Turkish Journal of Electrical Engineering and Computer Sciences

Volume 32 | Number 4 Article 4

7-26-2024

A new approach: ordinal predictive maintenance with ensemble binary decomposition (OPMEB)

OZLEM ECE YUREK

DERYA BIRANT

Recommended Citation

YUREK, OZLEM ECE and BIRANT, DERYA (2024) "A new approach: ordinal predictive maintenance with ensemble binary decomposition (OPMEB)," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 32: No. 4, Article 4. https://doi.org/10.55730/1300-0632.4086
Available at: https://journals.tubitak.gov.tr/elektrik/vol32/iss4/4

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2024) 32: 534 – 554
© TÜBİTAK
doi:10.55730/1300-0632.4086

A new approach: ordinal predictive maintenance with ensemble binary decomposition (OPMEB)

Özlem Ece YÜREK[1], **Derya BİRANT**[2,*]

[1]The Graduate School of Natural and Applied Sciences, Dokuz Eylül University, İzmir, Turkiye
[2]Department of Computer Engineering, Faculty of Engineering, Dokuz Eylül University, İzmir, Turkiye

Received: 22.02.2024 • **Accepted/Published Online:** 04.06.2024 • **Final Version:** 26.07.2024

Abstract: Predictive maintenance (PdM), a fundamental element of modern industrial systems, employs machine learning to monitor equipment conditions, estimate failure probabilities, and optimize maintenance schedules. Its core objective is to enhance equipment reliability, extend lifespan, and minimize costs through data-driven insights by enabling efficient maintenance scheduling, reducing downtime, and optimizing resource allocation. In this paper, we propose a novel ordinal predictive maintenance with ensemble binary decomposition (OPMEB) method for the PdM domain, considering the hierarchical nature of class labels reflecting the machine's health status, including categories like healthy, low risk, moderate risk, and high risk. The proposed OPMEB method was validated by executing on the C-MAPSS, AI4I 2020, and a real-world hydraulic system's condition datasets. The experimental outcomes were evaluated with four distinct metrics: accuracy, recall, precision, and F-measure. The findings showed the improvement in the model's predictive capabilities achieved by the proposed approach when compared to the traditional ordinal classification algorithm. Furthermore, the experimental results demonstrated the superior performance of the OPMEB method over other ordinal binary decomposition methods, including OneVsAll, OneVsFollowers, and OneVsNext.

Key words: Predictive maintenance, ordinal classification, binary decomposition, machine learning, classification, ensemble learning

1. Introduction

Management of maintenance planning and optimization is a very critical issue in various industry areas. Several maintenance strategies have been proposed to construct an effective solution to schedule maintenance properly and to ensure the reliability and safety of the systems by minimizing downtime. The most common strategies can be categorized into three approaches [1]. Run-to-failure (R2F) is the most basic strategy, where maintenance occurs only when a machine's components break down. This approach leads to long shutdown times and unplanned maintenance actions, making it very costly and the least effective option. The second strategy, preventive maintenance (PvM), has been used as a solution to these problems. PvM schedules the maintenance at planned time intervals, preventing unexpected failures and downtimes. However, it can result in maintenance actions that occur either too early or too late, leading to inefficient use of components and increased operating costs. In response to the growing industrial demand for efficiency, availability, cost-effectiveness, and safety, a third major maintenance strategy has evolved: predictive maintenance (PdM). PdM predicts the health status of machine components to determine the optimal time for maintenance, ensuring more appropriate maintenance

*Correspondence: derya@cs.deu.edu.tr

decisions. The main purpose of the PdM strategy is to increase the useful life of the components, save cost and energy by reducing fault rates, and maximize the production and operational availability of components and systems. Fault detection, diagnosis, and prognosis are the major principles of PdM. The PdM approach is capable of detecting a failure that will occur, identifying a specific type of failure, and predicting the remaining useful life (RUL) of the machine's components. High-accuracy forecasting through machine learning algorithms is crucial across various domains, as these predictions offer a significant and positive contribution to decision and policy makers in diverse fields [2, 3]. For instance, high-accuracy predictions play a pivotal role in constructing a robust and efficient maintenance strategy.

Ordinal classification (OC) is a unique type of multiclass classification in which the classes possess a natural underlying sequence. In traditional classification algorithms, the significant inherent order information is disregarded, whereas OC considers the relationships among class labels. It has been observed that considering this ordering information among classes leads to improved predictions when estimating the target value [4]. Naive, threshold, and ordinal binary decomposition approaches are the three main categories for the OC algorithm [5]. The naive approaches, in this context, refer to the usage of other machine learning paradigms such as regression, nominal, and cost-sensitive classification to obtain the model. The threshold approaches acquire a collection of thresholds by dividing the target class values into consecutive intervals, with each class label being assigned to an interval determined by these thresholds [4]. In ordinal binary decomposition (OBD) approaches, the main principle is based on the concept of "divide and conquer", as it involves dividing the ordinal label into several binary labels. Subsequently, the ultimate class labels are selected by consolidating the binary outputs into a single one. In this study, we introduce a novel algorithm that utilizes the OBD approach to enhance the performance of the classic OC algorithm.

PdM applications often overlook the structured information inherent in class labels, representing the machine's health status. Numerous studies have consistently demonstrated that the OC approach consistently outperforms nominal classification methods when dealing with datasets featuring ordered class targets [4–11]. This paper proposes a novel ordinal predictive maintenance with ensemble binary decomposition (OPMEB) algorithm that involves the decomposition of ordered multiclass problems into multiple binary subproblems. We aimed to enhance the predictive performance of the OC algorithm by introducing the new OBD method and to demonstrate the applicability of OC in the PdM domain, primarily because it is feasible to categorize the machine's health status according to the risk of failure. For instance, a machine that has been in operation for a short period poses no risk, whereas one with an extended operational history may present a critical risk in terms of potential failure.

The key contributions and novelty of this work can be listed as follows. (i) It has been demonstrated that the OC algorithm outperforms traditional classification algorithms in the field of PdM. (ii) A novel ordinal predictive maintenance with ensemble binary decomposition (OPMEB) algorithm is proposed by integrating PdM and OC paradigms, further enhancing the success of the OC algorithm. (iii) This study is also original in that it provides a comparative analysis of alternative OBD methods, such as OneVsAll (OVA), OneVsFollowers (OVF), and OneVsNext (OVN). (iv) The proposed OPMEB approach can be utilized with any ordinal data without necessitating prior knowledge of the specific PdM dataset, thus rendering it versatile and widely applicable. (v) The demonstrated superior performance of the OPMEB approach across diverse datasets, including C-MAPSS, AI4I 2020, and real-world hydraulic system datasets, underscores its robustness and generalizability across various PdM scenarios in terms of accuracy, recall, precision, and F-measure evaluation metrics.

In the experimental studies, the OPMEB algorithm was tested on three versions of the C-MAPSS and AI4I 2020 datasets, each discretized into three, four, and five ordinal class labels. Furthermore, the performance of the OPMEB algorithm was examined on a real-world hydraulic system dataset, which encompassed three distinct fault types, with both three and four ordinal class labels. Then, we conducted a comparative analysis by contrasting its performance with the standard OC algorithm [4]. Additionally, the results were compared with other OBD methods including OVA, OVF, and OVN. The results indicated that the OPMEB method effectively categorizes machine health states within the PdM domain, demonstrating its adaptability and suitability for diverse industrial machinery contexts.

This paper comprises five sections. Section 2 presents the related work in the literature on the subject. In Section 3, the novel proposed approach is explained thoroughly. Section 4 provides an overview of the datasets and showcases the experimental and comparative outcomes. Lastly, Section 5 discusses the final observations and possible future research paths.

2. Related work

Predictive maintenance (PdM) has become increasingly critical in recent years owing to its powerful strategy to present effective maintenance plans. A great number of studies have been introduced in different research areas, such as automotive [12], aerospace [13–16], energy [17–20], manufacturing [21–26], and transportation [6, 27, 28]. Understanding the present status, key issues, gaps, challenges, and future research directions in PdM is crucial. Review articles play a critical role by summarizing all available literature information. Numerous systematic literature reviews in various areas showcase the current state-of-the-art machine learning techniques applied in PdM [12, 13, 29, 30]. Jain et al. [12] addressed machine learning techniques for automotive PdM and vehicle health diagnosis, while Stanton et al. [13] highlighted difficulties and opportunities in aircraft PdM.

