



## Research article

## Process monitoring for quality — A multiple classifier system for highly unbalanced data

Carlos A. Escobar<sup>a,\*</sup>, Daniela Macias<sup>b</sup>, Ruben Morales-Menendez<sup>b</sup><sup>a</sup> Global Research & Development, General Motors, Warren, MI, USA<sup>b</sup> Tecnológico de Monterrey, Monterrey, NL, Mexico

## ARTICLE INFO

## Keywords:

Defect detection  
 Binary classification  
 Manufacturing  
 Ensemble learning  
 Quality control  
 Machine learning  
 Unbalanced data  
 Multiple classifier system  
 Meta-learning

## ABSTRACT

In *big data*-based analyses, because of hyper-dimensional feature spaces, there has been no previous distinction between *machine learning algorithms (MLAs)*. Therefore, multiple diverse algorithms should be included in the analysis to develop an adequate model for detecting/recognizing patterns exhibited by classes. If multiple classifiers are developed, the next natural step is to determine whether the prediction benchmark set by the top performer can be improved by combining them. In this context, *multiple classifier systems (MCSs)* are powerful solutions for difficult pattern recognition problems because they usually outperform the best individual classifier, and their diversity tends to improve resilience and robustness to high-dimensional and noisy data. To design an *MCS*, an appropriate fusion method is required to optimally combine the individual classifiers and determine the final decision. *Process monitoring for quality* is a *Quality 4.0* initiative aimed at defect detection via binary classification. Because most mature organizations have merged traditional quality philosophies, their processes generate only a few defects per million of opportunities. Therefore, manufacturing data sets for binary classification of quality tends to be highly/ultra-unbalanced. Detecting these rare quality events is one of the most relevant intellectual challenges posed to the fourth industrial revolution, *Industry 4.0 (I 4.0)*. A new *MCS* aimed at analyzing these data structures is presented. It is based on eight well-known *MLAs*, an *ad hoc* fitness function, and a novel meta-learning algorithm. For predicting the final quality class, this algorithm considers the prediction from a set of classifiers as input and determines which classifiers are reliable and which are not. Finally, to demonstrate the superiority of the *MLAs* over extensively used fusion rules, multiple publicly available data sets are analyzed.

## 1. Introduction

The *fourth industrial revolution (I 4.0)* is changing the way we work, live, and interact with one another. It is founded on new technologies such as *industrial Big Data*, *Industrial Inter-net of Things (IIoT)*, and *Artificial Intelligence (AI)*, and it fuses physical systems with virtual ones, thus enabling a smart and connected world. *I 4.0* is impacting all scientific disciplines, industries, and economies. A simulation study demonstrated that by 2030, 70% of companies might adopt at least one type of *AI* technology and potentially generate an additional economic activity of \$13 trillion USD [1]. However, new business models such as *Amazon*, *Facebook*, and *Uber* have recently emerged and are propelling the *I 4.0*. Manufacturing science has been innovating, advancing, and evolving after the beginning of the first industrial revolution [2]. At present, manufacturing is the driving economic force of the most advanced countries [3]. Top-ranking nations in overall manufacturing environment [4] have the highest *gross domestic product (GDP)* [5]. Although not a

trivial task, the driving technologies in *I 4.0* have the capacity to further the *state-of-the-art* of manufacturing science.

From the perspective of manufacturing quality, most mature organizations have merged traditional quality control methods to create high-conformance production environments. Process monitoring charts have been developed to improve the process capability index at the industrial benchmark of *sigma level four* [6, 7]. This sigma level generates 6,210 *defects per million of opportunities (DPMOs)* [8, 9]. Detecting these defects to move manufacturing processes to the next *sigma level* is one of the primary intellectual challenges posed to *AI* [10] (see Figure 1).

