*Computer Systems Science & Engineering*
DOI: 10.32604/csse.2023.028083
***Article***

Tech Science Press

# An Ordinal Multi-Dimensional Classification (OMDC) for Predictive Maintenance

**Pelin Yildirim Taser***

Izmir Bakircay University, Department of Computer Engineering, Izmir, 35665, Turkey
*Corresponding Author: Pelin Yildirim Taser. Email: pelin.taser@bakircay.edu.tr

Received: 02 February 2022; Accepted: 04 March 2022

**Abstract:** Predictive Maintenance is a type of condition-based maintenance that assesses the equipment's states and estimates its failure probability and when maintenance should be performed. Although machine learning techniques have been frequently implemented in this area, the existing studies disregard to the natural order between the target attribute values of the historical sensor data. Thus, these methods cause losing the inherent order of the data that positively affects the prediction performances. To deal with this problem, a novel approach, named *Ordinal Multi-dimensional Classification (OMDC)*, is proposed for estimating the conditions of a hydraulic system's four components by taking into the natural order of class values. To demonstrate the prediction ability of the proposed approach, eleven different multi-dimensional classification algorithms (traditional Binary Relevance (BR), Classifier Chain (CC), Bayesian Classifier Chain (BCC), Monte Carlo Classifier Chain (MCC), Probabilistic Classifier Chain (PCC), Classifier Dependency Network (CDN), Classifier Trellis (CT), Classifier Dependency Trellis (CDT), Label Powerset (LP), Pruned Sets (PS), and Random k-Labelsets (RAKEL)) were implemented using the Ordinal Class Classifier (OCC) algorithm. Besides, seven different classification algorithms (Multilayer Perceptron (MLP), Support Vector Machine (SVM), k-Nearest Neighbour (kNN), Decision Tree (C4.5), Bagging, Random Forest (RF), and Adaptive Boosting (AdaBoost)) were chosen as base learners for the OCC algorithm. The experimental results present that the proposed OMDC approach using binary relevance multi-dimensional classification methods predicts the conditions of a hydraulic system's multiple components with high accuracy. Also, it is clearly seen from the results that the OMDC models that utilize ensemble-based classification algorithms give more reliable prediction performances with an average Hamming score of 0.853 than the others that use traditional algorithms as base learners.

**Keywords:** Machine learning; multi-dimensional classification; ordinal classification; predictive maintenance

## 1 Introduction

In industrial systems, equipment maintenance dramatically influences an equipment's operation time and efficiency [1]. *Predictive Maintenance (PdM),* which is one of the maintenance management strategies, monitors the equipment's performance and condition to reduce the probability of failure using predictive tools. The main aim of the PdM is to monitor equipment's condition and estimate the potential equipment failure based on historical data. Thus, it provides preventing equipment failure with regularly scheduled and corrective maintenance.

*Machine learning (ML)* techniques have been preferred as a powerful predictive tool for maintaining industrial environments in recent years [2]. It discovers hidden patterns within complex and multivariate data where the traditional methods (e.g., statistical inference methods) remain inadequate. In most of the ML-based PdM approaches, Multilayer Perceptron (MLP) [3,4], Convolutional Neural Network (CNN) [5–7], Long Short-term Memory Network (LSTM) [8–10], Support Vector Machine (SVM) [11,12], k-Nearest Neighbour (kNN) [13], Random Forest (RF) [14–17], and Decision Tree [18] were used. For example, Kiangala et al. [5] proposed a novel PdM framework that combines time-series imaging and CNN models to detect a conveyor system's impairments and reduce the risk of incorrect faults. According to the experimental results, the proposed method outperforms the traditional classification approaches. In another study [18], a novel decision support system based on decision trees (DTs) were developed for determining under what circumstances a machine learning-based PdM strategy is economically viable, with the goal of assisting practitioners in making sound decisions.

In addition to these single label predictive maintenance approaches, a few ML-based approaches classify multi-target attributes [19,20]. For example, Last et al. [19] developed a multi-target probability estimation model (M-IFN) to estimate the probability and timing of a failure in each subsystem in a car. In another study [20], a multi-label recurrent neural network model was presented for predicting faults of multi-dimensional time-series data. The proposed approaches in both of these studies give more accurate prediction performances than the conventional solutions. In this study, eleven different *multi-dimensional classification (MDC)* algorithms (BR, CC, BCC, PCC, MCC, CDN, CT, CDT, LP, PS, and RAKEL algorithms) are preferred for estimating the conditions of a hydraulic system's four components.

In the existing ML-based PdM approaches, it is assumed that there is no inherent natural order between the class values in the experimental data. However, in most real-world scenarios, the experimental datasets' class values have an order relation among them. As shown in the previous studies [21,22], the ordinal classification strategy often outperforms the nominal classification for the datasets consisting of class labels having ordered information. Considering this motivation, a novel approach, named *Ordinal Multi-dimensional Classification (OMDC)*, is proposed for predicting multi-dimensional attributes by taking into the natural order of class values. This approach combines *MDC* and *ordinal classification (OC)* paradigms to improve prediction performance. The OMDC is a general approach that can be used for a prediction task in various fields. It was utilized to predict the conditions of a hydraulic system's four components as a case study. In the experiments, the proposed OMDC approach was applied to a well-known and real-world hydraulic system's condition dataset used in many ML-based studies [23–25].

The novelty and main contributions of this article are as follows: (i) This study gives a brief survey of MDC and OC, (ii) It is the first attempt to implement the OC method for predictive maintenance, (iii) It proposes a novel approach, named OMDC, that combines *MDC* and *OC* paradigms to improve prediction performance, (iv) This study is also original in that it implements the OMDC approach to classify the conditions of the hydraulic system's four components by considering the relationships between the class values of components, (v) This is also the first study that classifies multiple components of the hydraulic systems, (vi) It implements eleven different MDC algorithms using MLP, SVM, kNN, C4.5, Bagging, RF, and AdaBoost classification algorithms as base learners and compared with each other in terms of Hamming Score, Exact Match, Hamming Loss, and ZeroOne Loss metrics.

## 2 Background Information

### *2.1 Ordinal Classification*

ML techniques are generally grouped under two main types: supervised learning and unsupervised learning. Classification, a commonly applied supervised learning method, assigns unlabeled data to predefined classes using an accurate model. The multi-class classification process is divided into two categories: nominal classification and ordinal classification. The nominal classification method assumes that there is no order between the target attribute's class labels. However, in most real-world scenarios, the datasets' class labels can have a natural order. For example, the internal pump leakage of hydraulic systems can be classified as “no leakage”, “weak leakage”, and “severe leakage”. The order among these class labels is stated as “severe leakage” > “weak leakage” > “no leakage”. To deal with this problem, an *ordinal classification* (OC) paradigm was proposed.

In the OC technique, a new sample's ordinal target value is predicted by considering the predefined classes' ordering relation [26]. The ordinal dataset $D = \{(x_i,\ \ y_i)_{i=1}^{m}\ \epsilon\ X\ x\ Y\}$ has $m$ items, where $X\ \epsilon\ R^d$ and the class labels $y_i\ \epsilon\ Y = \{C_1, C_2, \ldots, C_k\}$ having $k$ classes with a linear order relation $C_1 <\ C_2 < \cdots < C_k$. The OC aims to categorize a new sample $x_i$ with one of the class labels $y_i$ with a ranking structure.