In the realm of artificial intelligence, machine learning and, more recently, deep learning, have surfaced as effective methodologies for constructing PdM models, attributed to their proficiency in executing failure prediction tasks. Machine learning paradigms, including regression, classification, and clustering, are employed in various studies to predict anomalies, failures, and unusual behaviors in machines successfully in different sectors as summarized in Table 1. For instance, the growing interest in using machine learning in manufacturing has led to the development of many different machine learning algorithms for various situations [21–26].

Another crucial aspect is the prediction of the remaining useful time (RUL) value, which holds significant importance as it shows the duration a machine is expected to operate before requiring replacement or revealing potential failures [15, 21, 22, 27]. In this regard, numerous deep learning methodologies have been suggested to address PdM challenges, such as forecasting the RUL of turbofan engines [16], railway equipment [27], and fault diagnosis of conveyor motors [25], and semiconductor lasers [23]. Moreover, CNNs have emerged as the leading deep learning architecture for predicting faults in various machinery across diverse domains, encompassing conveyor motors, turbines, and turbofan engines [16, 19, 25]. Additionally, they introduced various deep neural network (DNN) models, including LSTM networks [3] and autoencoders, for PdM tasks [21, 27]. When clustering is preferred in machine learning, the K-means algorithm successfully recognizes faults [17, 26]. Research has shown that artificial intelligence-driven methods, encompassing both machine learning and deep learning, display enhanced efficacy and precision in PdM tasks such as estimating RUL, and diagnosing faults [31].

Table 1. Summary of different machine learning techniques in PdM systems.

Ref.	Year	Algorithm	Application	Approach	Industry
[16]	2023	Convolutional neural network (CNN), Monte Carlo dropout	RUL prediction of turbofan engine	Regression	Aviation
[17]	2023	K-Means	Detection of unusual behavior in wind turbines	Clustering	Energy
[23]	2022	Gated recurrent unit (GRU), autoencoder	Health prognosis of semiconductor laser	Classification, Regression	Manufacturing
[24]	2022	Decision tree (DT)	Failure prediction of the gearbox for roasting oilseeds	Classification	Manufacturing
[27]	2022	Long short-term memory (LSTM), autoencoder	RUL prediction of railway equipment	Regression	Transportation
[19]	2021	Recurrent neural network (RNN), Convolutional neural network (CNN)	Prediction of the real-time power of a turbine	Regression	Energy
[20]	2020	DT, K-nearest neighbor (KNN)	Prediction of wind turbine blade delamination	Classification	Energy
[25]	2020	CNN, SVM	Fault diagnosis of conveyor motors	Classification	Industry
[26]	2020	K-Means	Fault recognition model for rotating machinery	Clustering	Manufacturing
[31]	2020	Multilayer perceptron (MLP), support vector machine (SVM)	Fault prediction of a centrifugal pump in the oil and gas industry	Classification	Industry

Recently, ordinal classification has demonstrated successful applications across diverse research domains such as transportation [6], human activity recognition [7], and image processing [8]. In [9], an ordinal classification algorithm based on an ensemble approach is presented. This proposed method makes a final estimation through a weighted voting system by minimizing the cost of classification. In [10], the authors investigated semisupervised learning for ordinal classification and presented extensive experimental study results to show the success of the proposed algorithm. Ensemble techniques and ordinal classification are already explored areas in the literature; however, the combination of these fields, particularly in conjunction with the binary decomposition approach in predictive maintenance, has yet to be deeply investigated, representing an intriguing avenue for further research. Our method aims to combine the strengths of these fields to address the challenges inherent in the classification problems in predictive maintenance. The proposed approach leverages the ordinal classification task with the ensemble learning principles to address classification tasks, considering relatively uncharted territory in the literature.

The PdM studies aforementioned focus on different research areas by applying various machine learning techniques. Although there are many studies in the field of PdM in the literature, the number of papers applying the ordinal classification algorithm in the PdM area is almost nonexistent. To the best of our knowledge, ordinal classification has never been considered comprehensively; only in [11], the author applied the ordinal classification method in the field of PdM but in a different way. In our study, we introduce a novel ordinal predictive maintenance with ensemble binary decomposition (OPMEB) method which utilizes the ordinal classification algorithm by presenting a new ordinal binary decomposition technique that considers the hierarchical nature of class labels reflecting the machine's health status for the PdM domain.

3. Materials and methods

This section contains background information concerning the techniques applied in this study and describes the proposed approach, "ordinal predictive maintenance with ensemble binary decomposition" called OPMEB in detail.

3.1. Ensemble learning

Ensemble learning combines predictions from multiple classifiers to improve accuracy and robustness in machine learning, particularly in classification tasks [6]. The fundamental idea behind ensemble algorithms is to leverage the collective intelligence of a diverse set of base classifiers. Instead of relying solely on the predictions of a single classifier, ensemble methods combine these predictions in a strategic manner to form a unified and typically more accurate classification. This collective decision-making process tends to outperform the individual classifications provided by each base classifier in isolation [9]. In essence, ensemble learning enhances predictive performance by merging the strengths of individual classifiers.

3.2. Ordinal classification

Ordinal classification (OC) is a supervised learning problem that represents a unique form of multiclass classification characterized by an inherent order among the classes. For instance, class labels of a target value for a machine's components can have ranking values such as healthy, low risk, moderate risk, high risk, and critical failure, arranged from the most favorable condition to the most severe.

The OC algorithm [4] involves predicting the label, denoted as y, for a given input vector x, where X is a d-dimensional input space, $\mathrm{X} \epsilon \mathbb{R}^d$. The label y belongs to a label space Y, which consists of n distinct labels, represented as $y \in Y = \{C_1, C_2, \ldots, C_{n-1}, C_n\}$ where $C_1 < C_2 < \ldots < C_{n-1} < C_n$. The "$<$" symbol indicates the ordering relationship between the labels. The main goal is to discover a classification function, denoted as $f : X \rightarrow Y$, which accurately forecasts the label y for a given x new input.

3.3. The ordinal binary decomposition approach

One of the major approaches for OC is the binary decomposition method. The fundamental concept of this approach involves breaking down the ordinal problem into multiple binary classification subproblems, treating each problem independently by constructing multiple models. Subsequently, the binary outputs are combined to determine the final label during the classification phase, enabling the prediction of the ordinal class.

One of the significant approaches for ordinal binary decomposition (OBD), known as OrderedPartitions, involves assigning a label of 1 to classes with higher ranking order while labeling the remaining ones as -1 to indicate their negative status. In the OC algorithm they introduced [4], Frank and Hall utilized the decision tree C4.5 as the base learner, and subsequently, the final decision of various binary classifiers was ascertained through the computation of the respective probabilities assigned to each class. Then, the class with the highest probability among all the classifiers is selected.

Different binary decomposition techniques have been proposed to address the question of how to effectively decompose the ordinal target variable into a series of binary variables [5]. In the OneVsAll (OVA) technique, each binary dataset consists only of instances belonging to a single class. This means that classifiers are trained using instances exclusively from one class in each binary dataset. Only the instances belonging to the current class is assigned to 1, while instances from all other classes are labeled as -1, ensuring that each binary classifier is focused on discriminating one class from the rest of the classes in the multiclass problem [32].

In the OneVsFollowers (OVF) technique, the first class is labeled as -1, and all the following classes with higher ranking order are labeled as 1. The classes with lower ranking order are labeled as 0 and not included in the dataset. This process is repeated for each subsequent class until all the ordered labels have been assigned, ensuring that each class is labeled based on its relative position in the ordinal sequence [33].

When employing the OneVsNext (OVN) technique, the dataset is constructed by including only the next class with a higher ranking order. The class being considered is labeled as 1, indicating its positive status. The current class is set to -1, denoting its negative status. All the remaining classes are assigned a label of 0 and are not included in the learning process. This approach ensures that each binary dataset focuses on the classification between a specific class and the next higher-ranked class while disregarding the other classes [33].