*Process monitoring for quality (PMQ)* is a *Quality 4.0*<sup>1</sup> initiative founded on *AI* (Figure 2). Detecting rare quality events or few *DPMOs* generated

<sup>1</sup> *Quality 4.0* is the fourth wave in the quality movement (1. *Statistical Quality Control*, 2. *Total Quality Management*, 3. *Six Sigma*, 4. *Quality 4.0*). Its quality philosophy is built on the statistical and managerial foundations of the previous philosophies. It leverages *IBD* and *AI* to solve an entirely new range of intractable engineering problems. *Quality 4.0* is founded on a new paradigm based on empirical learning, empirical knowledge discovery, and real-time data generation, collection, and analysis to enable smart decisions [13].

\* Corresponding author.

E-mail address: [carlos.1.escobar@gm.com](mailto:carlos.1.escobar@gm.com) (C.A. Escobar).

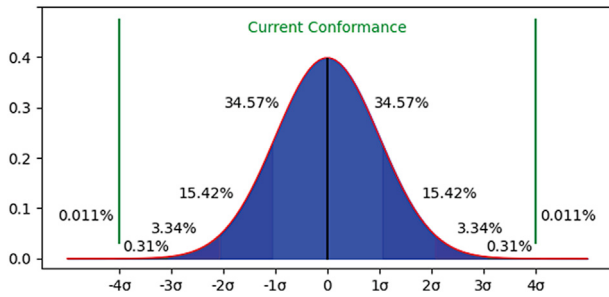


Figure 1. Current conformance rate across manufacturing companies.



Figure 2. PMQ in the context of Industry 4.0.

by a typical manufacturing process is one of the primary goals. Defect detection is formulated as a binary classification problem (good or defective) [11]. PMQ has evolved the traditional quality problem solving strategies (PDCA, DMAIC, IDDOV [12]) into a seven-step approach —(Identify, Acensorize, Discover, Learn, Predict, Redesign, Relearn: IADLPR<sup>2</sup>) — to effectively solve pattern classification problems and to guide (Figure 3) [12,13]. An approach to solve the *predict step* (or letter P) of the problem solving strategy is presented.

From the ML perspective, manufacturing-derived data sets for binary classification of quality tend to be highly/ultra-unbalanced (minority class count <1%). Therefore, a learning strategy (e.g., hyper-parameter tuning, classification threshold definition, and misclassification cost) must be developed to address this situation. Owing to the limited amount of information available of the minority class, it is extremely complicated for an MLA to capture this pattern.

In big data-based analyses, because of hyper-dimensional feature spaces, the data structure is not known in advance. There is, therefore, no *a priori* distinction between MLAs [14]. Multiple MLAs should be included in the analysis to develop an adequate model for the pattern exhibited by the classes. Moreover, based on empirical evidence, diversity tends to improve the prediction performance, resilience, and robustness of high-dimensional and noisy data [15].

If multiple classifiers are developed, the next natural step is to combine them to optimize prediction. The most common fusion rules are majority, simple majority, and unanimous voting [15]. Recently, a majority voting, *Multiple Classifier System (MCS)* concerning biological activities was developed [16]; this predictive system outperformed individual classifiers by addressing the over-fitting problem. However, in an MLA, the concept of drift [17, 18] addresses the fact that the statistical distributions of classes change over time in an unforeseen manner. This poses important technical and practical challenges because a stationary relationship is assumed between features and classes. Assumption is rarely held by manufacturing systems [13]. Moreover, the pitfall of using predefined classification rules was acknowledged in Ref. [19].

If extensively used or existing static *fusion rules* rarely generalize/sustain in manufacturing, the research question addressed in this study is: which *fusion rule* should be used to optimize the detection of rare events? A novel *meta-learning algorithm* [20] was developed to answer this question.

The process of developing an MCS is divided into two optimization goals: (1) coverage optimization, an approach aimed at increasing the hyper-dimensional space covered by a set of mutually complementary classifiers, and (2) decision optimization, a *meta-learning algorithm* aimed at designing an appropriate decision combination scheme (i.e., fuser) over a set of previously trained classifiers [21, 22].