The traditional nominal classification algorithms can be used to solve OC problems by ignoring the existing inherent order among the class labels. However, this approach causes losing the natural order of the data that gets more accurate classification performance. To handle this problem, in this study, the Ordinal Class Classifier (OCC) algorithm [27] that converts the OC problem to multiple binary classification problems was preferred for predicting the conditions of four hydraulic components (cooler, valve, internal pump leakage, and hydraulic accumulator). This algorithm converts $k$ different ordinal class labels $C_1$, $C_2$, …, $C_k$ of the ordinal dataset $D$ to binary values regarding their natural order $C_1 <\ C_2 < \cdots < C_k$. According to this approach, first, the $C_1$ classes are labeled as 0, and the classes with higher ranking order than $C_1\ \ (C_2,\ \ldots,\ \ C_k)$ are labeled as 1. In the next iteration, while the class values that are higher than $C_2\ \ (C_3,\ \ldots,\ \ C_k)$ are set as 1, the rest of them ($C_1$ and $C_2$) are set as 0. The exact process is continued to implement until there is no ordered label remains.

The OCC algorithm requires a base learner for predicting the ordinal-valued target attribute. In this study, MLP, SVM, kNN, C4.5, Bagging, RF, and AdaBoost classification algorithms were preferred as base learners for the OCC algorithm to classify multiple attributes of the experimental dataset.

- **MLP:** The MLP is a feedforward artificial neural network algorithm that consists of multiple interconnected layers of nodes in a weighted directed graph [28]. There are three types of layers in this algorithm: input layer, output layer, and hidden layer. The sample data is taken from the input layer, processed in the hidden layers as the computational engine, and then classified with a proper label by the output layer.
- **SVM:** In the SVM algorithm, each instance in the dataset is placed as a point in $n$-dimensional space, where $n$ is the attribute number [29]. After that, a hyperplane that separates two different classes is constructed to perform the instances' classification. The key point of this approach's success is choosing the hyperplane with maximum margin to avoid overfitting and reducing misclassification results.
- **kNN:** The kNN classifier is non-parametric and instance-based learning that classifies a new sample with a majority vote of its $k$ nearest neighbors with specific class labels [30]. To determine $k$ nearest

neighbors of a new sample, well-known distance metrics in the literature are used, such as Euclidean, Manhattan, Minkowski, etc.

- **C4.5:** C4.5 is the most commonly preferred decision tree algorithm that is grown using the depth-first strategy with the conjunction of rules [31]. The decision tree includes internal nodes, branches, and leaf nodes representing attributes, attribute values, and classes, respectively. Information gain and entropy metrics are used for the construction of the decision tree in this algorithm.
- **Bagging:** Bagging is an ensemble learning technique that generates multiple classification models from the multiple training sets created by choosing random and repeatable instances from the original dataset [32]. To classify a new sample using the Bagging method, each model produces an output, and then, these outputs obtained from each model are aggregated to get the single final decision.
- **RF:** The RF, a bagging-based ensemble classifier, creates a forest consisting of multiple decision trees constructed using a random feature subset selection method [33]**.** In this approach, the classification process of a new sample is performed by selecting a majority vote of all trees' predictions in the forest.
- **AdaBoost:** In the AdaBoost algorithm, the classification models are trained consecutively to convert weak learners to strong ones [34]. The algorithm increases the misclassified samples' weight values and decreases the correctly classified ones in each iteration.

### *2.2 Multi-Dimensional Classification*

The traditional classification process, one of the most well-established machine learning paradigms, assumed that a learning task has only one objective (single-label). However, in real-world classification problems, there can be several multiple target attributes (multi-dimensional) that need to be classified. For example, predicting multiple diagnoses of a given patient, text categorization, analysis of nonverbal expressions in speech, etc. These target attributes can be related, semi-related or unrelated to each other. The conventional classification algorithms are not coping well with the multi-dimensional decision-making problems. Therefore, a *multi-dimensional classification (MDC)* paradigm has emerged as a solution for this problem.

MDC is a subfield of supervised learning that simultaneously predicts multiple target attributes of a given single new sample. Let $D = \{(x_i,\ y_i)_{i=1}^{m}\ \epsilon\ X\ x\ Y\}$ be a multi-dimensional dataset with $m$ instances, where each instance $x_i = [x_{i1}, x_{i2}, \ldots, x_{ik}]$ is an input vector with $k$ attributes ($k \geq 2$) and $y_i = [y_{i1}, y_{i2}, \ldots, y_{in}]$ is a label vector including $n$ dimensions. Each target attribute has $d$ possible class labels, where $d \geq 1$. MDC aims to construct a function $f{:}X \rightarrow Y$ that predicts a single input sample's multiple target values by optimizing evaluation metrics.

MDC is a particular case of multi-label classification, where each label contains only binary values representing relevance or not, as opposed to having multiple values. MDC methods are closely related to multi-label classification approaches. Several multi-label classification algorithms in the literature, which apply the problem transformation approach, proposed a solution for MDC problems. In the problem transformation method, multi-dimensional problems are divided into multiple traditional single-label problems. The problem transformation approach is implemented using two standard methods: binary relevance and label powerset.

#### *2.2.1 Binary Relevance*

Binary relevance, one of the widespread methods of the problem transformation approach, trains several multi-class classifiers using a one-against-all strategy. In other words, the MDC problem is converted into multiple multi-class problems. In this study, the traditional BR method (BR) was implemented on the experimental dataset.

- **BR:** The BR algorithm transforms the MDC problem into several well-established multi-class classification problems [35]. In this method, conventional multiple multi-class classifiers are trained on the dataset independently for predicting each target attribute value. The BR algorithm is commonly preferred in MDC problems because of its simplicity and having linear complexity. However, it is unable to capture the relationships among the target attributes.

The learning system's generalization performance is decreased because of the BR method's ignorance of the correlations between the target attributes. To overcome this problem, the implementation of the BR method with chaining structure is proposed. The target attribute information is passed between classifiers linked in a chain to consider label correlations in the chaining strategy. In this study, the conventional CC algorithm, its different improved versions such as BCC, MCC, PCC, and CDN were used.

- **CC**: The CC method links multiple multi-class classifiers along with a chaining order where each classifier deals with the binary relevance problem [36]. According to this approach, a multi-class classifier is constructed for each target attribute depending on the predictions of preceding classification models in the chain.
- **BCC:** The BCC algorithm implements the CC method using Bayesian networks that present the class values' dependency relations [37]. It also uses a maximum spanning tree depending on marginal label dependence to improve the classification performance.
- **PCC:** The PCC algorithm applies the CC method in the training phase of the multi-dimensional classification process [38]. However, in the testing phase, it uses Bayes-optimal inference by discovering all possible chain paths.
- **MCC:** The MCC method, which is proposed for obtaining a good chain sequence, uses a Monte Carlo optimization technique to search the space of possible chain sequences [39]. Besides the efficient inference performance it provides, the MCC algorithm also handles high-dimensional training sets.
- **CDN:** The CDN algorithms construct a cyclic directed graphical model that gives a non-rational presentation of correlations between the target attributes [40]. It predicts the input data labels using a Gibbs sampling inference.

The existing classifier chain algorithms in the literature show accurate classification performances, but they have scalability limitations on the large volume of datasets when constructing a fully cascaded chain. A classifier trellis method was proposed as a scalable solution for MDC problems to solve this high computational bottleneck. This study implements the traditional CT and CDT algorithms.

- **CT**: The CT algorithm generates a structure in which the target attributes are placed in an ordered method consistent with simply computable mutual information metrics [41]. The main advantage of this method is that it scales up to the vast number of target attributes.
- **CDT:** The CDT is an improved version of the CDN algorithm that uses a cyclic directed graphical model with a trellis structure instead of a fully connected network [42].