3.4. The proposed approach: ordinal predictive maintenance with ensemble binary decomposition (OPMEB)

In predictive maintenance (PdM) studies, classification, and regression techniques are applied to carry out target class and remaining useful life (RUL) predictions. During these investigations, the inherent order among classes is often overlooked despite its relevance. When considering the health status of machines, the risk of failure follows a discernible pattern. Machines that are in their early operational stages exhibit a lower risk of malfunction, while those that have been in operation for a while may transition to a moderate-risk category. However, as machines continue to operate over an extended period, the risk profile tends to ascend, potentially reaching higher or even critical risk levels, ultimately leading to potential breakdowns. It also indicates the inherent presence of a natural hierarchy among the health condition categories within PdM data. Although traditional classification methods ignore this order, its positive impact on prediction power and improved accuracy results were already demonstrated in [4]. Motivated by this insight, we proposed a novel algorithm named ordinal predictive maintenance with ensemble binary decomposition (OPMEB) in the PdM domain, utilizing the OC algorithm to leverage this order and enhance predictive outcomes.

Figure 1 demonstrates the comprehensive process of the OPMEB approach. This sequence involves distinct phases, including constructing different ordinal binary decomposition (OBD) approaches, generating binary datasets for each different OBD approach, training models, executing directional decision-making strategies in classification, evaluating prediction performance, and determining the best prediction performer. The initial step involves dynamically constructing multiple ordinal binary decomposition approaches based on the target class number. By forming upward and downward unions of classes, considering the inherent order among existing classes, all possible combinations are applied to create a variety of OBD approaches. For an n-class dataset, $(n-1)^2$ different OBD approaches are generated, denoted as OBD_1, OBD_2, OBD_3, and extending up to $OBD_{(n-1)^2}$. Each OBD_x decomposes the original multiclass problem into a unique set of simpler multiclass subproblems. In the subsequent step, corresponding subbinary datasets are created for each OBD approach. For an n-class dataset, $n-1$ subdatasets are generated by assigning labels of 0, -1, and 1 to indicate the instances from lower- and higher-ranking classes within each subdataset, represented as SD_1, SD_2, ..., $SD_{(n-1)}$.

During the training step, the C4.5 classification algorithm is applied to each generated subdataset while retaining the natural order between the class values. Training occurs for each subclass, leading to $n-1$ applications of the C4.5 binary classifier for an n-class dataset. This yields distinct models, labeled as M_1, M_2, ..., $M_{(n-1)}$, each trained with a subset associated with different classes, enabling diverse training scenarios.

After obtaining prediction results from all models, the OPMEB approach assigns a class to the input data using binary classifiers constructed in the previous phase. This involves applying a decision-making strategy in either the forward or backward direction. This strategy involves verifying each model's predictions using three distinct decision techniques: forward, forward iteration, and backward. The forward method aligns binary classifiers, prioritizing the identification of the lowest-level classes first and the highest-level last. Conversely, the backward method orchestrates binary classifiers in the opposite direction. During forward iteration, classifiers prioritize the prediction of lower-level classes before progressing to higher levels. This process involves the prediction result of the next higher-level class and validating it until a negative class is encountered.

Following that, 10-fold cross-validation is conducted and the most successful classification strategy is identified for each subdataset. Afterward, an overall assessment determines the optimal directional decision-making strategy exhibiting the highest performance across all subdatasets. The path with the most successful predictive power is preferred among all obtained results. Finally, using the ensemble learning approach through a collective assessment of these results based on their prediction performance, the combination of the OBD technique and the directional decision-making strategy achieving the highest prediction performance is determined for the given PdM dataset when applying the C4.5 classification algorithm. For instance, the OPMEB method determines that the OBD_2 + backward approach or OBD_5 + forward approach yields the best performance. Consequently, this specific combination is selected for future predictions to assist maintenance strategists and decision-makers.

3.5. The formal definition of the proposed OPMEB approach

In an ordinal dataset D with k instances, denoted as $D=\{(x_i, y_i) \mid i=1,2,\ldots,k\}$, each data point (x_i, y_i) consists of an input x_i and a corresponding machine's health status class label y_i. The input vector x_i, belonging to the d-dimensional feature space $\mathrm{X} \subseteq \mathbb{R}^d$, is paired with a class label y_i associated with the health status set $Y=\{c_1, c_2, \ldots, c_n\}$, representing statuses like healthy, low risk, moderate risk, and high risk, where n represents the number of classes. The status classes are ordered consistently as $c_1 < c_2 < \ldots < c_n$, denoting their order. The primary objective in this context is to determine a decision function $f : X \rightarrow Y$ which accurately predicts the health status class for any machine's data with the best possible fit.

Definition 1 *The OPMEB method aims to develop an improved classification approach in the PdM domain, taking into consideration the inherent order of class labels representing the health status of machines to accurately predict the future state or performance of the system. This is achieved through a novel ordinal binary decomposition (OBD) approach.*

The proposed OPMEB method comprises four main steps. In the first step, various OBD approaches are dynamically constructed based on the target class number in the given dataset. In the second step, for each derived OBD approach, the ordinal PdM problem, involving n health statuses of machines, is transformed into $n-1$ binary classification problems. These states represent categories such as *healthy* < *low risk* < *moderate risk* < *high risk*. In the third step, a base learner is employed to construct $n-1$ models for each binary dataset individually, enabling predictions to be made. In the final step, during the interpretation, a directional decision-making strategy is employed to make a prediction. Finally, all prediction results are evaluated and the model with the best predictive performance is chosen.

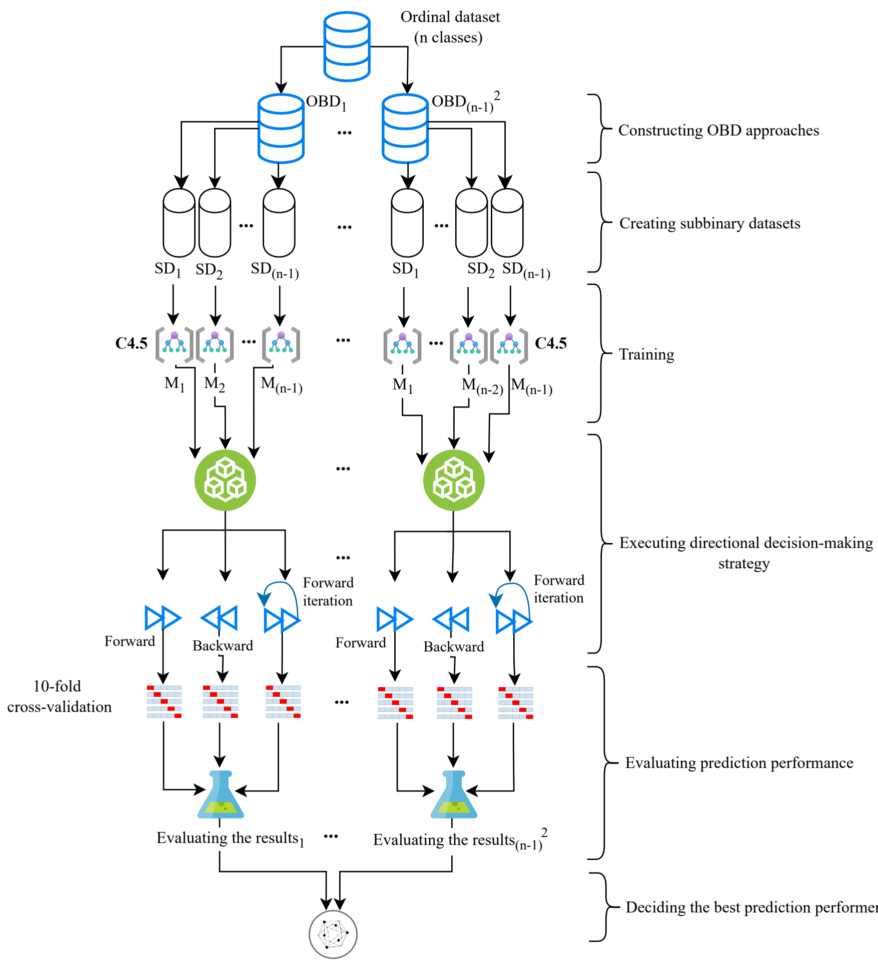

Figure 1. A comprehensive summary of the proposed OPMEB approach.

