially are not previously known. Repairs completed in production cost only a fraction of the repairs done to the equipment in the field. We aim to help the manufacturing company to make valuable changes to its production process to prevent future expensive inspection errors by interpreting both the data and the training as interpretable and accurate models. This is predictive maintenance, but in reverse, to predict potential future errors to prevent future expensive repairs. The data from production has not yet been examined in this light at the manufacturing company. Thus the present analysis provides an opportunity to produce and uncover information about the production condition of the equipment's impact on predictive maintenance.

A. DATA DESCRIPTION

The case data set is constructed by joining the production data and the inspection data of the equipment. The case data set contains 227,996 observations and 29 variables. The features are anonymized measurements from the manufacturing state of the equipment. Each row in the data set represents an error in production. The features describe e.g., the type of equipment, the location of the production or the characteristics of the production error. The goal is to predict which characteristics of the production error will fail and occur as critical inspection errors in the later in-field inspections. The data set contains three different data types: numerical, categorical, and time-based. The characteristics of the data set are summarized in Table 2.

TABLE 2. Characteristics of the case data set.

Variables	Factors	Factor levels	Numeric	Integer	Kurtosis	Skewness	Missingness
25	12	2-14	9	4	2.1-769.9	-1.3-27.7	0-22%

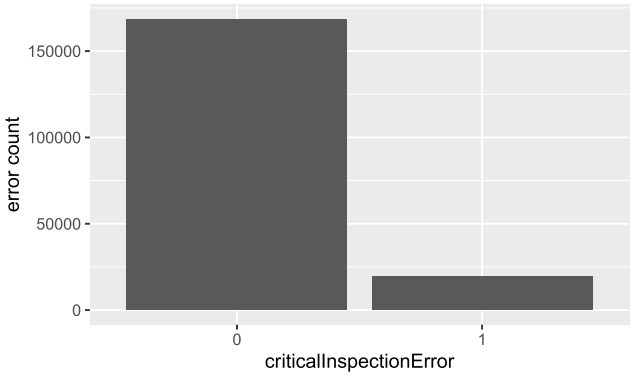


FIGURE 2. The distribution of the response variable *criticalInspectionError* in the training data set. 0 corresponds to *nomatch* and 1 corresponds to *match*.

The case data are messy due to the typical complexity of real-world data. We partition the data set into three separate data sets ranked according to the date of production, training (80%), validation (10%), and testing (10%).

1) THE RESPONSE VARIABLE

The response variable in the case data is a binary outcome variable taking the value *match* (1) if a production error matches with a critical inspection error, and *nomatch* (0) otherwise. Critical inspection errors either require repair at first inspection or are expected to require repair within the lifetime of the equipment. The partition of errors as critical is based on the error type, size, and severity. A match is determined based on the overlap in the placement measurements between the production errors and the inspection errors on the equipment. The training data set contains 4,300 different pieces of equipment. Among the inspection errors registered in the original inspection data set on these specific pieces of equipment, roughly half of these match with one or more errors in production. Approximately 12% of the errors from production match with a critical inspection error, and approximately half of these need repair at first inspection. The training data set contains 188,190 rows. The distribution of the response variable *criticalInspectionError* is plotted in Fig. 2. The response variable is imbalanced with *match* as the minority class.

V. FRAMEWORK

The framework for error classification and prediction through multiple stages of preprocessing, feature selection, BN or CT learning, and interpretable prediction is illustrated in Fig. 3. The goal is to provide a framework to allow a probabilistic model to learn from a large amount of data and generate interpretable insights. The BNs and the CTs are used to extract relationships between the variables, which provide insights into the errors found in the inspection as a prediction task based on production conditions. It is beneficial to create an appropriate model to represent the system for future analysis of predicting expensive errors that occur rarely.

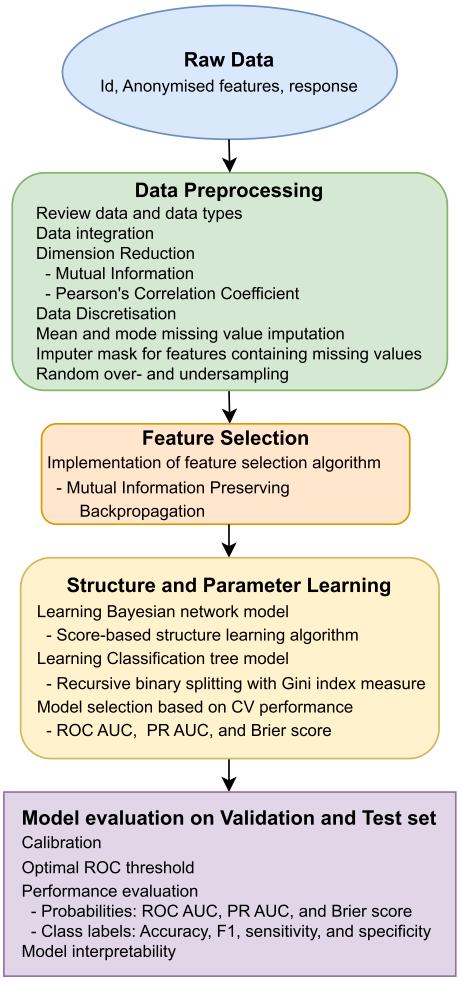


FIGURE 3. Design flow showing the preprocessing stages to obtain the appropriate processed data for input to train the models.

The case data set contains several similar variables, e.g., different variations of equipment product group, and location specification of production. As the data set contains many categorical variables, we perform a combination of Normalized Mutual Information and Pearson's Correlation Coefficient to detect multicollinearity. The threshold value for removing variables is ± 0.85.

A. STEP 1: DATA PREPROCCESING

Some manipulations of the case data set must be performed before applying the selected interpretable models.

1) DATA DISCRETIZATION

It is feasible to train BNs and CTs with a data set containing both discrete and continuous variables, but it is not necessarily efficient or suitable for BNs. It requires the continuous variables to be normally distributed, which they are far from being, with for example kurtosis values both below and above three. Therefore, the continuous variables are discretized unsupervised before training the BNs. The method used for this is an extension of mutual information, whereby as much covariance as possible is preserved compared to simpler and popular choices like quantile or interval discretization. The method is called Hartemink's Information-Preserving Discretization and it maximizes conditional mutual information (see below) concerning the rest of the data set. This is the reason for considering both the conditional dependencies and independencies between variables in the domain [27].