### *2.2.2 Label Powerset*

Label powerset transforms MDC problems into multi-class classification problems by combining all unique class labels in the training set as a new single label [43]. As a result of this process, each instance in the training set has only one target attribute with one class label. In this study, the traditional LP, PS, and RAKEL algorithms were utilized for the MDC problem.

- **LP:** The LP algorithm converts multi-dimensional data labels to an atomic label representing a distinct multi-class subset [44]. After that, the conventional multi-class classification algorithms are applied to this transformed dataset. Therefore, correlations between the target attributes are considered in this approach. The main drawback of this method is that it presents high time complexity in the worst case.

- **PS:** The PS method prunes infrequently occurring label sets to exploit the most significant correlations among the multi-dimensional dataset [45]. This approach reduces time complexity and improves classification accuracy with minimal information loss.
- **RAKEL:** The RAKEL method iteratively generates a $k$ size ensemble combination of a small random subset of labels [46]. After that, it aggregated the obtained class outputs from each classifier to get the final class value.

## 3 Ordinal Multi-Dimensional Classification (OMDC)

The proposed approach, named Ordinal Multi-Dimensional Classification (OMDC), predicts the conditions of a hydraulic system's four components (cooler, valve, internal pump leakage, and hydraulic accumulator) by considering the natural order of class labels. In this approach, well-known MDC algorithms, which apply binary relevance and label powerset methods, are used for classifying multi-target attributes of an experimental dataset. Furthermore, the conventional OCC algorithm is applied for each target attribute's class values that have an inherent order. Finally, four different traditional classification algorithms (MLP, SVM, kNN, and C.5) and three different ensemble-based classification algorithms (Bagging, RF, and AdaBoost) are preferred as base learners for the OCC algorithm in this study.

**Definition (OMDC)**. The OMDC aims to construct a classification model to predict multiple target attributes of a dataset without losing inherent order between the class values.

Let $D = \{(x_i,\ y_i)_{i=1}^{m}\ \epsilon\ X\ x\ Y\}$ be a multi-dimensional ordinal dataset with $m$ instances, where each instance $x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$ is an input vector with $d$ attributes ($d \geq 2$) and $y_i = [y_{i1}, y_{i2}, \ldots, y_{in}]$ is a label vector including $n$ dimensions. Each target attribute $y_{in} = \{C_1, C_2, \ldots, C_k\}$ has $k$ possible class labels, where $k \geq 1$, with a linear order relation $C_1, < C_2 < \cdots < C_k$. The purpose of ordinal decomposition is to convert the original ordinal classification problem involving $k$ classes into $k-1$ binary classification problems in order to encode the ordering information included within the class labels. For the $i^{th}$ binary problem, the label $y_{jn}$ is substituted by $y_{jn} = 0,\ \forall y_{jn} \leq C_i$, and $y_{jn} = 1,\ \forall y_{jn} > C_i$. The OMDC approach intend to construct a function $f{:}X \rightarrow Y$ that correctly estimates a new sample's multiple target values with a high accuracy.

In this method, the multi-dimensional classification problem is transformed into the conventional multi-class classification problem using binary relevance and label powerset methods. In the binary relevance method, each target attribute value of the dataset is estimated by conventional multi-class classifiers independently. The label powerset method combines all class values of the multiple target attributes in the multi-dimensional dataset as a new single label. To implement these methods, BR, CC, BCC, PCC, MCC, CDN, CT, CDT, LP, PS, and RAKEL algorithms are preferred in this approach.

After the problem transformation phase, the OCC algorithm is applied to convert ordinal multiple-class values to binary datasets by considering the order relationship among the labels. To give a clear example, consider the hydraulic system's condition dataset used in this study. It includes the measured process values of a hydraulic test rig with its condition properties (cooler, valve, internal pump leakage, and hydraulic accumulator) that will be predicted. The values of four different target attributes of this dataset have an inherent order. For example, the cooler condition attribute has three values with a linear order relation “full efficiency” > “reduced efficiency” > “close to total failure”. First, “close to total failure” class values are labeled as 0, and the rest are set as 1. In the next iteration, “full efficiency” class value, which has higher ranking order than “reduced efficiency” and “close to total failure”, is set as 1, and the others (“reduced efficiency” and “close to total failure”) are labeled as 0. Then, the same processes are implemented on all class labels of each target attributes in the dataset.

In this OMDC method, seven different classification algorithms (MLP, CVM, kNN, C.5, Bagging, RF, and AdaBoost) are used as base learners for the OCC algorithm. Let $M_i$ indicates the model created for the $i^{th}$ binary problem for $i = 1, 2, \ldots, m - 1$. Given an instance $x$, the class label obtained from $M_i(x)$ is regarded as an estimation of the probability $P(L > C_i|x)$, where L is the class label associated with $x$. In this algorithm, the probabilities for each $k$ class of each $n$ target attributes in the dataset are computed as follows:

$$P(C_1|\ x\ ) = 1 - P\ \ (L > C_1|\ x)$$

$$P(C_j|\ x\ ) = P(L > C_{j-1}|x) - P\ \ (L > C_j|\ x) \tag{1}$$

$$P(C_k|\ x\ ) = P(L > C_{k-1}|x)$$

where $j = \{2, 3, \ldots, k\}$.

The probabilities for the first and final classes are predicted using a single model. The probability that an example belongs to the first class $P(C_1|x)$ is equal to $1 - P\ \ (L > C_1|\ x)$, since we will evaluate just the first model that discriminates between c1 and the remainder. In a similar way, the probability of $P(C_k|\ x\ )$ is determined using the one-*vs.*-followers approach from $P(L > C_{k-1}|x)$. The probability is dependent on a pair of models for intermediate classes $1 < j < k$, because the ordinal structure of a middle class $C_j$ may be expressed by two indicators: higher than its preceding class $(P(L > C_{j-1}|x))$ and itself $(P\ \ (L > C_j|\ x))$.

The general overview of the proposed OMDC approach is given in Fig. 1. In the first step (data acquisition), various measured process values from multiple sensors (pressure, motor power, volume flows, temperature, vibration, cooling efficiency, and efficiency factor) and the conditions of the hydraulic system (cooler, valve, internal pump leakage, and hydraulic accumulator) are obtained from a hydraulic test rig. In the training step, first, the problem transformation process is applied to the raw experimental dataset. In this phase, binary relevance (BR, CC, BCC, PCC, MCC, CDN, CT, and CDT) and label powerset algorithms (LP, PS, and RAKEL) are applied for converting MDC problem into a multi-class classification problem. Then, the OCC algorithm is selected as a base learner for each binary relevance and label powerset algorithm using MLP, SVM, kNN, C4.5, Bagging, RF, and AdaBoost algorithms. Thus, the classification process is performed by preserving the inherent order between the class values. In the next step, each model's performance of the hydraulic test rig's condition estimations is evaluated using the $n$-fold cross-validation technique selecting $n$ as 10. In the prediction step, a new sample's conditions are predicted using the OMDC approach. Finally, Hamming Score, Exact Match, Hamming Loss, and ZeroOne Loss results of each OMDC model are presented in the last step.