Definition 2 *The OPMEB method dynamically generates multiple OBD approaches, involving the transformation of a multiclass problem into a set of binary subproblems by forming essential upward and downward unions of classes in distinct manners based on the given dataset and considering the number of classes.*

Let OBD_x denote the x^{th} ordinal binary decomposition approach. For an n-class ordinal dataset OBD_1, OBD_2, ..., $OBD_{(n-1)^2}$ approaches are applied to the original dataset D, where x spans from 1 to $(n-1)^2$. Each OBD_x is formulated by assigning labels to lower and higher classes in different ways, indicating their ordinal relationships. In all these formulations, the preceding labels can be extended from 0 to $n-1$, while the subsequent labels can be extended from 1 to $n-2$. It means at least one subsequent higher-class labeling is performed. For example, in a 4-class dataset D, the preceding class number to be labeled is 2, and the subsequent class number is 1. Then, $Y' = \{c_{i-2}, c_{i-1}, c_i, c_{i+1}\}$. The label $y_j \in Y'$ linked with the instance x_j is substituted with $y_j = -1$, $\forall y_j \leq c_i$, and, $y_j = 1$, $\forall y_j > c_i$, and $y_j = 0$ for the others. In other words, when considering class c_i, class values higher than c_i are labeled as 1, class values lower than or equal to c_i are labeled as -1, and the rest are labeled as 0. Labels 1, -1, and 0 represent positive, negative, and unselected class statuses, respectively. Unselected signifies that they are not included in the binary subdataset. By applying this labeling process for each OBD_x, the OPMEB method transforms the ordinal classification problem with n classes into $n-1$ binary classification problems, encoding the ordinal sequence of the class labels. In this way, a collection of all potential OBD approaches is generated, each capturing different aspects of the ordinal relationships within the dataset.

Figure 2 shows different OBD approaches formulated for a 4-class ordinal dataset scenario as an example. The matrices illustrate the distribution and arrangement of classes within each method's decomposition formulations. The values R_1, R_2, R_3, and R_4 in Figure 2 serve as examples representing the risk classes associated with a machine's health status, denoting *healthy, low risk, moderate risk,* and *high risk*, respectively. This binary dataset contains a target value, determined by checking if the class value in the original dataset is equal to, below, or above the rank of the associated class, with T used to indicate the target class. In these matrices, the columns present the binary subproblems, while the rows indicate the role of each class within each subproblem. Each element M_{ij} in the decomposition table M takes values from the set {-1, 1, 0}, where 1 or -1 represents the assigned positive or negative class, respectively, and 0 indicates an unselected class that is not considered in the learning process. Each decomposition matrix M displays a range denoted as $[L:x][H:y]$. Here, L and H represent the lower and higher classes, respectively. x indicates how many preceding classes will be labeled as lower, and y denotes the number of subsequent higher classes to be labeled. All these matrices present how classes are organized in the different OBD formulations of each approach. The objective is to experiment with all possible combinations of labeling lower and upper classes in a distinct manner, resulting in the generation of diverse OBDs.

[L:2][H3]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	-1	-1
R_2	1	-1	-1
R_3	1	1	-1
R_4	1	1	1

[L:2][H2]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	-1	-1
R_2	1	-1	-1
R_3	1	1	-1
R_4	0	1	1

[L:2][H1]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	-1	-1
R_2	1	-1	-1
R_3	0	1	-1
R_4	0	0	1

[L:1][H3]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	-1	0
R_2	1	-1	-1
R_3	1	1	-1
R_4	1	1	1

[L:1][H2]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	-1	0
R_2	1	-1	-1
R_3	1	1	-1
R_4	0	1	1

[L:1][H1]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	-1	0
R_2	1	-1	-1
R_3	0	1	-1
R_4	0	0	1

[L:0][H3]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	0	0
R_2	1	-1	0
R_3	1	1	-1
R_4	1	1	1

[L:0][H2]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	0	0
R_2	1	-1	0
R_3	1	1	-1
R_4	0	1	1

[L:0][H1]	$T > R_1$	$T > R_2$	$T > R_3$
R_1	-1	0	0
R_2	1	-1	0
R_3	0	1	-1
R_4	0	0	1

Figure 2. Generated binary datasets with the proposed decomposition method for 4-class labels.

The labeling process for lower and higher classes varies, leading to different OBD approaches. Ordered-Partitions, OVA, OVF, OVN, and similar OBD approaches label lower and higher classes in unique ways, thereby generating different binary subdatasets. Models are then trained differently for each specific binary dataset. The crucial aspect in this context lies in understanding how the underlying binary datasets are formed for each OBD approach. In this study, the C4.5 base learner is applied to the generated binary datasets to build $n-1$ models in the training stage. Let M_i, for $i=1,2,\dots,n-1$, denote the model generated for the ordinal classification problem. A separate model M_i is trained on their corresponding subdatasets: SD_1, SD_2, ..., $SD_{(n-1)}$, respectively.

Definition 3 *The OPMEB method establishes a rule for forecasting unseen inputs once the prediction results are obtained. Following the learning process, predictions from each model are examined to conclude a label as a result. This is achieved through the implementation of a directional decision-making strategy, which includes three decision techniques: forward, forward iteration, and backward.*

The forward method starts with the initial model, M_1. If an unseen instance is classified as 0 by M_1, it proceeds to the next model, and this process continues until the last model, $M_{(n-1)}$, is assessed. The method concludes when a model predicts 1 for the test instance, and the outcome of that specific model is selected. This method aligns binary classifiers, with a priority on identifying the lowest-level classes first and progressing towards the higher levels. Let $\text{Class}(x)$ be a function as defined in Equation (1):

$$\text{Class}(x) = C_{i+1} \text{ where } i = \begin{cases} \exists M_j(x) \in \{1\}, & \min\{j \mid M_j(x) = 1\} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

This notation states that the assigned class for the test instance x is C_{i+1}, where i is the index of the first model that predicts a positive label for the given test instance x, and i can take values from 0 to $n-1$. The notation $\min\{j \mid M_j(x)=1\}$ denotes the minimum index j satisfying the condition $M_j(x)=1$. If there is no such j, then i is set to 0 concerning the lowest-level class.

The forward iteration method fundamentally follows the same logic as the forward method, but the key difference is that in forward iteration, the prediction made by the subsequent model is also verified. If the following model predicts a positive label for the given test instance x, then the prediction made by that subsequent model proceeds to be verified as well. This iterative validation process persists until the subsequent model predicts a negative label. The final selection is then made based on the prediction of the last model that forecasted a positive label.

Drawing upon the same logic, the backward method initiates with the $M_{(n-1)^{\text{th}}}$ model. All the subsequent steps follow the same procedure as the forward method but in reverse order. This method is designed to align binary classifiers, prioritizing the identification of higher-level classes first and progressing toward the lower levels.

For instance, in a labeled $[L:1][H:2]$ dataset with four target classes, M_1 compares C_1 with C_2, C_3; M_2 compares C_1, C_2 with C_3, C_4; and M_3 compares C_2, C_3 with C_4. Predictions from three different models are evaluated using directional decision-making strategies. Our example scenario is as follows: M_1 predicts 0; M_2 predicts 1; M_3 predicts 1 for a classification task. Applying the forward method, the result of the first positive prediction is considered correct, and for this case, it is assigned the C_3 class. According to the backward and forward iteration method, the C_4 class is assigned.

After applying the aforementioned methods, the obtained results are evaluated, and the one with the highest success among all these results is selected using the ensemble learning approach. The decision on which binary decomposition method and directional decision-making strategy to be used together comes from this evaluation.

3.6. The algorithmic structure of the proposed OPMEB approach

Algorithm 1 illustrates the pseudocode for the introduced OPMEB approach, structured into three distinct steps. In the initial step, the algorithm iterates through potential lower and higher class pairs (L, H), and binary datasets D_i are constructed for each pair. Instances in the original dataset are assigned new labels {-1, 1, 0} based on their ordinal relationship with the current class. During the second step, an individual model M_i is constructed to train an ordinal classifier (C_i) for the current class using the training dataset D_i associated with class i. Then, the set of ordinal classifiers (C^*) is updated by adding the newly trained classifier C_i. Upon completion of the loop, this step builds a collection of ordinal classifiers (C^*) by training individual classifiers for each class in the ordinal dataset. This approach ensures that the ordinal relationships among classes are taken into account during the training process. In the last step, the algorithm iterates through each directional predictor p in the set P and predicts class labels for each instance x in T. The results are stored in a predicted test set T', and it dynamically updates the best performer based on model performance. The process ensures the selection of the most effective combination of ordinal classifiers and directional predictors for accurate ordinal class label predictions.