Formally, the conditional mutual information of two jointly discrete random variables $\mathbf{X}$ and $\mathbf{Y}$ with sets of possible outcomes $\{x_1, x_2, \ldots\}$ and $\{y_1, y_2, \ldots\}$, respectively, is defined in equation (5):

$$I(\mathbf{X}; \mathbf{Y}|\mathbf{Z}) = \sum_{(k=1)}^{n} \sum_{(i=1)}^{n} \sum_{(j=1)}^{n} p(z_k, x_i, y_j) \times \log\left(\frac{p(z_k), p(z_k, x_i, y_j)}{p(z_k, x_i)p(z_k, y_j)}\right), \quad (5)$$

where $p(z_k, x_i, y_j)$, $p(z_k, x_i)$, and $p(z_k, y_j)$ are the joint probability functions of $\mathbf{Z}$, $\mathbf{X}$, and $\mathbf{Y}$, respectively, and $p(z_k)$ is the marginal probability function of $\mathbf{Z}$. Here $\mathbf{Z}$ denotes the remaining dataset variables.

All the continuous variables are discretized into three categories. The number three is chosen out of the feasible numbers based on the out-of-sample F1 score in cross validation. With only discrete variables as inputs, the BNs conditional probability distributions are defined by conditional probability tables. This improves both the efficiency of inference and the interpretability of the BN model.

2) MEAN OR MODE IMPUTATION

As the case data set is strongly characterized by missing values, efforts are made to preserve as much information as possible. Incomplete cases causes lost information and biased estimates. Unless the missing data are missing completely at random (MCAR), it is desired to handle these without omitting them. As the missing data does not appear to be MCAR, a common practice is to impute missing values and then proceed as if the imputed values are true values [28], [29], [30], [31]. One of the most common procedures is mean or mode imputation. In [29] it is shown that mean imputation is completely appropriate and leads to a consistent estimation of the prediction function, which makes the imputation method very useful in practice. This is due to a supervised-learning setting, where the aim is to minimize a prediction risk by estimating a regression function. In [30], the method outperformed several other imputation approaches such as K-Nearest Neighbors and Iterative with and without a mask. The information of missingness can be relevant for predicting the outcome in cases where the outcome depends on missingness explicitly or the missingness itself carries information i.e. outcome is missing not at random (MNAR). For these reasons, it can be useful after imputation to add new binary features that encode whether a value was originally missing or not: the ''mask'' or one-hot encoding [28], [30], [31].

3) RANDOM OVER- AND UNDERSAMPLING

The distribution of the response variable causes a moderate degree of imbalance. The minority class is the value match; a match with an inspection error. BNs and CTs, among many other machine learning models, suffer from low performance because of the extreme unequal distribution in the response variable as they assume balanced class distributions. The models become biased toward the majority class, resulting in biased predictions and misleading accuracies. Score-based algorithms assign a score to each candidate BN and try to maximize it with a heuristic search algorithm. Greedy search algorithms such as hill-climbing are driven by minimizing the BIC score, which estimates the average test set RSS across the observations. Thus, this causes minimization of the overall RSS to which the minority class contributes far less than the majority class. The data set is modified into a balanced distribution using random oversampling and random undersampling, without adding synthetic observations. The concept of random oversampling is balancing out the data by randomly oversampling the minority class. The method does not lead to information loss but might cause overfitting. On the other hand, random undersampling balances out the data by randomly undersampling the majority class, which do cause information loss [24].

B. STEP 2: FEATURE SELECTION

Mutual Information Preserving Backpropagation is utilized as a supervised model selection method. In the first iteration, all variables are used for training. In the second iteration, the variable with the lowest normalized mutual information score with the response variable is removed from the set of training variables, and so on.

All the generated models are evaluated with a rolling window cross validation setup, illustrated in Fig. 4.

All the models are evaluated based on ROC AUC, PR AUC, and Brier scores. In this step, the predicted probabilities of the models are evaluated. When dealing with an imbalanced classification problem, the choice of an appropriate metric must be investigated. The ROC AUC represents the classifier's ability to separate the classes, the PR AUC represents the ability to predict the minority class and the Brier score measures the accuracy of the probabilistic predictions. After the features are selected, the structure is learned.

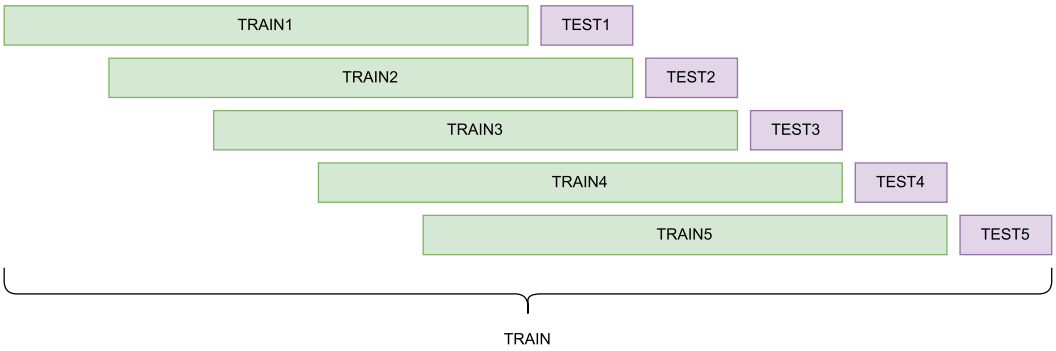


FIGURE 4. The rolling window cross validation setup.

C. TRAINING PIPELINES

Based on the presented statistical models and the different pipeline elements, we present the seven different model approaches in the current paper. The training pipelines for each model are listed in Fig. 5.

D. STEP 3: CALIBRATION AND OPTIMAL THRESHOLD

The response variable is imbalanced, which requires strategies for calibration of the predicted probabilities and how to select an appropriate threshold.