Algorithm 1 presents the pseudocode of the proposed OMDC approach. First, the base learner of the OCC algorithm is selected from seven different classification algorithms. Then, the multi-dimensional dataset $D = \{(x_i,\ y_i)_{i=1}^{m}\ \epsilon\ X\ x\ Y\}$ is transformed into a multi-class dataset using a problem transformation method *ptt* (binary relevance or label powerset). Multiple multi-class classifiers are generated for classifying each target attribute if binary relevance is selected as a problem transformation method. Then, a new sample's target attributes are predicted with a binary relevance algorithm using the OCC as a base learner. As a second alternative, if the label powerset method is chosen, new single labels are constructed for each instance from each combination of all unique class labels. In this method, a new sample $X$* is classified with a label powerset algorithm that uses the OCC as a base learner.

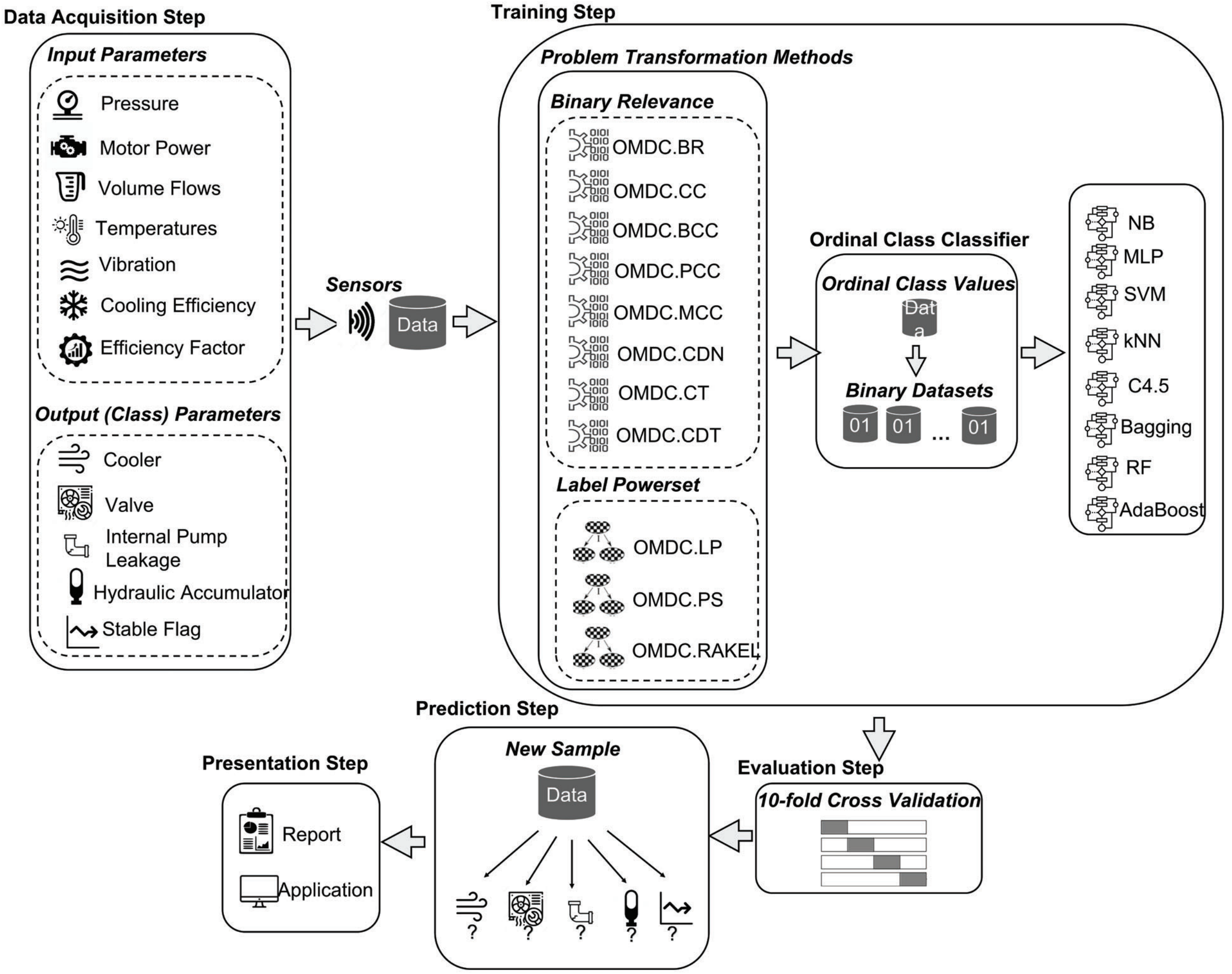

**Figure 1:** The general overview of the proposed OMDC approach

**Algorithm 1:** The pseudocode of the proposed OMDC approach

```
Algorithm OMDC: Ordinal Multi-Dimensional Classification
Inputs:
    $D$: the ordinal multi-dimensional dataset $D = \{(x_i, \ y_i)_{i=1}^{m} \epsilon X \ x \ Y\}$
    $x_i$: an instance $x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$ is an input vector with $d$ attributes in the dataset $D$
    $y_i$: the class labels $y_i = [y_{i1}, y_{i2}, \ldots, y_{in}]$ is a label vector with $n$ dimension in the dataset $D$, where each $y_{in} = \{C_1, C_2, \ldots, C_k\}$ is the set of possible class values with a linear order relation $C_1 < C_2 < C_3 < \cdots < C_k$.
    $m$: the number of instances in the dataset $D$
    $ptt$: the problem transformation type (binary relevance and label powerset)
    $B$: a base classification algorithm
    $X^*$: a new sample that will be predicted
Output:
    $Y$: predicted class vector
```

(Continued)

**Algorithm 1: (continued)**

```
Begin:
  OrdinalClassClassifier.BaseAlgorithm = B
  if (ptt = Binary Relevance)
    //Training phase
    for $i = 1$ to $n$ do
      $D_{1i} = \{\}$
        for $j = 1$ to $m$ do
          $D_{1i} = D_{1i}$ U $(x_j, y_{ji})$
        end for
        $Classifier_i$ = OrdinalClassClassifier ($D_{1i}$)
    end for
    //Classification phase
    $Y = \{\}$
    for $i = 1$ to $n$ do
      $Y = Y$ U $Classifier_i$ ( $X*$)
    end for
    return $Y$
  if (ptt = Label Powerset)
  //Training phase
  $D_2 = \{\}$
  for $i = 1$ to $m$ do
    for $j = 1$ to $n$ do
      $y_i = y_i$ U $y_{ij}$
    end for
    $D_2 = D_2$ U $(x_i, y_i)$
  end for
  Classifier = OrdinalClassClassifier ($D_2$)
  //Classification
  $Y = \{\}$
  for $i = 1$ to $n$ do
    $Y = Y$ U Classifier ($X*$)
  end for
  return $Y$
  End Algorithm
```

## 4 Experimental Study

In the experimental studies, the proposed OMDC approach was implemented on a real-world hydraulic system's condition dataset to demonstrate its classification performance. The proposed application was developed using the MEKA open-source multi-label classification framework based on the WEKA library [47] in the Java programming language. In this approach, eleven different MDC algorithms were implemented on the experimental dataset using the OCC. Furthermore, seven different classification algorithms were chosen as base learners for the OCC algorithm independently. The *k* value of the kNN algorithm was set as 3. Also, the number of iteration parameter values of AdaBoost and Bagging algorithms were selected as 100. The implemented OMDC models with different MDC algorithms were compared to each other using the *n*-fold cross-validation technique selecting *n* as 10. The classification performances of each model were evaluated using Hamming Score, Exact Match, Hamming Loss, and ZeroOne Loss metrics.