The computational time complexity of the initial part is $O((k-1)^2 \times n)$, where n represents the number of instances and k denotes the number of classes in the dataset. The time complexity for the second step is $O((k-1) \times T(n))$, where $T(n)$ indicates the time needed for the execution of a base learner on n instances. For the last step, the total computational complexity is $O(q \times m \times (k-1) \times |P|)$, where q is the number of directional predictors, m is the size of the test set, and $|P|$ is the size of the set of directional predictors. So, the total time complexity is $O((k-1)^2 \times n + (k-1) \times T(n) + q \times m \times (k-1) \times |P|)$ since the method builds $(k-1)^2$ models.

4. Experimental studies

In the experimental studies, the C-MAPSS, AI4I 2020, and a real-world hydraulic system's condition datasets were used to show the impact of the proposed approach on prediction success in the field of predictive maintenance (PdM). It is important to note that the proposed algorithm was tested on three different configurations of PdM datasets, encompassing datasets from three distinct domains. This study aimed to demonstrate how the OPMEB technique enhances the accuracy of predictions within the PdM domain.

The OPMEB approach was developed using the C# programming language, employing the WEKA machine learning library [34]. In the experiments, the C4.5 classification algorithm was used as a base learning algorithm for the ordinal classification (OC) algorithm with its default parameters. The performance of the proposed algorithm was measured using accuracy, precision, recall, and F-measure metrics. Accuracy provides an overall measure of the model's correctness, while recall and precision offer deeper insights into the model's ability to identify relevant instances and its exactness in doing so, respectively. Accuracy is quantified by determining the ratio of accurately predicted observations to the overall count of observations in the dataset as given in Equation (2):

$$\text{Accuracy} = \frac{TN + TP}{FP + TN + FN + TP} \tag{2}$$

where false positives (FP) represent the count of incorrectly classified data examples, true positives (TP) denote the count of accurately classified positive data examples, false negatives (FN) indicate the count of misclassified positive data examples, and true negatives (TN) signify the count of correctly predicted negative data examples. Precision represents the proportion of positive observations that are accurately classified compared to all the positive outcomes. (Equation (3))

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Algorithm 1 Ordinal predictive maintenance with ensemble binary decomposition (OPMEB)

Inputs:
D: the ordinal dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ with n instances
X: input feature set, an input vector $x_i \in X$
Y: ordinal class labels, a class label $y_i \in Y = \{c_1, c_2, \ldots, c_k\}$ with a relationship $c_1 \prec c_2 \prec \ldots \prec c_k$
k: the number of classes
C^*: ordinal classifiers
P: directional predictors, a predictor $p \in P = \{\text{forward}, \text{forward iteration}, \text{backward}\}$
L: lower class
H: higher class
T: test set that will be predicted
T': predicted test set
Output:
M: ordinal classification model $M = \{C^*, P\}$
Begin Algorithm:
for $L \leftarrow 0$ to $k-2$ **do**
  for $H \leftarrow 1$ to $k-1$ **do**
    // Step 1 - Generation of binary datasets from the ordinal dataset, D
    for $i \leftarrow 1$ to $k-1$ **do**
      for all (x_j, y_j) in D **do**
        if $(c_{i-L} \preceq y_j \preceq c_i)$ **then**
          D_i.Add$(x_j, -1)$ // Class values less than or equal to c_i are labeled as -1
        else if $(c_i \prec y_j \preceq c_{i+H})$ **then**
          D_i.Add$(x_j, 1)$ // The class values greater than c_i are labeled as 1
        else
          D_i.Add$(x_j, 0)$ // Do not add this instance, skip it
        end if
      end for
    end for
    // Step 2 - Generation of unified binary classifiers
    for $i \leftarrow 1$ to $k-1$ **do**
      C_i = Train(D_i) // Constructing a classifier on the training set employing a learning algorithm
      $C^* = C^* \cup C_i$
    end for
    // Step 3 - Evaluation of directional decision-making strategy
    for all p in P **do**
      for all x in T **do**
        $y = p(C^*, x)$
        T'_i.Add(x, y)
      end for
      if M is empty **then**
        $M = \{C^*, p\}$
      else
        $M = \text{BestPerformer}(M, \{C^*, p\})$
      end if
    end for
  end for
end for
End Algorithm

Recall shows the proportion of right predictions for a specific class relative to all the correct predictions attributed to that class (Equation (4)).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

Lastly, the F-measure serves as a valuable performance indicator of prediction quality, computed as the harmonic mean of precision and recall as defined in Equation (5). This measure yields values within the range of 0 to 1, where 1 indicates the best performance.

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

These metrics are critical in the PdM field as they directly impact the reliability and efficiency of maintenance scheduling. High accuracy ensures the model's general reliability, while high recall ensures that most potential failures are detected, minimizing unexpected downtimes. Precision, on the other hand, ensures that maintenance actions are necessary and not overly frequent, which optimizes resource use. Therefore, our model's performance, as indicated by these metrics, demonstrates its effectiveness in predicting maintenance needs accurately and efficiently, thereby contributing significantly to reducing operational costs and improving equipment uptime in industrial settings.

In this work, the n-fold cross-validation technique with n chosen as 10 was used to compute the classification accuracies. This validation technique, which includes randomly dividing the data into ten separate and equal partitions, is iterated n times with changing parts for the training and testing phases. Each repetition involves reserving one partition for testing purposes, while the remaining partitions are used to train the model. The validity of the model is evaluated based on the average error at the conclusion.

We planned three experiments to explore the effects and outcomes of the following key aspects. In experiment 1, in order to assess the superiority of OC over the standard classification approach, we conducted an evaluation of the nominal and ordinal classification algorithms described in [4] using PdM datasets, aiming to demonstrate their relative performance. In experiment 2, we performed a comparative analysis between the OPMEB method and the conventional OC algorithm as described in [4] to demonstrate the efficacy of our approach on PdM datasets. In experiment 3, we evaluated the prediction performance of the OPMEB method against other ordinal binary decomposition techniques, including OneVsAll (OVA), OneVsFollowers (OVF), and OneVsNext (OVN), in order to establish its superiority on PdM datasets. Finally, the results obtained from the experiments have all been thoroughly analyzed and illustrated via charts and tables.

4.1. Dataset description

To validate the proposed approach's efficacy in predictive maintenance (PdM), we utilized the C-MAPSS, AI4I 2020, and hydraulic system datasets. The C-MAPSS dataset [35], developed by NASA, features simulated data on aircraft turbofan engine degradation generated through a model-based simulation program. C-MAPSS comprises FD001, FD002, FD003, and FD004 subdatasets, representing distinct operating and fault conditions. For this study, we focused on subset FD004, comprising 61,249 instances and 26 attributes. The dataset includes engine numbers, operational sensor settings, and multivariate temporal data collected from 21 sensors per flight cycle, along with run-to-failure (R2F) data for these sensor measurements. Over time, the engine units begin to degrade until a failure occurs, so the main objective is to predict the RUL as the target attribute. The AI4I 2020 PdM dataset [36] on milling processes shows real maintenance data that industries often deal with. The dataset contains information about failures of milling machines, comprising 10,000 instances, with each row having 14 features stored in columns. The milling machine failure comprises five distinct independent modes: tool wear, heat dissipation, power, overstrain, and random failures. A real-world hydraulic system's condition dataset [37] is constructed based on the measured process values obtained from multiple sensors on a hydraulic test rig, including temperature, motor power, vibration, cooling efficiency, volume flows, efficiency factors, and pressure, as well as four fault types of hydraulic components such as cooler performance, valve status, internal pump leakage, and the state of the hydraulic accumulator. It consists of 2205 instances with 17 inputs and four target attributes. In this study, we examined valve condition, internal pump leakage, and hydraulic accumulator fault types, each with ordinal target class values of 4, 3, and 4, respectively.