1) CALIBRATION

As we use random over- and undersampling and also a weighted loss function to balance the train data set, the predicted probabilities are most likely not well-calibrated. The probability values are not representative of the true probabilities. This is an issue in production planning, as the predicted probabilities need to be trustworthy when determining potential risks. Platt scaling [33] is probably one of the most widely known approaches for probability calibration. Given the margins of real scalars, the scores can be transformed into probability estimates with logistic regression. Thus, Platt scaling has the same parameters as a univariate logistic regression model. The predictive function is as in equation (6),

$$g(s; w, b) = \frac{1}{1 + exp(-w \cdot s - b)}, \tag{6}$$

where $w \in \mathbb{R}$ is the shape parameter and $b \in \mathbb{R}$ is the location parameter. The parameters are estimated by maximizing the log-likelihood on the validation set [32], [33].

2) THRESHOLD SELECTION

When the data is calibrated, a decision threshold needs to be set when operating in uncertain conditions. When predicting class labels, the default threshold is 0.5 in binary cases. However, it may not represent an optimal interpretation of the predicted probabilities. There are four reasons for this [34]: 1) The predicted probabilities are not calibrated, 2) the metric used to train the model is different from the metric used to evaluate a final model, 3) the class distribution is severely skewed, and 4) the cost of one type of misclassification is more important than another type of misclassification.

Thus, there is often a need to change the default decision threshold when interpreting the predictions of a model. All classifiers in current paper generate positive or negative class predictions by applying a threshold to a predicted probability. The choice of the threshold will have a significant impact on the trade-offs of positive and negative errors [35]. The selected thresholds are the ROC optimal thresholds. The ROC optimal threshold method selects the threshold values that balance the true positive rate with the false positive rate. Thus both classification classes are equally important. This is due to the production costs, as we want to predict as many inspection errors correctly as possible, without repairing everything in production. A ROC curve is a diagnostic plot evaluating the set of probability predictions made by the model on the validation data set. The false positive rate is plotted against the true positive rate. The curve is a useful tool for understanding the trade-off between the true positive rate and the false positive rate for different thresholds values. Based on the curve, a threshold for the optimal balance between false positives and true positives can be obtained. This is determined by optimizing the Geometric Mean ($\bar{G}$) in equation (7), which is a useful metric for imbalanced classification [36],

$$\bar{G} = \sqrt{\text{Sensitivity} \cdot \text{Specificity}}. \tag{7}$$

Other approaches to selecting the threshold have been tested. E.g. the PR AUC optimal threshold. However, it tends to select a threshold that is too sensitive, which results in a large proportion of false positives. False positives result in potentially having to spend time and money in production correcting errors that will not pose a future threat to the equipment. There is thus a balance between having a high recall score as well as a balance between true positives and false positives.

E. THE EXPERIMENTAL SETUP

The experimental setup is listed below.

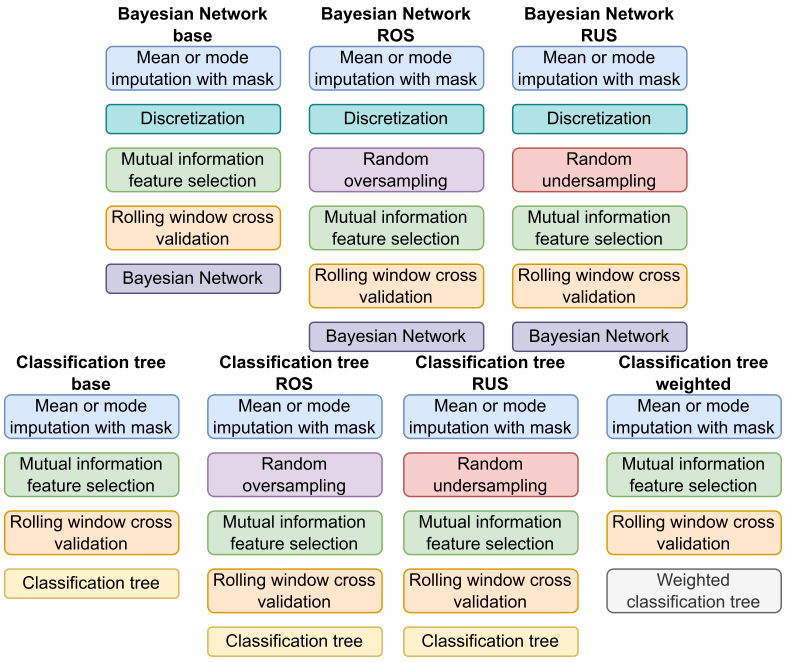


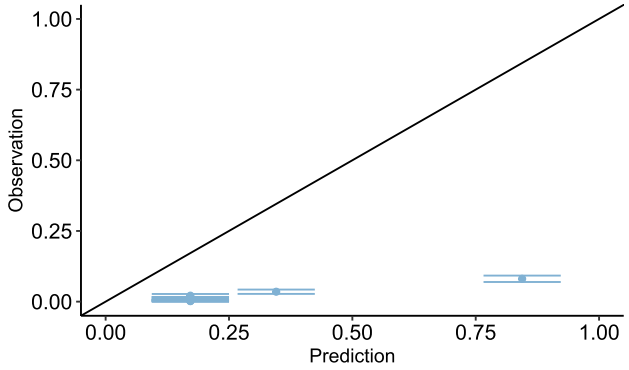
FIGURE 5. Training pipelines for each model.

TABLE 3. Performance table of all models presented in Fig. 5 for each step in the experimental setup. The best performing model based on F1 score is bolded.