- **Hamming Score:** It evaluates the average across the ratio of the correctly classified labels to the total number of labels.
- **Exact Match:** It gives the percentage of instances with all labels correctly classified.
- **Hamming Loss:** It presents the ratio of incorrectly classified labels to the total number of labels.
- **ZeroOne Loss:** It gives the ratio of instances whose actual value is not equal to the predicted value.

These evaluation metrics (Hamming Score, Exact Match, Hamming Loss, and ZeroOne Loss) are evaluated as shown in Eqs. (2)–(5), respectively.

$$Hamming\ Score = \frac{1}{|N|}\sum_{i=1}^{|N|}\frac{|y_i \cap_i^p|}{|y_i \cup_i^p|} \tag{2}$$

$$Exact\ Match = \frac{1}{|N|}\sum_{i=1}^{|N|} I(y_i = p_i) \tag{3}$$

$$Hamming\ Loss = \frac{1}{|N||L|}\sum_{i=1}^{|N|}\sum_{j=1}^{|L|} I(y_{ij} \neq p_{ij}) \tag{4}$$

$$ZeroOne\ Loss = \frac{1}{|N|}\sum_{i=1}^{|N|} I(y_i \neq p_i) \tag{5}$$

where *I* refers indicator function, *p* is predicted value, *y* represents actual value, $|N|$ is the number of instances, and $|L|$ is the number of target attributes.

### *4.1 Dataset Description*

This paper introduces a new concept called OMDC and demonstrates its efficiency on a real-world hydraulic system's condition dataset [48]. It is constructed by a hydraulic test rig's measured process values from multiple sensors (pressure, motor power, volume flows, temperature, vibration, cooling efficiency, and efficiency factor) and the conditions (cooler, valve, internal pump leakage, and hydraulic accumulator). It consists of 2205 instances with seventeen input and four target attributes. Tabs. 1 and 2 present the basic characteristics of the experimental dataset's input and target attributes, respectively.

**Table 1:** The characteristics of numerical-valued input attributes

| Input attributes | Unit | Sampling rate | Min. value | Max. value |
|---|---|---|---|---|
| PS1 pressure | bar | 100 Hz | 155.392 | 180.923 |
| PS2 pressure | bar | 100 Hz | 104.406 | 131.589 |
| PS3 pressure | bar | 100 Hz | 0.84 | 2.023 |
| PS4 pressure | bar | 100 Hz | 0 | 10.207 |
| PS5 pressure | bar | 100 Hz | 8.366 | 9.979 |
| PS6 pressure | bar | 100 Hz | 8.322 | 9.857 |
| EPS1 motor power | w | 100 Hz | 2361.747 | 2740.641 |
| FS1 volume flow | I/min | 10 Hz | 2.019 | 6.723 |
| FS2 volume flow | I/min | 10 Hz | 8.858 | 10.403 |
| TS1 temperature | °C | 1 Hz | 35.314 | 57.899 |
| TS2 temperature | °C | 1 Hz | 40.859 | 61.958 |
| TS3 temperature | °C | 1 Hz | 38.246 | 59.423 |
| TS4 temperature | °C | 1 Hz | 30.391 | 53.06 |
| VS1 vibration | mm/s | 1 Hz | 0.524 | 0.839 |
| CE cooling efficiency | % | 1 Hz | 17.556 | 47.904 |
| CP cooling power | kW | 1 Hz | 1.062 | 2.84 |
| SE efficiency factor | % | 1 Hz | 18.277 | 60.755 |

**Table 2:** The characteristics of ordinal-valued target attributes

| Target attributes | Labels |
|---|---|
| Cooler | Close to total failure |
| | Reduced efficiency |
| | Full efficiency |
| Valve | Optimal switching behavior |
| | Small lag |
| | Severe lag |
| | Close to total failure |
| Internal pump leakage | No leakage |
| | Weak leakage |
| | Severe leakage |
| Hydraulic accumulator | Optimal pressure |
| | Slightly reduced pressure |
| | Severely reduced pressure |
| | Close to total failure |

The proposed OMDC is a general approach that can be applied to any field using any experimental dataset. Although the used dataset in this study is publicly available, the OMDC methodology can also be applied to private datasets by safeguarding them using some coverless information hiding techniques [49,50].

### 4.2 *Results*

In the experiments of this study, the proposed OMDC approach was executed on the hydraulic system's condition dataset to demonstrate its high accurate classification performance. The MDC algorithms were applied in the OMDC approach using the OCC algorithm with seven different classification algorithms as a base learner. The generated MDC models were compared with each other in terms of some evaluation metrics (Hamming Score, Exact Match, Hamming Loss, and ZeroOne Loss).

Tabs. 3 and 4 represent the Hamming Score and Exact Match results of the implemented OMDC models using the traditional and ensemble-based classification algorithms as base learners on the experimental dataset, respectively. The Hamming Score and Exact Match metrics give a degree ranging from 0 to 1 to present the success of the proposed algorithm. The fact that these values are close to 1 indicates that the model offers a classification ability with high accuracy. Furthermore, the evaluated Hamming Loss and ZeroOne Loss results of the same OMDC models are presented in Tabs. 5 and 6. Unlike Hamming Score and Exact Match, there is an inverse ratio between the classification ability of the model, and the Hamming Loss and ZeroOne Loss values. It means that the classification models having lower Hamming Loss and ZeroOne Loss values provide higher prediction performances than the others. According to the results obtained from these tables, it is clearly seen that OMDC.PCC model using AdaBoost algorithm as a base learner presents the highest Hamming Score and Exact Match values among the other OMDC models with 0.989 and 0.959, respectively. Also, it is observed that OMDC.CC, OMDC.BCC, OMDC. MCC, OMDC.CDN, and OMDC.CT algorithms using AdaBoost and RF base learners reached high classification abilities quite close to Hamming Score and Exact Match values of OMDC.PCC. Furthermore, when the obtained results are considered in general, it is possible to say that the OMDC models with binary relevance algorithms are more successful than the label powerset algorithms for predicting the hydraulic system's multiple conditions.

**Table 3:** Hamming score results of the OMDC models

| | | Traditional algorithms | | | | Ensemble-based algorithms | | |
|---|---|---|---|---|---|---|---|---|
| | | MLP | SVM | kNN | C.5 | Bagging | RF | AdaBoost |
| Binary relevance | OMDC.BR | 0.612 | 0.56 | 0.633 | 0.629 | 0.636 | 0.639 | 0.639 |
| | OMDC.CC | 0.866 | 0.707 | 0.947 | 0.961 | 0.979 | 0.987 | 0.988 |
| | OMDC.BCC | 0.872 | 0.674 | 0.951 | 0.959 | 0.979 | 0.987 | 0.988 |
| | OMDC.PCC | 0.877 | 0.709 | 0.948 | 0.961 | 0.979 | 0.988 | 0.989 |
| | OMDC.MCC | 0.877 | 0.709 | 0.948 | 0.96 | 0.979 | 0.988 | 0.988 |
| | OMDC.CDN | 0.815 | 0.703 | 0.877 | 0.953 | 0.948 | 0.942 | 0.988 |
| | OMDC.CT | 0.877 | 0.683 | 0.949 | 0.96 | 0.979 | 0.987 | 0.988 |
| | OMDC.CDT | 0.81 | 0.702 | 0.885 | 0.953 | 0.948 | 0.944 | 0.986 |
| Label powerset | OMDC.LP | 0.588 | 0.46 | 0.625 | 0.616 | 0.631 | 0.636 | 0.634 |
| | OMDC.PS | 0.587 | 0.46 | 0.624 | 0.616 | 0.631 | 0.636 | 0.633 |
| | OMDC.RAKEL | 0.587 | 0.46 | 0.624 | 0.616 | 0.631 | 0.636 | 0.633 |