The hydraulic system's condition dataset is already ordinal, while the C-MAPSS and AI4I 2020 PdM datasets are not specifically intended for ordinal classification. To implement the OC algorithm, the target attribute values, originally numerical, were transformed into ordinal class labels using equal bin discretization which involves dividing the target variable into different bins with an equal number of instances in each bin. The transformation was performed in response to the algorithm's requirement for ordinal class representations. Varied bin configurations were applied to the same dataset, resulting in the generation of distinct datasets from the original one. The target value for the C-MAPSS and AI4I 2020 PdM datasets were discretized into three, four, and five ordinal class labels, respectively, leading to the creation of three different versions. Categorical labels were assigned to establish an ordering relation among them. For example, in the case of a 4-class dataset with labels R_1, R_2, R_3, and R_4, each label corresponds to different risk factors of machines. The ordering of the labels, such as $R_4 > R_3 > R_2 > R_1$, reflects the magnitude of risk, representing *high risk, moderate risk, low risk,* and *healthy*, respectively, based on the RUL value associated with each instance in the dataset.

4.2. Experimental results

4.2.1. The results of experiment 1

In the first experiment, the aim is to observe the prediction performance of nominal and ordinal classification algorithms in the PdM domain. The expectation here is to observe that in the ordinal-transformed PdM datasets, as has been previously demonstrated in different domains [4–11], the ordinal classification algorithm achieves predictions with a high level of accuracy. For this experiment, we worked with three different versions of the C-MAPSS, AI4I 2020, and hydraulic system's condition datasets, each having three, four, and five target classes. Since the ordinal classification algorithm employs the C4.5 decision tree algorithm as its base learner, the same algorithm was selected for nominal classification as well. Then, we applied both nominal and ordinal classification algorithms [4] to each dataset. The results of both the C-MAPSS and AI4I 2020 datasets, each with ordinal target class versions of 3, 4, and 5, and for the hydraulic system dataset, specifically for valve condition, internal pump leakage, and hydraulic accumulator fault types, each with ordinal target class values of 4, 3, and 4, respectively, were analyzed and compared based on accuracy, recall, precision, and F-measure evaluation metrics. For each metric, the most successful results are highlighted in bold. In Table 2, the abbreviations H-valve, H-pump, and H-acc represent the fault types for valve condition, internal pump leakage, and hydraulic accumulator in the hydraulic system's condition datasets, respectively. Upon thorough analysis of the obtained results, it is evident that the ordinal classification algorithm achieved superior performance compared to the traditional classification algorithm across all performance metrics in the different PdM datasets. It experimentally confirmed that considering the order of class labels in the PdM domain can result in the construction of superior models compared to the nominal classification.

Table 2. Comparison of the results of the nominal and ordinal classification algorithms [4] on the PdM datasets in terms of accuracy (%), precision, and recall.

Dataset	Accuracy (%)		Precision		Recall	
	Nominal	Ordinal	Nominal	Ordinal	Nominal	Ordinal
C-MAPSS (3-class)	86.84	**88.19**	0.8690	**0.8830**	0.8680	**0.8820**
C-MAPSS (4-class)	82.21	**83.30**	0.8230	**0.8350**	0.8220	**0.8330**
C-MAPSS (5-class)	78.02	**79.82**	0.7810	**0.8010**	0.7800	**0.7980**
AI4I 2020 (3-class)	69.97	**73.84**	0.7000	**0.7490**	0.6990	**0.7380**
AI4I 2020 (4-class)	64.70	**65.15**	0.6470	**0.6710**	0.6470	**0.6510**
AI4I 2020 (5-class)	59.93	**61.40**	0.5990	**0.6310**	0.5990	**0.6140**
H-valve (4-class)	80.58	**85.19**	0.8050	**0.8660**	0.8060	**0.8520**
H-pump (3-class)	85.51	**92.39**	0.8540	**0.9260**	0.8550	**0.9240**
H-acc (4-class)	87.42	**88.92**	0.8740	**0.8930**	0.8740	**0.8890**
Average	77.24	**79.80**	0.7724	**0.8061**	0.7723	**0.7979**

Figure 3 displays the F-measure outcomes across all PdM datasets, including the average results derived from these datasets. It highlights the superior performance of the OC algorithm compared to traditional classification algorithms in the PdM domain, as evidenced by the improvement in F-measure when applied to PdM datasets.

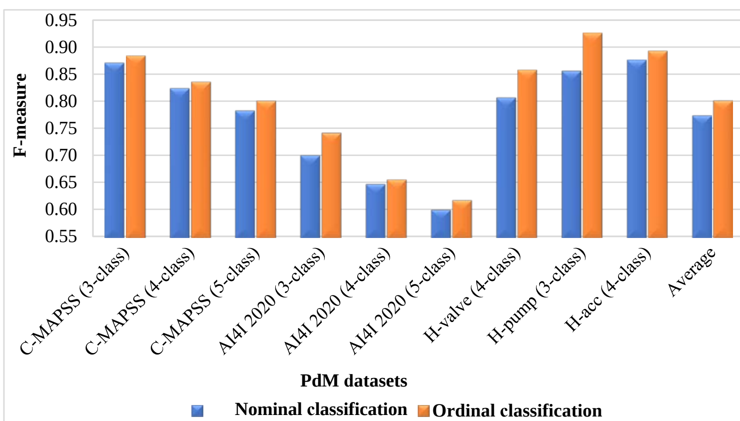

Figure 3. The performance improvement provided by applying the OC algorithm [4] to the PdM datasets in terms of F-measure.

4.2.2. The results of experiment 2

The main and most important goal is to show the superiority of the proposed novel algorithm, denoted as OPMEB, in the PdM domain over OC results. For this experiment, we worked with three different versions of the C-MAPSS, AI4I 2020, and hydraulic system's condition datasets. The OC and OPMEB algorithms were applied to PdM datasets to predict the health status of different machines. Table 3 provides a comparative analysis and indicates that our proposed algorithm, OPMEB, consistently outperforms the OC algorithm across all evaluation metrics for all PdM datasets, as evidenced by higher accuracy, precision, and recall metrics. The consistent improvement in accuracy across all datasets is clearly demonstrated. For instance, in the case of the 4-class C-MAPSS dataset, OC achieved an accuracy of 83.30%, while the OPMEB method demonstrated an accuracy of 86.02%. In the AI4I 2020 (3-class) dataset, OPMEB significantly outperforms OC with an accuracy of 80.09% compared to 73.84%. In the H-acc (4-class) dataset, OPMEB achieves a remarkable accuracy of 98.91%, far exceeding OC's 88.92%. Across all datasets, the average accuracy improves from 79.80% (OC) to 84.78% (OPMEB). The superior precision achieved by OPMEB, averaging 0.8508 compared to OC's 0.8061, indicates that OPMEB is more effective in correctly identifying relevant instances without being misled by irrelevant ones. This is particularly evident in complex datasets like AI4I 2020 (4-class), where OPMEB's precision is significantly higher. OPMEB's higher recall, averaging 0.8478 versus OC's 0.7979, demonstrates its capability to capture a higher proportion of true positives. This is crucial in industrial applications where missing a critical event could lead to significant consequences.

Table 3. Comparison of the results of the OC [4] and OPMEB (proposed) algorithms on the PdM datasets in terms of accuracy (%), precision, and recall.

Dataset	Accuracy (%)		Precision		Recall	
	OC	OPMEB	OC	OPMEB	OC	OPMEB
C-MAPSS (3-class)	88.19	**89.26**	0.8830	**0.8933**	0.8820	**0.8926**
C-MAPSS (4-class)	83.30	**86.02**	0.8350	**0.8614**	0.8330	**0.8602**
C-MAPSS (5-class)	79.82	**81.46**	0.8010	**0.8159**	0.7980	**0.8146**
AI4I 2020 (3-class)	73.84	**80.09**	0.7490	**0.8034**	0.7380	**0.8009**
AI4I 2020 (4-class)	65.15	**73.94**	0.6710	**0.7519**	0.6510	**0.7394**
AI4I 2020 (5-class)	61.40	**69.72**	0.6310	**0.7044**	0.6140	**0.6972**
H-valve (4-class)	85.19	**88.38**	0.8660	**0.8852**	0.8520	**0.8838**
H-pump (3-class)	92.39	**95.21**	0.9260	**0.9522**	0.9240	**0.9521**
H-acc (4-class)	88.92	**98.91**	0.8930	**0.9891**	0.8890	**0.9891**
Average	79.80	**84.78**	0.8061	**0.8508**	0.7979	**0.8478**

Furthermore, Figure 4 presents the F-measure values for all PdM datasets. This figure highlights the performance comparison between the OC algorithm and the OPMEB method. It emphasizes the improved efficiency and usefulness of the OPMEB technique, confirming its relevance and promising potential in the domain of PdM when compared to the conventional ordinal classification approach. Therefore, it can be inferred that the proposed approach has the potential to attain high F-measure values for ordinal PdM data from different domains.