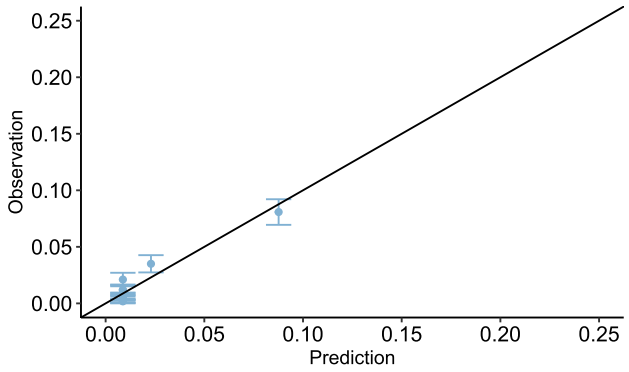
Cross Validation								
Model	ROC AUC	PR AUC	Brier	Calibrated Brier	Accuracy	F1	Sensitivity	Specificity
BN base	0.854	0.250	0.068	-	-	-	-	-
BN ROS	0.834	0.244	0.152	-	-	-	-	-
BN RUS	0.830	0.171	0.149	-	-	-	-	-
CT base	0.836	0.330	0.063	-	-	-	-	-
CT ROS	0.835	0.119	0.128	-	-	-	-	-
CT RUS	0.834	0.119	0.122	-	-	-	-	-
CT Weighted	0.827	0.212	0.125	-	-	-	-	-
AutoML	0.884	0.379	0.059	-	-	-	-	-
Validation Set								
Model	ROC AUC	PR AUC	Brier	Calibrated Brier	Accuracy	F1	Sensitivity	Specificity
BN base	0.835	0.062	0.027	-	0.679	0.090	0.880	0.676
BN ROS	0.835	0.093	0.124	0.019	0.679	0.090	0.880	0.676
BN RUS	0.858	0.088	0.140	0.019	0.738	0.110	0.885	0.735
CT base	0.739	0.073	0.023	-	0.806	0.106	0.637	0.809
CT ROS	0.741	0.029	0.110	0.017	0.806	0.106	0.637	0.809
CT RUS	**0.724**	**0.025**	**0.116**	**0.017**	**0.863**	**0.135**	**0.592**	**0.868**
CT Weighted	0.741	0.029	0.110	0.017	0.809	0.106	0.637	0.809
AutoML	0.920	0.189	0.019	-	0.826	0.159	0.912	0.825
Test Set								
Model	ROC AUC	PR AUC	Brier	Calibrated Brier	Accuracy	F1	Sensitivity	Specificity
BN base	0.804	0.043	0.019	-	0.723	0.050	0.743	0.723
BN ROS	0.696	0.008	-	0.007	0.692	0.029	0.706	0.692
BN RUS	0.787	0.034	-	0.010	0.800	0.068	0.738	0.801
CT base	0.740	0.056	0.014	-	0.826	0.065	0.608	0.828
CT ROS	0.738	0.030	-	0.010	0.828	0.065	0.608	0.830
CT RUS	**0.715**	**0.029**	**-**	**0.010**	**0.890**	**0.090**	**0.553**	**0.893**
CT Weighted	0.738	0.016	-	0.010	0.826	0.065	0.608	0.828
AutoML	0.891	0.084	0.034	-	0.523	0.039	0.979	0.518

1) **Cross Validation**: Mutual Information Preserving Back-propagation for feature selection and parameter tuning, based on out-of-sample performance.

2) **Validation Set**: The models selected in cross validation are evaluated on the validation set. The predicted probabilities are, if necessary, calibrated



(a) Calibration curve of the CT RUS before Platt's scaling.



(b) Calibration curve of the CT RUS after Platt's scaling.

FIGURE 6. Calibration curves of the CT RUS model before and after Platt's scaling.

with Platt's scaling and the ROC optimal threshold is calculated.

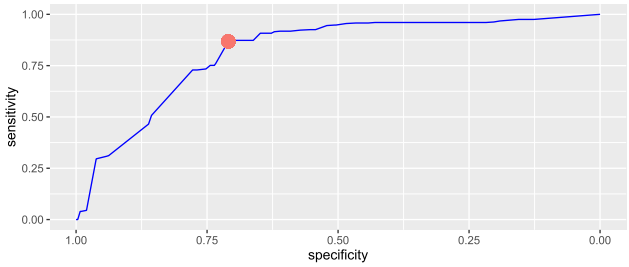
3) **Test set**: The models selected in cross validation and their calibration and thresholds selected on the validation set are finally evaluated on the test data set.

VI. RESULTS

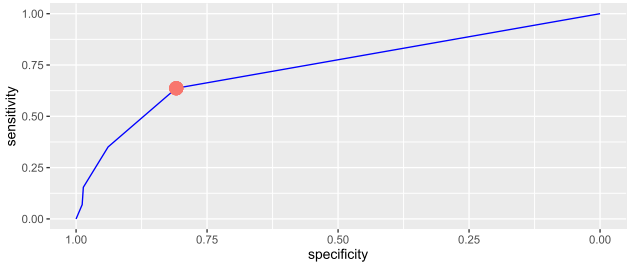
In this section, the correlation between production errors and critical inspection errors is evaluated. Critical inspection errors are assessed as either requiring repair at the first inspection or requiring repair during the lifetime of the equipment. The performance metrics from the three steps in the experimental setup are in Table 3.

The performance metrics ROC AUC, PR AUC, and Brier score for the best-performing models in the cross validation are summarized in Table 3. Generally, the selected BNs have higher ROC AUC scores than the selected CTs. However, there is a slight tendency for the CTs to have a lower Brier score. The value of the ROC AUC is high for all the models and indicates a relatively strong correlation between errors in production and critical errors in the first inspection.

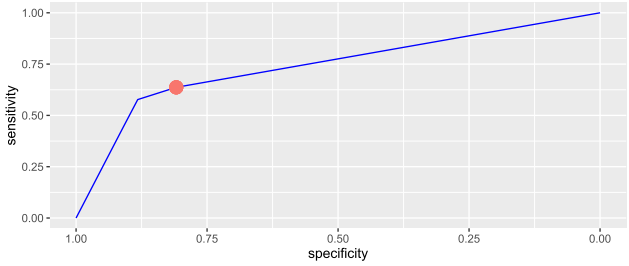
Based on the selected feature spaces for each model and the cost complexity parameter, the models are refitted on the latest two-thirds of the training data set and evaluated on the validation set. The performance metrics are listed in Table 3.



(a) The ROC curve of the BN RUS predicted probabilities. The red dot marks the ROC optimal threshold.