**Table 4:** Exact match results of the OMDC models

| | | Traditional algorithms | | | | Ensemble-based algorithms | | |
|---|---|---|---|---|---|---|---|---|
| | | MLP | SVM | kNN | C.5 | Bagging | RF | AdaBoost |
| Binary relevance | OMDC.BR | 0.155 | 0.12 | 0.174 | 0.171 | 0.179 | 0.181 | 0.181 |
| | OMDC.CC | 0.565 | 0.326 | 0.83 | 0.858 | 0.927 | 0.957 | 0.958 |
| | OMDC.BCC | 0.558 | 0.24 | 0.832 | 0.853 | 0.927 | 0.956 | 0.958 |
| | OMDC.PCC | 0.591 | 0.331 | 0.833 | 0.86 | 0.927 | 0.959 | 0.959 |
| | OMDC.MCC | 0.59 | 0.331 | 0.832 | 0.858 | 0.927 | 0.959 | 0.958 |
| | OMDC.CDN | 0.418 | 0.323 | 0.671 | 0.834 | 0.81 | 0.787 | 0.957 |
| | OMDC.CT | 0.579 | 0.257 | 0.832 | 0.856 | 0.927 | 0.957 | 0.956 |
| | OMDC.CDT | 0.41 | 0.319 | 0.679 | 0.836 | 0.81 | 0.794 | 0.953 |
| Label powerset | OMDC.LP | 0.138 | 0.036 | 0.172 | 0.167 | 0.172 | 0.178 | 0.178 |
| | OMDC.PS | 0.137 | 0.036 | 0.171 | 0.167 | 0.172 | 0.178 | 0.177 |
| | OMDC.RAKEL | 0.137 | 0.036 | 0.171 | 0.167 | 0.172 | 0.178 | 0.177 |

**Table 5:** Hamming loss results of the OMDC models

| | | Traditional algorithms | | | | Ensemble-based algorithms | | |
|---|---|---|---|---|---|---|---|---|
| | | MLP | SVM | kNN | C.5 | Bagging | RF | AdaBoost |
| Binary relevance | OMDC.BR | 0.388 | 0.44 | 0.367 | 0.371 | 0.364 | 0.361 | 0.361 |
| | OMDC.CC | 0.134 | 0.293 | 0.053 | 0.039 | 0.021 | 0.013 | 0.012 |
| | OMDC.BCC | 0.128 | 0.326 | 0.049 | 0.041 | 0.021 | 0.013 | 0.012 |
| | OMDC.PCC | 0.123 | 0.291 | 0.052 | 0.039 | 0.021 | 0.012 | 0.011 |
| | OMDC.MCC | 0.123 | 0.291 | 0.052 | 0.04 | 0.021 | 0.012 | 0.012 |
| | OMDC.CDN | 0.185 | 0.297 | 0.123 | 0.047 | 0.052 | 0.058 | 0.012 |
| | OMDC.CT | 0.123 | 0.317 | 0.051 | 0.04 | 0.021 | 0.013 | 0.012 |
| | OMDC.CDT | 0.19 | 0.298 | 0.115 | 0.047 | 0.052 | 0.056 | 0.014 |
| Label powerset | OMDC.LP | 0.412 | 0.54 | 0.375 | 0.384 | 0.369 | 0.364 | 0.366 |
| | OMDC.PS | 0.413 | 0.54 | 0.376 | 0.384 | 0.369 | 0.364 | 0.367 |
| | OMDC.RAKEL | 0.413 | 0.54 | 0.376 | 0.384 | 0.369 | 0.364 | 0.367 |

The average of the Hamming score values obtained by each OMDC model using seven different classification algorithms was evaluated, and the results are illustrated in the graph given in Fig. 2. The comparative results in this graph state that the OMDC.PCC model achieved the best hydraulic components' condition prediction performance with an average Hamming Score of 0.922. It can also be concluded that the OMDC models using binary relevance algorithms, except BR, outperform the label powerset algorithms. Thus, it can be said that the OMDC models using binary relevance algorithms can successfully predict the conditions of a hydraulic system's four components by considering the natural order of class labels.

**Table 6:** ZeroOne loss results of the OMDC models

| | | Traditional algorithms | | | | Ensemble-based algorithms | | |
|---|---|---|---|---|---|---|---|---|
| | | MLP | SVM | kNN | C.5 | Bagging | RF | AdaBoost |
| Binary relevance | OMDC.BR | 0.845 | 0.88 | 0.826 | 0.829 | 0.821 | 0.819 | 0.819 |
| | OMDC.CC | 0.435 | 0.674 | 0.17 | 0.142 | 0.073 | 0.043 | 0.042 |
| | OMDC.BCC | 0.442 | 0.76 | 0.168 | 0.147 | 0.073 | 0.044 | 0.042 |
| | OMDC.PCC | 0.409 | 0.669 | 0.167 | 0.14 | 0.073 | 0.041 | 0.041 |
| | OMDC.MCC | 0.41 | 0.669 | 0.168 | 0.142 | 0.073 | 0.041 | 0.042 |
| | OMDC.CDN | 0.582 | 0.677 | 0.329 | 0.166 | 0.19 | 0.213 | 0.043 |
| | OMDC.CT | 0.421 | 0.743 | 0.168 | 0.144 | 0.073 | 0.043 | 0.044 |
| | OMDC.CDT | 0.59 | 0.681 | 0.321 | 0.164 | 0.19 | 0.206 | 0.047 |
| Label powerset | OMDC.LP | 0.862 | 0.964 | 0.828 | 0.833 | 0.828 | 0.822 | 0.822 |
| | OMDC.PS | 0.863 | 0.964 | 0.829 | 0.833 | 0.828 | 0.822 | 0.823 |
| | OMDC.RAKEL | 0.863 | 0.964 | 0.829 | 0.833 | 0.828 | 0.822 | 0.823 |

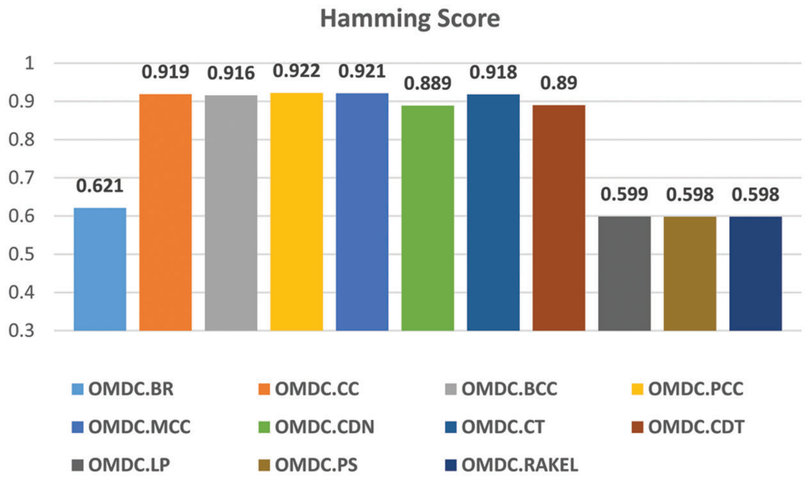

**Figure 2:** The average hamming score values of OMDC models

In addition to these results, the Hamming loss values of seven different base learners of the OMDC models were presented in Fig. 3. This graph demonstrates that the SVM algorithm presents the worst performance among the other base learners in each OMDC model with the high Hamming loss values. It is also understood from this figure that the AdaBoost algorithm has the lowest Hamming loss value, so this means that the AdaBoost outperforms the other base learners for each OMDC model. This is because of the fact that the AdaBoost is an ensemble learning algorithm and ensemble learners succeed more than traditional individual learners.