A prediction optimization approach, PMQ-O, is presented. It is an effective strategy aimed at developing an MCS with the capacity to analyze highly/ultra-unbalanced data. The proposed approach is based on the following: (1) a list of eight diverse MLAs, (2) an *ad hoc* fitness function, and (3) a new *meta-learning algorithm* that searches for an optimal solution. These three components address the two optimization

goals. As demonstrated by multiple empirical studies, PMQ-O is a step forward in moving manufacturing processes to the next sigma level.

The rest of this study is organized as follows. A general scheme of PMQ is presented in Section 2. A brief theoretical background of this study is presented in Section 3. An optimizer is presented in Section 4. Empirical studies are outlined in Section 5, wherein a virtual case study is presented, aimed at explaining the optimizer in a step-by-step manner, followed by the analysis of six real data sets. Industry 4.0 technologies and sustainability are mentioned in Section 6. Finally, Section 7 concludes this paper. Table 1 summarizes the acronym definitions.

## 2. Process monitoring for quality

In PMQ, the *Big Models (BM)* learning paradigm [23] is applied to process data to develop a classifier aimed at defect detection. Using the notation of eqn. (1), a positive label refers to a defective item, whereas a negative one refers to a good quality item.

$$Quality = \begin{cases} 1 & \text{if } i^{\text{th}} \text{ item is defective}(+) \\ 0 & \text{if } i^{\text{th}} \text{ item is good}(-) \end{cases} \quad (1)$$

The predictive performance of a classifier is summarized using a *confusion matrix (CM)* (see Table 2).

Because predictions are performed under uncertainty, a classifier can commit FP (type-I,  $\alpha$ ) and FN (type-II,  $\beta$ ) errors [24]. In the context of binary classification of quality, an FP error occurs when a classifier labels a good item as a defective one, whereas an FN error occurs when a defective item is labeled as good. Errors are computed by the following equations:

$$\alpha = \frac{FP}{FP + TN} \quad (2)$$

$$\beta = \frac{FN}{FN + TP} \quad (3)$$

Figure 4 shows a typical high-conformance process controlled by PMQ where observational (i.e., empirical) data are used to train a classifier following the BM learning paradigm [23]. From a theoretical perspective, applying ML to detect the minority class is the primary challenge, whereas from practical/business perspectives, the goal is to develop defect-free processes [10]. Because an FN can be reevaluated at a minimum cost and continue in the value-adding process, the classifier must exhibit a predominantly high detection ability ( $\beta \approx 0$ ) and the smallest possible  $\alpha$  error.

PMQ uses a seven-step problem solving strategy to guide innovation. IADLPR<sup>2</sup> is developed based on theory and our knowledge of complex manufacturing systems. As per empirical results, this strategy increases

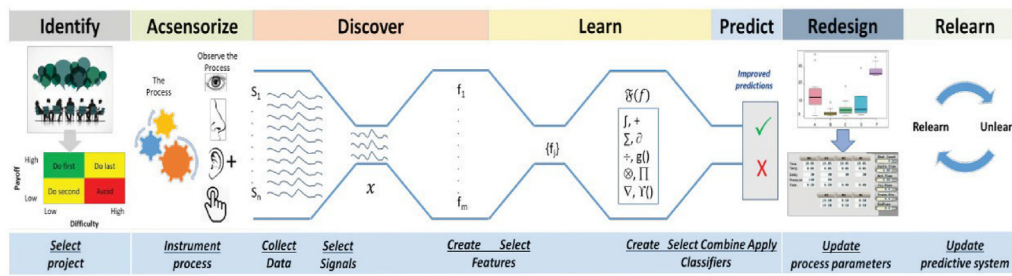


Figure 3. PMQ problem solving strategy.

Table 1. Acronym definitions.