Lastly, it can be inferred that the OPMEB algorithm, as proposed, holds significant promise in attaining enhanced accuracy, precision, recall, and F-measure values when applied to ordinal versions of PdM datasets. Thus, it is evident that the OPMEB algorithm can successfully forecast the conditions of the machines by taking into account the inherent order of class labels.

4.2.3. The results of experiment 3

The key objective of this experiment is to showcase the predictive efficacy of the OPMEB algorithm within the PdM domain, contrasting it with other OBD algorithms such as OVA, OVF, and OVN, across various PdM datasets. Three distinct OBD methodologies, alongside the OPMEB algorithm, were applied to datasets encompassing the C-MAPSS, AI4I 2020, and hydraulic system conditions. As seen in Table 4, the outcomes revealed that the OPMEB algorithm consistently achieved accuracy equal to or higher than the OVA, OVF, and OVN approaches across all PdM datasets. For example, OPMEB achieved the highest accuracy across all class configurations of C-MAPSS: 89.26% for 3-class, 86.02% for 4-class, and 81.46% for 5-class. Similarly, the OPMEB method exhibited the best prediction performance for AI4I 2020. In the H-valve and H-pump datasets, OPMEB achieved the same accuracy as OVF but surpassed OVA and OVN with accuracies of 88.38% and 95.21%, respectively. In summary, across all datasets, OPMEB achieved an average accuracy of 84.78%, significantly higher than the average accuracies of OVA (81.74%), OVF (83.61%), and OVN (73.25%).

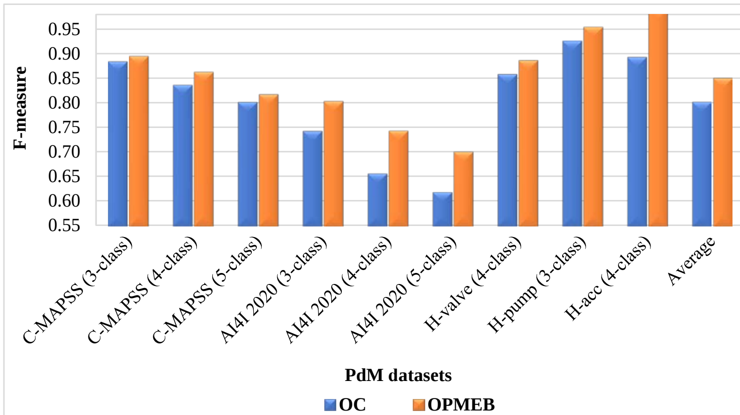

Figure 4. Comparison of the OC [4] and OPMEB (proposed) algorithm results on the PdM datasets in terms of F-measure.

Table 4. Comparison of the OVA [32], OVF [33], OVN [33] approaches, and the OPMEB algorithm results on PdM datasets in terms of accuracy (%).

Dataset	Accuracy (%)			
	OVA	OVF	OVN	OPMEB
C-MAPSS (3-class)	88.61	88.25	88.11	**89.26**
C-MAPSS (4-class)	83.27	84.26	83.49	**86.02**
C-MAPSS (5-class)	78.88	80.71	75.49	**81.46**
AI4I 2020 (3-class)	78.38	78.67	71.22	**80.09**
AI4I 2020 (4-class)	69.66	71.76	57.92	**73.94**
AI4I 2020 (5-class)	63.11	68.50	49.70	**69.72**
H-valve (4-class)	85.82	**88.38**	69.05	**88.38**
H-pump (3-class)	93.16	**95.21**	81.86	**95.21**
H-acc (4-class)	94.71	96.72	82.41	**98.91**
Average	81.74	83.61	73.25	**84.78**

5. Conclusion and future works

Over time, predictive maintenance (PdM) has gained significant attention as effective maintenance strategies become crucial for ensuring continuous production, and extended machine lifespan. PdM, driven by machine learning models and data analysis, has become a key player in improving equipment efficiency and reliability while minimizing operational expenses. Its widespread adoption across industries has made PdM an essential aspect of modern industrial practices, offering numerous competitive advantages, such as cost reduction in maintenance, enhanced product quality, improved system efficiency, and reliability.

This paper introduces a novel ordinal binary decomposition algorithm, OPMEB, and conducts comprehensive comparisons to assess its performance. The proposed OPMEB approach demonstrates its effectiveness in the PdM domain as it builds a classification model that takes into account the ranking of the health status of machines. The main purpose of this work was to showcase that the OPMEB algorithm exhibits higher efficiency in the context of OC, particularly within the domain of PdM. To validate our approach, the OPMEB method was applied to the C-MAPSS, AI4I 2020, and a real-world hydraulic system's condition datasets, encompassing different domains. The results were assessed based on four distinct evaluation metrics: accuracy, recall, precision, and F-measure. On average, there is an enhancement in accuracy across all datasets, rising from 79.80% with the OC method to 84.78% with the OPMEB approach. Our findings consistently revealed that the proposed method outperformed the OC algorithm across all evaluation metrics for all datasets, underscoring its superiority in PdM applications. In addition, the OPMEB method was compared to other ordinal binary decomposition approaches in the literature, such as OneVsAll (OVA), OneVsFollowers (OVF) and OneVsNext (OVN) to evaluate its performance comprehensively. The OPMEB method attains an average accuracy of 84.78% across all datasets, markedly surpassing the average accuracies of OVA (81.74%), OVF (83.61%), and OVN (73.25%). Once again, the results demonstrated the superiority of the OPMEB algorithm, surpassing the other approaches. As a result, this comparison highlights the effectiveness of our proposed approach, leading to the conclusion that the OPMEB approach has an important potential for achieving higher success rates in PdM applications.

The principal findings of this study can be briefly summarized as follows. The ordinal classification algorithm consistently outperformed the nominal classification algorithm across all PdM datasets. The proposed novel algorithm, OPMEB, demonstrated significant superiority over the traditional OC approach across three different PdM datasets, as evidenced by all evaluation metrics. Notably, OPMEB surpassed the traditional OC algorithm by an impressive average margin of 5.98% in terms of accuracy, underscoring its advancement in PdM tasks. Moreover, in specific instances such as the H-acc dataset, this margin for accuracy increased, with OPMEB's improvement reaching almost 10%. OPMEB showcased its effectiveness by achieving higher accuracy compared to other OBD algorithms, namely OVA, OVF, and OVN, across various PdM datasets. Although the accuracy values were equal in a few instances, upon evaluating all remaining results and looking at the average performance, it becomes evident that OPMEB outperforms its counterparts. While our method was specifically applied to the PdM domain, the OPMEB method is applicable to various other real-world OC problems. As such, we anticipate that our proposed technique will make valuable contributions to different business domains in future research studies. Thanks to the advantages of the proposed method such as ease of implementation, scalability, and compatibility, it can be integrated and operated with existing industrial systems. It can be used to give rise to groundbreaking solutions and practical applications in the field of the industrial technologies such as digital twin, internet of things, virtual/augmented reality.

In future work, we can improve the OPMEB algorithm by trying out different base learner algorithms and conducting parameter tuning for these learners. This exploration may yield improved prediction accuracy within the PdM domain.

References

[1] Susto GA, Schirru A, Pampuri S, McLoone S, Beghi A. Machine learning for predictive maintenance: a multiple classifier approach. IEEE Transactions on Industrial Informatics 2015; 11 (3): 812-820. https://doi.org/10.1109/TII.2014.2349359

[2] Altan A, Karasu S. The effect of kernel values in support vector machine to forecasting performance of financial time series and cognitive decision making. The Journal of Cognitive Systems 2019; 4 (1): 17-21.