Finally, the average Hamming score values of the OMDC models using traditional and ensemble-based classification algorithms were calculated to show the effect of these algorithms on the prediction performances. The obtained values are presented in Fig. 4. When the results are considered in general, it is seen that the OMDC models that apply ensemble-based classification algorithms give more reliable prediction performances than the others that use traditional algorithms as base learners.

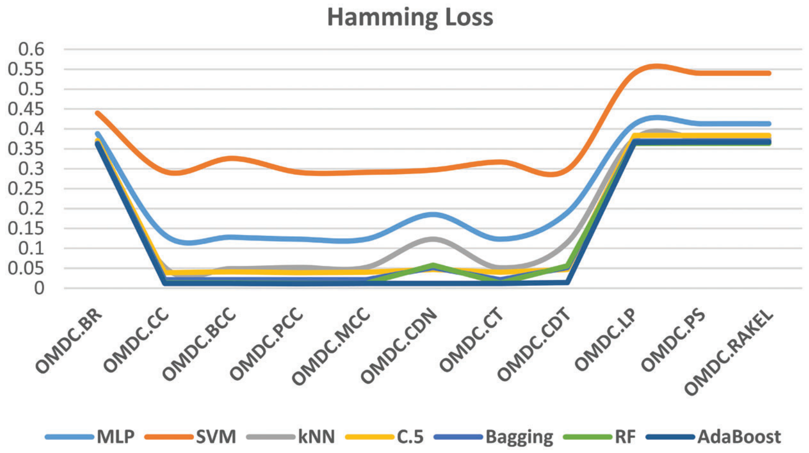

**Figure 3:** The average hamming loss values of base learners of the OMDC models

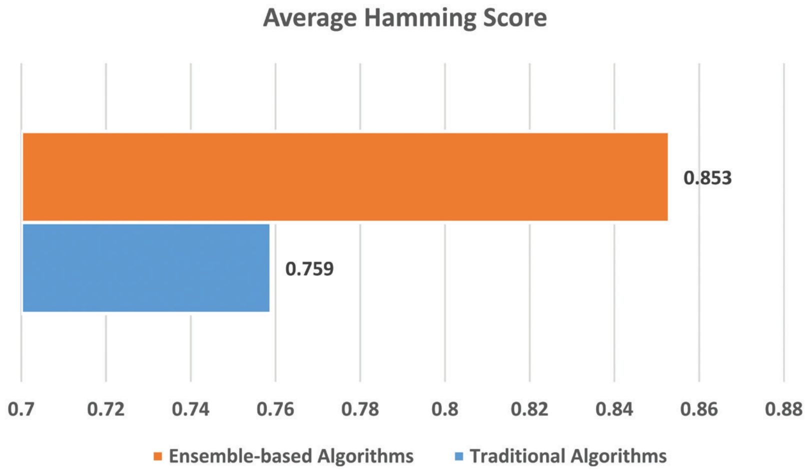

**Figure 4:** The average hamming score values of OMDC models using traditional and ensemble-based classification algorithms

## 5 Conclusion

PdM plays a vital role in maximizing productivity, product quality, and production agility of the manufacturing ecosystem. In recent years, machine learning methods have been preferred as an effective predictive method for maintaining industrial environments. The existing machine learning-based PdM studies ignore the inherent order relation between the target attribute values of the historical sensor data. To bridge the gap, this paper proposes an OMDC approach to predict the conditions of a hydraulic system's four components by combining MDC and OC paradigms to improve prediction performance. In the experiments, the OMDC approach was executed on a real-world dataset using the eleven different MDC algorithms with seven different base learners. The generated OMDC approaches were compared in terms of performance metrics. The results show that OMDC.PCC model using AdaBoost algorithm as a base learner provides the highest Hamming Score (0.989) and Exact Match (0.959) values among the other OMDC models. Also, it is clearly seen from the experimental results that the OMDC models using binary relevance algorithms present more accurate prediction performance than the others. Furthermore, the results indicate that the OMDC models that apply ensemble-based classification algorithms have higher prediction performances with an average Hamming score of 0.853 than those that use traditional algorithms as base learners. As a future study, the OMDC approach can be implemented using adaptive MDC algorithms on the same hydraulic system's condition dataset. In addition, a novel ordinal clustering method can be developed for clustering ordinal sensor data with the purpose of predictive maintenance.

**Funding Statement:** The author received no specific funding for this study.

**Conflicts of Interest:** The author has no conflicts of interest to report regarding the present study.

## References

[1] T. Carvalho, F. Soares, R. Vita, R. Francisco, J. Basto *et al.,* “A systematic literature review of machine learning methods applied to predictive maintenance,” *Computers & Industrial Engineering*, vol. 137, pp. 106024, 2019.

[2] M. Becherer, M. Zipperle and A. Karduck, “Intelligent choice of machine learning methods for predictive maintenance of intelligent machines,” *Computer Systems Science and Engineering*, vol. 35, no. 2, pp. 81–89, 2020.

[3] A. Massaro, S. Selicato and A. Galiano, “Predictive maintenance of bus fleet by intelligent smart electronic board implementing artificial intelligence,” *IoT*, vol. 1, no. 2, pp. 180–197, 2020.

[4] A. Buabeng, A. Simons, N. K. Frempong and Y. Y. Ziggah, “A novel hybrid predictive maintenance model based on clustering, smote and multi-layer perceptron neural network optimised with grey wolf algorithm,” *SN Applied Sciences*, vol. 3, no. 593, pp. 1–24, 2021.

[5] K. Kiangala and Z. Wang, “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*, vol. 8, pp. 121033–121049, 2020.

[6] W. Silva and M. Capretz, “Assets predictive maintenance using convolutional neural networks,” in *Proc. SNPD*, Toyama, Japan, pp. 59–66, 2019.

[7] Q. Wang and G. Wu, “Effective latent representation for prediction of remaining useful life,” *Computer Systems Science and Engineering*, vol. 36, no. 1, pp. 225–237, 2021.

[8] N. Kolokas, T. Vafeiadis, D. Ioannidis and D. Tzovaras, “Forecasting faults of industrial equipment using machine learning classifiers,” in *Proc. INISTA*, Sinaia, Romania, pp. 1–6, 2018.

[9] F. Vita and D. Bruneo, “On the use of LSTM networks for predictive maintenance in smart industries,” in *Proc. SMARTCOMP*, Washington, DC, USA, pp. 241–248, 2019.

[10] Q. Wang, S. Bu and Z. He, “Achieving predictive and proactive maintenance for high-speed railway power equipment with LSTM-RNN,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6509–6517, 2020.

[11] T. Praveenkumar, M. Saimurugan, P. Krishnakumar and K. I. Ramachandran, “Fault diagnosis of automobile gearbox based on machine learning techniques,” in *Proc. GCMM*, Vellore, India, pp. 2092–2098, 2014.

[12] G. A. Susto, S. McLoone, A. Schirru and S. Pampuri, “Prediction of integral type failures in semiconductor manufacturing through classification methods,” in *Proc. ETFA*, Cagliari, Italy, pp. 1–4, 2013.