Acronym	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BM	Big Models
CM	Confusion Matrix
DPMO	Defects Per Million of Opportunities
FN	False Negative
FP	False Positive
GDP	Gross Domestic Product
IADLPR <sup>2</sup>	Identify, Acensorize, Discover, Learn, Predict, Redesign, Relearn
IIoT	Industrial Internet of Things
I4.0	Industry 4.0
KNN	K-Nearest Neighbors
MCS	Multiple Classifier System
LR	Linear Regression
ML	Machine Learning
MLA	Machine Learning Algorithm
MPCD	Maximum Probability of Correct Decision
NB	Naive Bayes
PMQ	Process Monitoring for Quality
PMQ-O	Prediction Optimization approach
RF	Random Forest
RBF	Radial Basis Fusion
QMC	Quality Monitoring and Control
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

Table 2. Confusion matrix.

	Predicted good	Predicted bad
Good item	True Negative (TN)	False Positive (FP)
Defective item	False Negative (FN)	True Positive (TP)

the likelihood of success by addressing the primary challenges of manufacturing systems [11,13,23]. Table 3 lists the primary goals for each step of this strategy.

### 3. Theoretical background

The overall research goal of AI is to create technologies that augment human intelligence or take over risky jobs not appropriate for humans. ML, robotics, computer vision, natural language processing, and expert systems are the common AI areas [25].

ML serves as a tool for information extraction, data pattern recognition, and prediction. Although it becomes increasingly challenging to build first-principle models in these increasingly complex processes, data-driven process modeling, monitoring, prognosis, and control have recently received considerable attention. Manufacturing must be reimagined in the new era of data science and information technology.

The primary goal of data mining is to extract beneficial details and transfer them to practical knowledge to improve process decision support. Multiple tasks require pattern recognition, reasoning, and decision-making under complex conditions. Furthermore, they often deal with *ill-defined problems*, noisy data, model uncertainties, combinatorially large search spaces, non-linearities, and the requirement for speedy solutions. Such features are reported in many issues in the process industry—in synthesis, design, control, scheduling, optimization, and risk management.

There are three different types of MLA: *unsupervised methods*, *supervised learning methods (SVMs)*, and *reinforcement learning methods*. Supervised and unsupervised learning methods account for 80%–90% of all industrial applications, [26]. With I 4.0, IIoT, and *cloud computing*, MLA-based *quality monitoring and control (QMC)* has successfully resurfaced in various industrial domains. For example, real-time data acquisition systems enable an MLA application to accurately assess the quality of a manufacturing process.

A feedback (*adaptive and self-updating*) system with sufficient data is essential to ensure the effectiveness of an MLA over time, which enables the real-time improvement in quality inspection. Furthermore, an MLA can be used to devise complex models and algorithms that can lend themselves to prediction. These algorithms allow for people to produce reliable, repeat-able decisions and uncover hidden insights by learning from historical relationships and trends in the data.

An algorithm integrating feedback and an adaptable process that is analyzed over time in the long run to improve SVM-based systems was proposed in [27]. The adaptive approach considerably improved the feedback process and effectively enhanced the system accuracy and reliability over time. The framework to pre-process the input data for SVM-based decision-making algorithms was applicable for MLAs, e.g., cost-effective SVM-based automated QMC. It incorporated inspection-related expenses (warranty cost, rework cost, inspection cost) and error types (type I and II errors) in the algorithm [28]. In this proposal, the quality check reflected the company priorities regarding cost-saving policy versus traditional SVM methods that employ the lowest inspection error rate criterion for classification. Essential features were considered the ability to learn from and make predictions based on defective parts.

In addition to these SVM-based applications, there are many more that combine two MLAs to generalize the practical approach: MCSs. To develop an MCS, three questions must be answered: (1) which classifiers should be included?, (2) which fitness function should be optimized?, and (3) how should the predictions (labels) of the classifiers be combined?