(b) The ROC curve of the CT base predicted probabilities. The red dot marks the ROC optimal threshold.



(c) The ROC curve of the CT RUS predicted probabilities. The red dot marks the ROC optimal threshold.

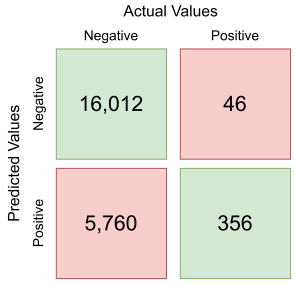
FIGURE 7. The ROC curve for the predicted probabilities for the BN RUS, CT base, and CT RUS models.

On the validation set, there has been a considerable reduction in the ROC AUC measure for the CTs. On the other hand, the BNs maintain closer to the same level in both ROC AUC and Brier scores. In addition, PR AUC has decreased significantly for the two model approaches compared with performance on the training data set in cross validation. The decrease in PR AUC indicates that the models are less effective at identifying the positive class.

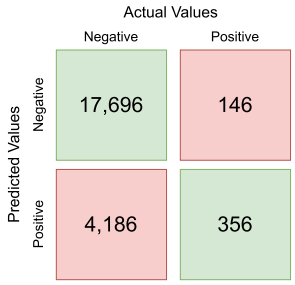
In Fig. 6b the calibration curves before and after Platt's scaling of the CT RUS are shown. Platt's scaling improves the underconfidence well, as the probabilities lie closer to the center on the diagonal. The improvement in the Brier score also appears in Table 3. The BN RUS is the best model based on ROC AUC, but the CTs all have lower Brier scores. The CTs predict probabilities closer to the true probabilities than the other simple models in the analysis.

Fig. 7a, 7b, and 7c are the ROC curves of the predicted probabilities and the ROC optimal thresholds of, respectively, BN RUS, the CT base, and the CT RUS.

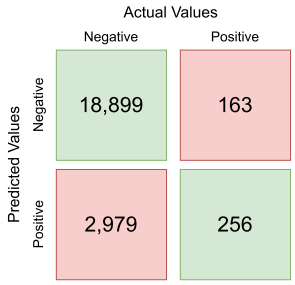
With the selected thresholds, we can evaluate the models' ability to classify the errors correctly. Table 3 shows the



(a) The confusion matrix of the BN RUS model.



(b) The confusion matrix of the CT base model.



(c) The confusion matrix of the CT RUS model.

FIGURE 8. The confusion matrices for the BN RUS, CT base, and CT RUS models on the validation set.

classifiers' accuracy, F1 score, sensitivity, and specificity. The CT RUS has the highest F1 score and accuracy.

To evaluate the ability of the selected models to classify, confusion matrices of the predicted classes on the validation set for, respectively, BN RUS, the CT base, and the CT RUS are now considered in Fig. 8. The BNs generally have good precision on the positive class but also many false positives. Thus the models are too sensitive as classifiers. This can also be read in the BN models' specificity, which is significantly lower than the corresponding for the CTs. The CT RUS provides a much more desirable confusion matrix, where the proportion of false negatives and false positives overall is considerably smaller than the other models; approximately one true positive to 12 false positives. In addition, the model can classify 60% of the positive class correctly. Note, the model is not expected to classify the entire positive class correctly, as other external factors might impact inspection errors.

Fig. 9 is the DAG structure of the BN RUS, the CT base, and the CT RUS. The included features vary across the models. Common to both CTs is the equipment V7 being the root node, contributing to the greatest gain to the models. All the models have a simple and easily interpretable structure. We focus on the CT RUS, which outperforms the other simple models. The depth of the CT is three, resulting in six different decision splits as leaf nodes along the values of the features.

If the classification model with RUS is reevaluated after scaling and moving the threshold, it is possible to set up the following three scenarios for the model to classify a match (cf. Fig. 9).

Scenario 1:
1) V7 not equal to f.
2) V8 less than g.
3) V9 not equal to h.

Scenario 2:
1) V7 equal to f.
2) V6 less than e.
3) V1 greater than or equal to a.

Scenario 3:
1) V7 equal to f.
2) V6 greater than or equal to e.

All prior steps are now evaluated on the test set, and all previous evaluation metrics are summarized for each model in Table 3. Again, the CT RUS outperforms the other simple models, both in relation to probabilities and class labels.

Fig. 10 is the confusion matrices of the predicted class labels for, respectively, the BN RUS, the CT base, and the CT RUS. The BN RUS has significantly more false negatives than the other models. The CT RUS again provides a much more desirable confusion matrix, where the proportion of false negatives and false positives overall is considerably smaller than the other models. Again, there is approximately one true positive to 19 false positives. However, the ability to classify the positive class has dropped slightly on the test set, but the model still captures 55%.

In the experiment, we obtain a CT RUS with only three scenarios for which it predicts a match with a critical inspection error, with respectively two, three, and two conditions. The model correctly classifies the inspection errors approximately 60% of the time on the validation set and 55% on the test set. On the validation data set, there is almost 1 true positive to 12 false positives and 1 to 19 on the test data set. Thus, the production errors greatly impact the presence of future critical inspection errors but have significantly decreased performance compared to the first experiment. With few conditions, we can narrow down and correctly leave out approximately 88% of all errors as future issues in the first inspection as critical errors, given no changes to the current production process.