[13] H. Wang, Z. Yu and L. Guo, “Real-time online fault diagnosis of rolling bearings based on KNN algorithm,” *Journal of Physics: Conference Series*, vol. 1486, no. 3, pp. 032019, 2020.

[14] M. Canizo, E. Onieva, A. Conde, S. Charramendieta and S. Trujillo, “Real-time predictive maintenance for wind turbines using big data frameworks,” in *Proc. ICPHM*, Dallas, TX, USA, pp. 1–8, 2017.

[15] A. Kusiak and Z. Zhang, “Optimization of power and its variability with an artificial immune network algorithm,” in *Proc. PSCE*, Phoenix, AZ, USA, pp. 1–8, 2011.

[16] R. Prytz, S. Nowaczyk, T. Rögnvaldsson and S. Byttner, “Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data,” *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 139–150, 2015.

[17] S. Ayvaz and K. Alpay, "Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time," *Expert Systems with Applications*, vol. 173, no. 114598, pp. 1–10, 2021.

[18] S. Arena, E. Florian, I. Zennaro, P. F. Orrù and F. Sgarbossa, "A novel decision support system for managing predictive maintenance strategies based on machine learning approaches," *Safety Science*, vol. 146, no. 105529, pp. 1–19, 2022.

[19] M. Last, A. Sinaiski and H. S. Subramania, "Predictive maintenance with multi-target classification models," in *Proc. ACIIDS*, Hue City, Vietnam, pp. 368–377, 2010.

[20] W. Zhang, D. K. Jha, E. Laftchiev and D. Nikovski, "Multi-label prediction in time series data using deep neural networks," *International Journal of Prognostics and Health Management*, vol. 10, no. 4, pp. 1–14, 2020.

[21] P. Campoy-Muñoz, P. Gutiérrez and C. Hervás-Martínez, "Addressing remitting behavior using an ordinal classification approach," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4752–4761, 2014.

[22] B. Nguyen, C. Morell and B. De Baets, "Distance metric learning for ordinal classification based on triplet constraints," *Knowledge-Based Systems*, vol. 142, pp. 17–28, 2018.

[23] T. Berghout, M. Benbouzid, S. M. Muyeen, T. Bentrcia and L. H. Mouss, "Auto-NAHL: A neural network approach for condition-based maintenance of complex industrial systems," *IEEE Access*, vol. 9, pp. 152829–152840, 2021.

[24] A. Mallak and M. Fathi, "Sensor and component fault detection and diagnosis for hydraulic machinery integrating lstm autoencoder detector and diagnostic classifiers," *Sensors*, vol. 21, no. 2, pp. 433–455, 2021.

[25] Y. Yoo, "Data-driven fault detection process using correlation based clustering," *Computers in Industry*, vol. 122, no. 103279, pp. 1–21, 2020.

[26] P. Yıldırım, U. Birant and D. Birant, "EBOC: Ensemble-based ordinal classification in transportation," *Journal of Advanced Transportation*, vol. 2019, pp. 1–17, 2019.

[27] E. Frank and M. Hall, "A simple approach to ordinal classification," in *Proc. ECML*, Freiburg, Germany, pp. 145–156, 2001.

[28] S. Raghu and N. Sriraam, "Optimal configuration of multilayer perceptron neural network classifier for recognition of intracranial epileptic seizures," *Expert Systems with Applications*, vol. 89, pp. 205–221, 2017.

[29] H. Wang, B. Zheng, S. Yoon and H. Ko, "A support vector machine-based ensemble algorithm for breast cancer diagnosis," *European Journal of Operational Research*, vol. 267, no. 2, pp. 687–699, 2018.

[30] S. Zhang, "Cost-sensitive KNN classification," *Neurocomputing*, vol. 391, pp. 234–242, 2020.

[31] X. Meng, P. Zhang, Y. Xu and H. Xie, "Construction of decision tree based on C4.5 algorithm for online voltage stability assessment," *International Journal of Electrical Power & Energy Systems*, vol. 118, pp. 105793, 2020.

[32] P. Yildirim and D. Birant, "Comparative analysis of ensemble learning methods for signal classification," in *Proc. SIU*, Izmir, Turkey, pp. 1–4, 2018.

[33] P. Yildirim and D. Birant, "The relative performance of deep learning and ensemble learning for textile object classification," in *Proc. UBMK*, Sarajevo, Bosnia and Herzegovina, pp. 22–26, 2018.

[34] Y. Wu, Y. Ke, Z. Chen, S. Liang, H. Zhao *et al.,* "Application of alternating decision tree with AdaBoost and bagging ensembles for landslide susceptibility mapping," *CATENA*, vol. 187, pp. 104396, 2020.

[35] B. Jia and M. Zhang, "Multi-dimensional classification via KNN feature augmentation," in *Proc. AAAI*, Honolulu, Hawaii, USA, pp. 3975–3982, 2019.

[36] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier chains for multi-label classification," in *Proc. ECML PKDD*, Bled, Slovenia, pp. 5782, 2009.

[37] J. Zaragoza, L. E. Sucar, E. F. Morales, C. Bielza and P. Larranaga, "Bayesian chain classifiers for multidimensional classification," in *Proc. IJCAI*, Barcelona, Spain, pp. 2192–2197, 2011.

[38] K. Dembczyinski, W. Cheng and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proc. ICML*, Haifa, Israel, pp. 1–8, 2010.

[39] J. Read, L. Martino and D. Luengo, "Efficient monte carlo optimization for multi-label classifier chains," in *Proc. ICASSP*, Vancouver, BC, Canada, pp. 3457–3461, 2013.

[40] Y. Guo and S. Gu, “Multi-label classification using conditional dependency networks,” in *Proc. IJCAI*, Barcelona, Spain, pp. 1300–1305, 2011.

[41] J. Read, L. Martino, P. Olmos and D. Luengo, “Scalable multi-output label prediction: From classifier chains to classifier trellises,” *Pattern Recognition*, vol. 48, no. 6, pp. 2096–2109, 2015.

[42] J. Read, B. Pfahringer, G. Holmes and E. Frank, “Classifier chains: A review and perspectives,” *Journal of Artificial Intelligence Research*, vol. 70, pp. 683–718, 2021.

[43] C. Kosemen and D. Birant, “Multi-label classification of line chart images using convolutional neural networks,” *SN Applied Sciences*, vol. 2, no. 1250, pp. 1–20, 2020.

[44] E. Cherman, M. Monard and J. Metz, “Multi-label problem transformation methods: A case study,” *CLEI Electronic Journal*, vol. 14, no. 1, pp. 1–10, 2011.

[45] J. Read, B. Pfahringer and G. Holmes, “Multi-label classification using ensembles of pruned sets,” in *Proc. ICDM*, Pisa, Italy, pp. 1–6, 2008.

[46] G. Tsoumakas and I. Vlahavas, “Random k-labelsets: An ensemble method for multilabel classification,” in *Proc. ECML*, Warsaw, Poland, pp. 406–417, 2007.

[47] R. Read, P. Reutemann, B. Pfahringer and G. Holmes, “Meka: A multi-label/multi-target extension to Weka,” *Journal of Machine Learning Research*, vol. 17, no. 21, pp. 1–5, 2016.

[48] N. Helwig, E. Pignalelli and A. Schütze, “Condition monitoring of a complex hydraulic system using multivariate statistics,” in *Proc. I2MTC*, Pisa, Italy, pp. 1–6, 2015.

[49] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, “A robust 3-D medical watermarking based on wavelet transform for data protection,” *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.

[50] X. R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, “Robust reversible audio watermarking scheme for telemedicine and privacy protection,” *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.