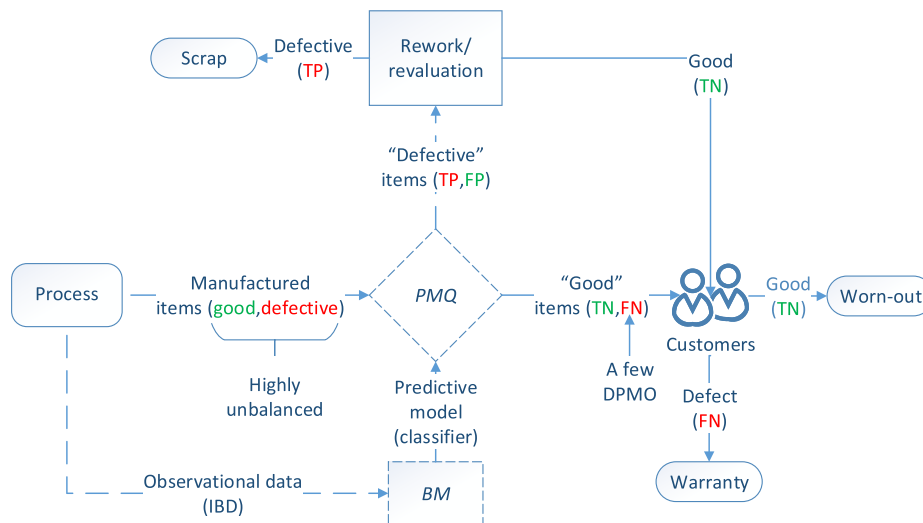


Figure 4. PMQ scheme.

Table 3. Seven steps and goals.

Step	Goal
Identify	to develop a prioritized portfolio of projects with a high business impact and likelihood of success.
Acsensorize	to observe the process and generate the raw empirical data to monitor the system to create features
Discover Learn	to develop the classifier using the <i>Big Models</i> learning paradigm
Predict	to optimize prediction (this paper goal)
Redesign	to derive engineering knowledge from the data mining results
Relearn	to develop a relearning strategy for classifier to learn new statistical distributions classes

An overview of *aggregation methods* and *meta-learning* in *ML* is presented. Thereafter, a brief theoretical review is provided to answer the first two questions, and the third one is answered in Section 4.

### 3.1. Aggregation methods

For an *ensemble learning* approach (voting by several classifiers), a group of homogeneous classifiers is called an *ensemble* [29], e.g., bagging and boosting [29, 30, 31]. However, a group of heterogeneous classifiers is called an *MCS* or *non-ensemble* [32, 33]. In the context of ideas expressed in this study, an *MCS* is a predictive approach that may include a combination of classifiers with *ensembles*.

*MCSs* are a powerful solution to difficult pattern recognition problems because they usually outperform the best individual classifiers [34]. This improvement has been analytically proven under certain conditions (e.g., majority voting by a group of independent classifiers) [35].

To design an *MCS*, an appropriate *fusion* method is required to optimally combine the individual classifier outputs to determine the final decision (classification). Heterogeneous or homogeneous modeling backgrounds can be integrated (e.g., *LR*, *SVM*, and *random forest (RF)*) to exploit the strengths of each individual classifier and to overcome the limitations of an optimal local solution developed by an individual classifier. More-over, high diversity helps by decreasing the classifier output correlation [36, 37] and providing better options to explore different decision combination schemes. The most common *fusion rules* are majority, simple majority, and unanimous voting [15].

*Meta-learning* is a broad topic in *AI* [20, 38]. This paradigm refers to an *MLA* that learns from the output of another *MLA* [39, 40]. A *meta-learning* algorithm is applied to dynamically learn or adjust the *fusion rule* of the *MCS* on the basis of the characteristics of the data at hand. It replaces static *fusion rules* or human involvement to manually evaluate different *fusion rules*. Figure 5 describes recent developments of *meta-learning* algorithms [41, 42, 43, 44]. A brief comment is included for each of them to briefly describe their diverse agendas.

### 3.2. List of MLA

To address the coverage optimization problem [22], a list of eight diverse and complementary *MLAs* is considered: *LR*, *SVM* [45], including *radial basis function (RBF)* kernels [46], *Naive Bayes (NB)* [47], *k-nearest neighbors (KNN)* [48], *artificial neural network (ANN)*<sup>2</sup> [49], *RF* [31], and *random undersampling boosting (RUS-Boost)* algorithms (see Table 4). The proposed list includes margin- and probability-based, linear and non-linear [49], parametric and non-parametric [50], stable and unstable [51], and generative and discriminative [52] algorithms<sup>3</sup>. These approaches can be used to solve a wide spectrum of binary classification problems.