Some of the equipment is inspected a second time within the five-year warranty period. The data availability of the second inspection on the equipment is limited, as only 18.6% have a second inspection. However, if we consider these 18.6%, it seems like the model is ahead of the predictions before the production error evolves into a match in the data set. If the data from the second inspection is considered, 302 of the false positives evolve into critical inspection errors at the second inspection. Thus considering the second

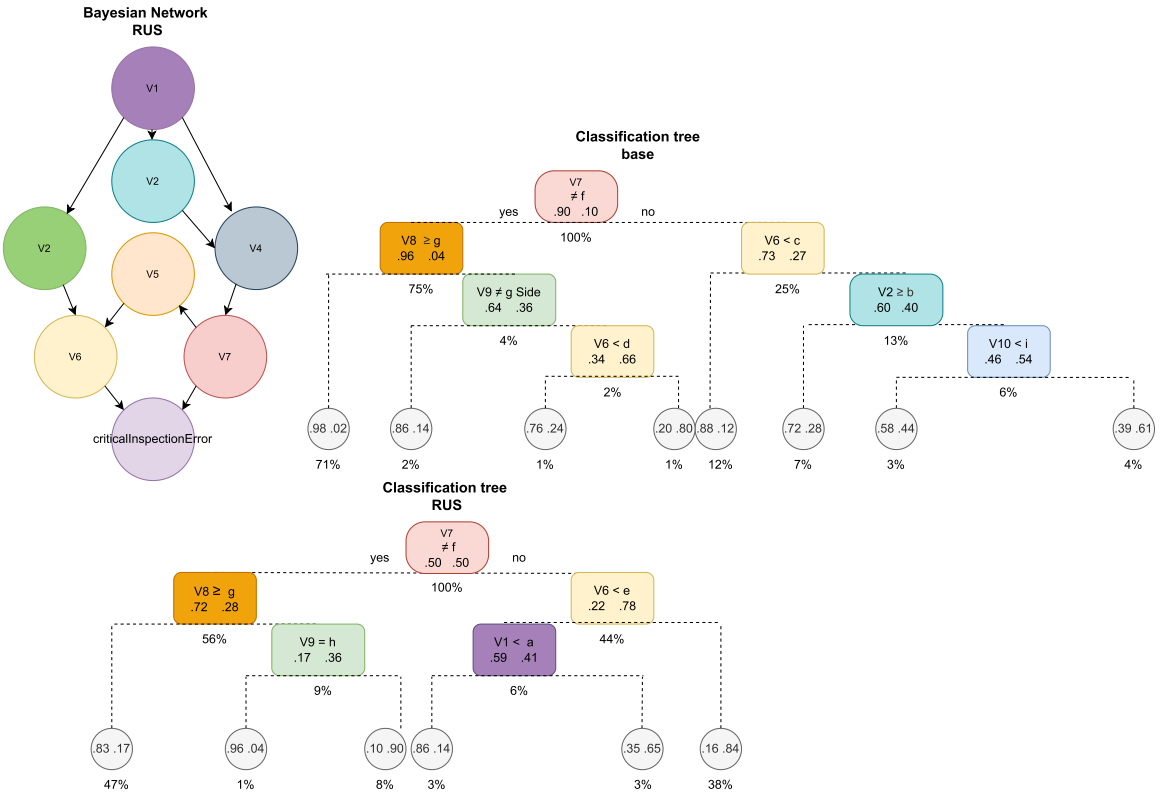


FIGURE 9. The structure of the BN RUS, the CT base, and the CT RUS.

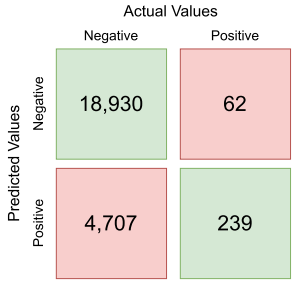
inspection, the model is able to correctly identify 433 true positives and reduce the false positives to 2,219 observations. Thus, the model can correctly detect one critical error within the first or second inspection against five false negatives. There is a time frame of five years within which the manufacturing company performs the inspection for the first time, unless the situation is problematic. The result is strong, as the manufacturing company expects the critical inspection errors that occur within the first five years to evolve into errors requiring repair within the equipment's lifetime. The model's results are good in terms of business, as production errors only cost a fraction of the cost of repair in the field. The result is not only beneficial as a contribution to research but has great potential to support quality controllers in corresponding manufacturers as the set of rules from the model is both easy and cheap to implement. However, the data from the second inspection is not included in the model, as we want to prevent constructing data that is untrustworthy. Equipment inspected more than once is often equipment considered problematic, and the data from the second inspection is considered biased.

Based on the experiment, there is enough signal in the production data to measure the correlation to critical inspection errors. Despite a simple learning structure and challenges with data generation, we managed to generate useful predictive performance. There is great potential in utilizing the easily interpretable model in collaboration with production planners. This is because it enables measures that can help to improve the risk of later maintenance on the products.

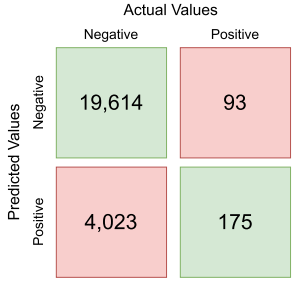
The case study has shown how to utilize the manufacture state of industrial equipment alone to predict future maintenance. This is distinct from the classical approach, as it does not take any historical or in-field data into account. Which creates the opportunity to utilize the results to change the production error repair process and reduce cost drastically by repairing the production errors while the equipment is still in production. The paper thus contributes by demonstrating another approach to the predictive maintenance field.

The framework presented in the paper applies interpretable models to predict future maintenance, which is another contribution to the field. This is achieved by using simple and interpretable models, which makes it possible to take action to prevent future inspection errors. In research, the common practice is to use black-box models to predict future maintenance. However, they only predict the condition of the equipment but cannot explain and prevent causes for the condition.

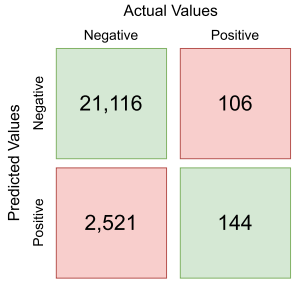
In [14] and [16], the researchers utilize optimal decision trees for predictive maintenance and benchmark the result against implementations with different popular black-box methods. With the method, they construct interpretable CTs with approximately the same performance as the black-box methods. In this regard, it could be interesting to expand the



(a) The confusion matrix of the BN RUS model.



(b) The confusion matrix of the CT base model.



(c) The confusion matrix of the CT RUS model.

FIGURE 10. The confusion matrices for the BN RUS, CT base, and CT RUS models on the test set.

paper to consider optimal CTs. However, the optimal CTs are not necessarily better out of sample.