[3] Karasu S, Altan A. Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. Energy 2022; 242. https://doi.org/10.1016/j.energy.2021.122964

[4] Frank E, Hall M. A simple approach to ordinal classification. In: European Conference on Machine Learning; Freiburg, Germany; 2001. pp. 145-156. https://doi.org/10.1007/3-540-44795-4_13

[5] Gutiérrez PA, Pérez-Ortiz M, Sánchez-Monedero J, Fernández-Navarro F, Hervás-Martínez C. Ordinal regression methods: survey and experimental study. IEEE Transactions on Knowledge and Data Engineering 2016; 28 (1): 127-146. https://doi.org/10.1109/TKDE.2015.2457911

[6] Yıldırım P, Birant UK, Birant D. EBOC: Ensemble-based ordinal classification in transportation. Journal of Advanced Transportation 2019. https://doi.org/10.1155/2019/7482138

[7] Daş DB, Birant D. Ordered physical human activity recognition based on ordinal classification. Turkish Journal of Electrical Engineering and Computer Sciences 2021; 29 (5): 2416-2436. https://doi.org/10.3906/elk-2010-75

[8] Rosati R, Romeo L, Vargas VM, Gutiérrez PA, Hervás-Martínez C et al. A novel deep ordinal classification approach for aesthetic quality control classification. Neural Computing and Applications 2022; 34: 11625–11639. https://doi.org/10.1007/s00521-022-07050-6

[9] Vega-Márquez B, Nepomuceno-Chamorro IA, Rubio-Escudero C, Riquelme JC. OCEAn: Ordinal classification with an ensemble approach. Information Sciences 2021; 580: 221-242. https://doi.org/10.1016/j.ins.2021.08.081

[10] Ünal F, Birant D, Şeker Ö. A new approach: semisupervised ordinal classification. Turkish Journal of Electrical Engineering and Computer Sciences 2021; 29 (3): 1797-1820. https://doi.org/10.3906/elk-2008-148

[11] Taser PY. An ordinal multi-dimensional classification (OMDC) for predictive maintenance. Computer Systems Science and Engineering 2023; 44 (2): 1499-1516. https://doi.org/10.32604/csse.2023.028083

[12] Jain M, Vasdev D, Pal K, Sharma V. Systematic literature review on predictive maintenance of vehicles and diagnosis of vehicle's health using machine learning techniques. Computational Intelligence 2022; 38 (6): 1990-2008. https://doi.org/10.1111/coin.12553

[13] Stanton I, Munir K, Ikram A, El-Bakry M. Predictive maintenance analytics and implementation for aircraft: challenges and opportunities. Systems Engineering 2023; 26 (2): 216-237. https://doi.org/10.1002/sys.21651

[14] Zhuang L, Xu A, Wang X. A prognostic driven predictive maintenance framework based on Bayesian deep learning. Reliability Engineering & System Safety 2023; 234. https://doi.org/10.1016/j.ress.2023.109181

[15] Mitici M, Pater I, Barros A, Zeng Z. Dynamic predictive maintenance for multiple components using data-driven probabilistic RUL prognostics: the case of turbofan engines. Reliability Engineering & System Safety 2023; 234. https://doi.org/10.1016/j.ress.2023.109199

[16] Lee J, Mitici M. Deep reinforcement learning for predictive aircraft maintenance using probabilistic Remaining-Useful-Life prognostics. Reliability Engineering & System Safety 2023; 230. https://doi.org/10.1016/j.ress.2022.108908

[17] Rodriguez PC, Marti-Puig P, Caiafa CF, Serra-Serra M, Cusidó J et al. Exploratory analysis of SCADA data from wind turbines using the k-means clustering algorithm for predictive maintenance purposes. Machines 2023; 11 (2): 270. https://doi.org/10.3390/machines11020270

[18] Granados DP, Ruiz MAO, Acosta JM, Lara SAG, Domínguez RAG et al. A wind turbine vibration monitoring system for predictive maintenance based on machine learning methods developed under safely controlled laboratory conditions. Energies 2023; 16 (5): 2290. https://doi.org/10.3390/en16052290

[19] Sun L, Liu T, Xie Y, Zhang D, Xia X. Real-time power prediction approach for turbine using deep learning techniques. Energy 2021; 233. https://doi.org/10.1016/j.energy.2021.121130

[20] Jiménez AA, Zhang L, Muñoz CQG, Márquez FPG. Maintenance management based on machine learning and nonlinear features in wind turbines. Renewable Energy 2020; 146: 316-328. https://doi.org/10.1016/j.renene.2019.06.135

[21] Hsu C, Lu Y, Yan J. Temporal convolution-based long-short term memory network with attention mechanism for remaining useful life prediction. IEEE Transactions on Semiconductor Manufacturing 2022; 35 (2): 220-228. https://doi.org/10.1109/TSM.2022.3164578

[22] Kang Z, Catal C, Tekinerdogan B. Remaining useful life (RUL) prediction of equipment in production lines using artificial neural networks. Sensors 2021; 21 (3): 932. https://doi.org/10.3390/s21030932

[23] Abdelli K, Grießer H, Pachnicke S. A machine learning-based framework for predictive maintenance of semiconductor laser for optical communication. Journal of Lightwave Technology 2022; 40 (14): 4698-4708. https://doi.org/10.1109/JLT.2022.3163579

[24] Arena S, Florian E, Zennaro I, Orrù PF, Sgarbossa F. A novel decision support system for managing predictive maintenance strategies based on machine learning approaches. Safety Science 2022; 146. https://doi.org/10.1016/j.ssci.2021.105529

[25] Kiangala KS, Wang Z. An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment. IEEE Access 2020; 8: 121033-121049. https://doi.org/10.1109/ACCESS.2020.3006788

[26] Wang Q, Liu J, Wei B, Chen W, Xu S. Investigating the construction, training, and verification methods of k-means clustering fault recognition model for rotating machinery. IEEE Access 2020; 8: 196515-196528. https://doi.org/10.1109/ACCESS.2020.3028146

[27] Hu L, Guoyong D. Estimate remaining useful life for predictive railways maintenance based on LSTM autoencoder. Neural Computing and Applications 2022. https://doi.org/10.1007/s00521-021-06051-1

[28] Ferreño D, Sainz-Aja JA, Carrascal IA, Cuartas M, Pombo J et al. Prediction of mechanical properties of rail pads under in-service conditions through machine learning algorithms. Advances in Engineering Software 2021; 151. https://doi.org/10.1016/j.advengsoft.2020.102927

[29] Azari MS, Flammini F, Santini S, Caporuscio M. A systematic literature review on transfer learning for predictive maintenance in industry 4.0. IEEE Access 2023; 11: 12887-12910. https://doi.org/10.1109/ACCESS.2023.3239784

[30] Surucu O, Gadsden SA, Yawney J. Condition monitoring using machine learning: a review of theory, applications, and recent advances. Expert Systems with Applications 2023; 221. https://doi.org/10.1016/j.eswa.2023.119738

[31] Orrù PF, Zoccheddu A, Sassu L, Mattia C, Cozza R et al. Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. Sustainability 2020; 12 (11): 4776. https://doi.org/10.3390/su12114776

[32] Kreßel U. Pairwise classification and support vector machines. In: Schölkopf B, Burges C, Smola AJ (editors). Advances in Kernel Methods: Support Vector Learning. Cambridge, MA, USA: MIT Press, 1999, pp. 255–268.

[33] Kwon Y, Han I, Lee K. Ordinal pairwise partitioning (OPP) approach to neural networks training in bond rating. Intelligent Systems in Accounting, Finance & Management 1997; 6 (1): 23-40.

[34] Witten IH, Frank E, Hall M, Pal CJ. Data Mining: Practical Machine Learning Tools and Techniques. 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2016.

[35] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: Proceedings of the 1st International Conference on Prognostics and Health Management (PHM08); Denver CO, USA; 2008. pp. 1-9. https://doi.org/10.1109/PHM.2008.4711414

[36] Matzka S. Explainable artificial intelligence for predictive maintenance applications. In: Third International Conference on Artificial Intelligence for Industries (AI4I); Irvine, CA, USA; 2020. pp. 69-74. https://doi.org/10.1109/AI4I49448.2020.00023

[37] Helwig N, Pignalelli E, Schütze A. Condition monitoring of a complex hydraulic system using multivariate statistics. In: IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings; Pisa, Italy; 2015. pp. 210-215. https://doi.org/10.1109/I2MTC.2015.7151267