The first seven *MLAs* were initially proposed in [23]. They include the *RF* algorithm, a bagging approach. The original list is complemented with a boosting method. Because rare quality event detection is one of the primary applications of *PMQ* [10], *RUSBoost* [53] was selected as a boosting algorithm. *RUSBoost* is a combination of *random undersampling* and *AdaBoost* [54]. *Random undersampling* is applied to the majority class to balance the ratio between minority and majority classes, following which *AdaBoost* is applied to the balanced-subset to build a model.

The primary causes of error (i.e., misclassifications) are noise, bias, and variance. *Ensemble learning* (e.g., bagging and boosting) tends to produce a more reliable classification than a single classifier, and therefore minimizes these sources of errors. Although these methods are designed to improve the stability and robustness to noise, they have slightly different agendas for solving the bias-variance tradeoff [55]. In general, boosting tends to reduce the bias problem. Bagging may solve the over-fitting (variance) problem while boosting can increase it. Including both types of *ensembles* may be a good idea to effectively solve

<sup>2</sup> *ANN* can be shallow or deep, depending on the available type of data.

<sup>3</sup> The authors acknowledge that some algorithms can change their taxonomy (e.g., from parametric to non-parametric) depending upon their definition.

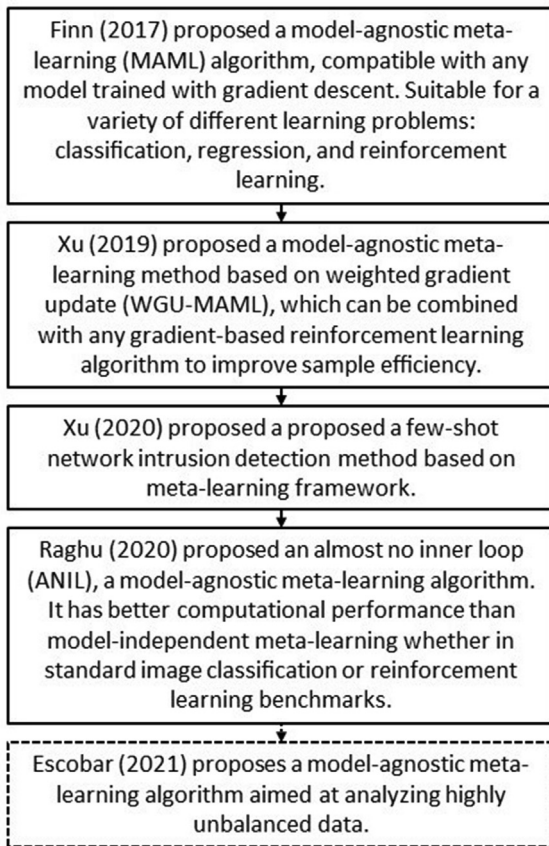


Figure 5. Recent developments of meta-learning algorithms.

the bias–variance trade-off. Once a set of classifiers has been defined and trained, the next natural step is to rank them on the basis of a fitness function.

### 3.3. Maximum Probability of Correct Decision (MPCD)

*Maximum Probability of Correct Decision (MPCD)* is the fitness function to be optimized. It is a probability-based measure of classification performance, based on the  $\alpha$  and  $\beta$  errors. It effectively analyzes highly/ultra-unbalanced data structures [11, 56] because its score essentially describes the ability of the classifier to detect the minority class, e.g., defective items. One has

$$MPCD = (1 - \alpha)(1 - \beta) \in [0, 1] \quad (4)$$

where a higher score indicates better classification performance.  $MPCD = 1$  describes the perfect separation of classes, whereas  $MPCD = 0$  describes either an  $\alpha = 1$  (all good called bad) or  $\beta = 1$  (all bad called good)<sup>4</sup>. Once the classifiers have been ranked and the top performer identified, the next intuitive challenge is to determine whether two or more classifiers can be combined to improve the predictive benchmark of the top performer.