VII. CONCLUSION

Current paper focused on the possibility of carrying out predictive maintenance in an industrial equipment manufacturing company based on production data with interpretable machine learning models. A gap in the existing predictive maintenance research field was detected in using the manufacture state of the industrial equipment to predict future maintenance. Combined with interpretable models, current research contributes to comprehending insights about the features' interaction in indicating impending failure and obtaining complete confidence that such a system continues to perform well if deployed in production.

Current paper successfully constructed a framework that applied interpretable models to predict future conditions of the equipment, which made it possible to take action to prevent future inspection errors. The manufacture state of the industrial equipment could be translated into concrete actions on the production side with a simple set of rules. The paper used Bayesian Networks with hill climbing structure learning and Classification Trees with ROC AUC as a loss function as the interpretable machine learning models. Various pipeline setups have been used to obtain models that were both simple and interpretable and, at the same time, could be applied to predict an imbalanced response variable. The models evaluated all had a simple structure.

Current paper only contributed to the investigation of a single case. Several similar case studies in different industries must be performed to investigate the detected gap further. The evidence from this single case can not stand alone.

REFERENCES

[1] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 3, May 2021, Art. no. 160, doi: 10.1007/s42979-021-00592-x.

[2] I. P. Colo, C. S. Sueldo, M. De Paula, and G. G. Acosta, "Intelligent approach for the industrialization of deep learning solutions applied to fault detection," *Expert Syst. Appl.*, vol. 233, Dec. 2023, Art. no. 120959, doi: 10.1016/j.eswa.2023.120959.

[3] Z. Bouzidi, L. S. Terrissa, N. Zerhouni, and S. Ayad, "QoS of cloud prognostic system: Application to aircraft engines fleet," *Eur. J. Ind. Eng.*, vol. 14, no. 1, pp. 34–57, 2020, doi: 10.1504/EJIE.2020.105080.

[4] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Comput. Ind. Eng.*, vol. 150, Dec. 2020, Art. no. 106889, doi: 10.1016/j.cie.2020.106889.

[5] H. Toumi, A. Meddaoui, and M. Hain, "The influence of predictive maintenance in industry 4.0: A systematic literature review," in *Proc. 2nd Int. Conf. Innov. Res. Appl. Sci., Eng. Technol. (IRASET)*, Meknes, Morocco, Mar. 2022, pp. 1–13, doi: 10.1109/iraset52964.2022.9737901.

[6] V. P. Koutras, S. Malefaki, and A. N. Platis, "Opportunistic maintenance on the automatic switching mechanism of a two-unit multi-state system," *Eur. J. Ind. Eng.*, vol. 15, no. 5, pp. 616–642, 2021, doi: 10.1504/EJIE.2021.10035757.

[7] K. Mulrennan, J. Donovan, L. Creedon, I. Rogers, J. G. Lyons, and M. McAfee, "A soft sensor for prediction of mechanical properties of extruded PLA sheet using an instrumented slit die and machine learning algorithms," *Polym. Test.*, vol. 69, pp. 462–469, Aug. 2018, doi: 10.1016/j.polymertesting.2018.06.002

[8] I. Amihai, R. Gitzel, A. M. Kotriwala, D. Pareschi, S. Subbiah, and G. Sosale, "An industrial case study using vibration data and machine learning to predict asset health," in *Proc. IEEE 20th Conf. Bus. Informat. (CBI)*, Vienna, Austria, vol. 1, Jul. 2018, pp. 178–185, doi: 10.1109/CBI.2018.00028.

[9] J. Deutsch and D. He, "Using deep learning-based approach to predict remaining useful life of rotating components," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 1, pp. 11–20, Jan. 2018, doi: 10.1109/TSMC.2017.2697842.

[10] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. D. P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106024, doi: 10.1016/j.cie.2019.106024.

[11] M. Khan, A. Ahmad, F. Sobieczky, M. Pichler, B. A. Moser, and I. Bukovský, "A systematic mapping study of predictive maintenance in SMEs," *IEEE Access*, vol. 10, pp. 88738–88749, 2022, doi: 10.1109/access.2022.3200694.

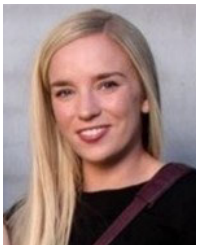
[12] M. Qasim, M. Khan, W. Mehmood, F. Sobieczky, M. Pichler and B. Moser, "A comparative analysis of anomaly detection methods for predictive maintenance in SME," in *Proc. 33rd Int. Conf. Database Expert Syst. Appl. (DEXA Workshops)*, 2022, pp. 22–31.

[13] J. Sharma, M. L. Mittal, and G. Soni, "Condition-based maintenance using machine learning and role of interpretability: A review," *Int. J. Syst. Assurance Eng. Manage.*, Dec. 2022, doi: 10.1007/s13198-022-01843-7.

[14] D. Bertsimas and J. Dunn, *Machine Learning Under a Modern Optimization Lens*, 1st ed. Waltham, MA, USA: Dynamic Ideas LLC, 2019.

[15] A. Ferraro, A. Galli, V. Moscato, and G. Sperlì, "Evaluating eXplainable artificial intelligence tools for hard disk drive predictive maintenance," *Artif. Intell. Rev.*, vol. 56, no. 7, pp. 7279–7314, Jul. 2023, doi: 10.1007/s10462-022-10354-7.

[16] M. Amram, J. Dunn, J. J. Toledano, and Y. D. Zhuo, "Interpretable predictive maintenance for hard drives," *Mach. Learn. With Appl.*, vol. 5, Sep. 2021, Art. no. 100042, doi: 10.1016/j.mlwa.2021.100042.
[17] A. Kanawaday and A. Sane, "Machine learning for predictive maintenance of industrial machines using IoT sensor data," in *Proc. 8th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Nov. 2017, pp. 87–90, doi: 10.1109/ICSESS.2017.8342870.
[18] N. Kolokas, T. Vafeiadis, D. Ioannidis, and D. Tzovaras, "Forecasting faults of industrial equipment using machine learning classifiers," in *Proc. Innov. Intell. Syst. Appl. (INISTA)*, Thessaloniki, Greece, Jul. 2018, pp. 1–6, doi: 10.1109/INISTA.2018.8466309.
[19] S. Butte, A. R. Prashanth, and S. Patil, "Machine learning based predictive maintenance strategy: A super learning approach with deep neural networks," in *Proc. IEEE Workshop Microelectron. Electron Devices (WMED)*, Apr. 2018, pp. 1–5, doi: 10.1109/WMED.2018.8360836.
[20] K. Kulkarni, U. Devi, A. Sirighee, J. Hazra, and P. Rao, "Predictive maintenance for supermarket refrigeration systems using only case temperature data," in *Proc. Annu. Amer. Control Conf. (ACC)*, Milwaukee, WI, USA, Jun. 2018, pp. 4640–4645, doi: 10.23919/ACC.2018.8431901.
[21] J.-Y. Hsu, Y.-F. Wang, K.-C. Lin, M.-Y. Chen, and J. H. Hsu, "Wind turbine fault diagnosis and predictive maintenance through statistical process control and machine learning," *IEEE Access*, vol. 8, pp. 23427–23439, 2020, doi: 10.1109/ACCESS.2020.2968615.
[22] B. Luo, H. Wang, H. Liu, B. Li, and F. Peng, "Early fault detection of machine tools based on deep learning and dynamic identification," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 509–518, Jan. 2019, doi: 10.1109/TIE.2018.2807414.
[23] W. Udo and Y. Muhammad, "Data-driven predictive maintenance of wind turbine based on SCADA data," *IEEE Access*, vol. 9, pp. 162370–162388, 2021, doi: 10.1109/ACCESS.2021.3132684.
[24] M. Scutari, "Learning Bayesian networks with the bnlearn R package," *J. Stat. Softw.*, vol. 35, no. 3, pp. 1–22, 2010, doi: 10.18637/jss.v035.i03.
[25] C. M. Carbery, R. Woods, and A. H. Marshall, "A Bayesian network based learning system for modelling faults in large-scale manufacturing," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Lyon, France, Feb. 2018, pp. 1357–1362, doi: 10.1109/ICIT.2018.8352377.
[26] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 1st ed. New York, NY, USA: Springer, 2013.
[27] A. J. Hartemink, "Principled computational methods for the validation and discovery of genetic regulatory networks," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2001.
[28] J. Poulos and R. Valle, "Missing data imputation for supervised learning," *Appl. Artif. Intell.*, vol. 32, no. 2, pp. 186–196, Apr. 2018, doi: 10.1080/08839514.2018.1448143.
[29] J. Josse, N. Prost, E. Scornet, and G. Varoquaux, "On the consistency of supervised learning with missing values," 2019, pp. 1–43, *arXiv:1902.06931*.
[30] A. Perez-Lebel, G. Varoquaux, M. Le Morvan, J. Josse, and J.-B. Poline, "Benchmarking missing-values approaches for predictive models on health databases," *GigaScience*, vol. 11, pp. 1–22, Apr. 2022, doi: 10.1093/gigascience/giac013.
[31] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*, 1st ed. Boca Raton, FL, USA: Taylor & Francis, 2019.
[32] T. S. Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: A survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112, no. 9, pp. 3211–3260, Sep. 2023, doi: 10.1007/s10994-023-06336-7.
[33] J. Platt, "Probabilities for SV machines," in *Advances in Large-Margin Classifiers*. Cambridge, MA, USA: MIT Press, 2000, pp. 61–74.
[34] J. Brownlee, "A gentle introduction to threshold-moving for imbalanced classification," Mach. Learn. Mastery, San Francisco, CA, USA. Accessed: Sep. 13, 2023. [Online]. Available: https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/
[35] A. Fernndez, S. Garca, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning From Imbalanced Data Sets*, 1st ed. New York, NY, USA: Springer, 2018.
[36] Z. H. Hoo, J. Candlish, and D. Teare, "What is an ROC curve?" *Emergency Med. J.*, vol. 34, no. 6, pp. 357–359, Jun. 2017, doi: 10.1136/emermed-2017-206735.



NANNA BURMEISTER received the B.Sc. and M.Sc. degrees in mathematics-economics from Aalborg University, Denmark, in 2020 and 2023, respectively. She is currently an Industrial Data Scientist in the wind industry. Her studies are specialized in operations research and are mainly focused on machine learning methods for predictive modeling and simulations.

RASMUS DOVNBORG FREDERIKSEN received the B.Sc. and M.Sc. degrees in mathematics-economy from Aalborg University, Denmark, in 2018, where he is currently pursuing the Ph.D. degree. He has since spent five years working in the wind industry. The main focus of his work are with machine learning methods for predictive modeling. He has undertaken several different projects within the data domain in the wind industry, from simulation models and BI dashboards to data analysis and ML/AI development. He is the holder of one patent based on the master's thesis.

ESBEN HØG was a Visiting Scholar with the Department of Statistics, University of California at Berkeley, in 1997. He initiated the Mathematics-Economics Program, Aalborg University, in 2009, and has been heading this program since. Before that, he was with the Science Park, Aarhus, and the Aarhus School of Business, Aarhus University, as an Associate Professor. He is currently an Associate Professor with the Department of Mathematical Sciences, Aalborg University. His recent publications are within using the theory and methods of copulas to analyze wind power futures, and within volumetric risk in wind power trading as well as within risk and fair pricing. His research interests include econometrics, energy markets, financial econometrics, risk management, quantitative finance, and mathematical statistics and OR in general.

PETER NIELSEN received the M.Sc. and Ph.D. degrees in engineering from Aalborg University, Denmark, in 2005 and 2008, respectively.

He has been heading the Operations Research Group, since 2011. He is currently an Associate Professor with the Department of Materials and Production, Aalborg University. He has coauthored more than 50 journal articles and numerous contributions to conferences papers and books. His research interests include artificial intelligence for autonomous (cyber-physical) systems, with a special emphasis on NP-hard problems that need to be solved in real-time or near real-time, and unmanned systems and their applications. He was the Chief Editor of *Production and Manufacturing Research*, for nine years, from 2013 to 2021.