## 4. Classification optimization

A meta-learning algorithm (*meta-classifier*<sup>5</sup>) is presented. Its primary goal is to determine the *optimal fusion scheme* for PMQ-O. It generates a

<sup>4</sup> Since MPCD is the fitness function, it is recommended to tune the hyper-parameters of the MLA with respect to this metric too.

<sup>5</sup> Meta-learning algorithm for classification tasks.

search space based on the set of classifiers and determines the *optimal fusion rule*.

Because manufacturing systems tend to be time dependent, the data set should be split after a time-ordered hold-out validation scheme (training and testing sets). The eight MLAs described in Table 4 are applied to the training set (e.g., first 70% samples) to create a set of classifiers. Thereafter, they are applied to the testing set (e.g., remaining 30 % samples). The *Real Labels (RLs)*, *Predicted Labels (PLs)*, and classifier list are the inputs of the optimizer that searches for a better solution (i.e., optimized prediction performance) with two possible outcomes: (1) top performer, a single classifier, or an *ensemble (RF or RUSBoost)*; this situation occurs when no classifier fusion surpasses the benchmark set by the top performer (see Figures 6 and 2) *MCS*, a set of classifiers (it may include only a few or all) with a *fusion rule* that improves the benchmark set by the top performer (see Figure 7). The top performer may or may not be included in the *MCS*.

### 4.1. Optimizer pseudo-code

The PMQ-O algorithm has three components (see Figure 8 and Table 5). It performs  $n$  iterations (lines 1–20), in each of which  $C_k^n$  number of combinations are generated, where different combinations created at each iteration (line 2). Once the combinations have been defined, the algorithm performs  $C$  iterations to determine the best one ( $\text{argmax}\{MPCD\}$ ) at each value of  $k$  (lines 3–16). To determine the best combination, a new vector is created, *SumLMCS* (see Eqn. 5), in which the values of the labels of the classifiers included in the combination are summed up (line 4). Thereafter, different fusion rules ( $r$ ) are explored for each combination, starting with zero up to  $k - 1$ . The number of different *fusion rules* to be explored is equal to  $k$  (lines 5–13). To evaluate each *fusion rule*, the final *PL*, *LMCS* (see Eqn. 6), of the combination (lines 6–10) is used to compute the associated *MPCD* (line 11).

The *PredictionCapacityList* is used to store the associated *MPCD* to each different value of  $r$  for the combination under analysis (line 12). *PredictionCapacityListNC* and *FusionRuleListNC* lists are used to store the best *fusion rule*  $r$  of all the combinations developed at each value of  $k$  (lines 14 and 15). *PredictionCapacityListK*, *FusionRuleListK*, and *SelectedCombinationListK* lists are used to store the information (*MPCD*,  $r$ , *combination index -nc-* respectively) of the best combination at each value of  $k$  (lines 17–19). The final combination, *PredCapacityFinal*, is identified on the basis of the *MPCD* value (line 21). Because the dimensions of *Com* change at every iteration, the final combination must be generated again. At this point, all the information is available and is collected in  $K$  (line 22). *SelectedCombinationFinal* identifies the combination index (line 23) and the associated *fusion rule* in *FusionRuleFinal* (line 24). Using the values of  $n$  and  $K$ , the set of combinations in  $C^n$  are generated (line 25), and the classifiers included in the index *SelectedCombinationFinal* are included in the *MCS* (line 26). Finally, the solution is reported (see Eqn. 7); *mcs*: the number of classifiers, *SelectedClassifiersFinal*, *FusionRuleFinal*, and *PredCapacityFinal* (lines 27–31).

$$SumPLMCS = \sum L_{nc} \quad (5)$$

$$PLMCS_{nci} = \begin{cases} 1 & \text{if } SumPLMCS_{nci} > r \Rightarrow i^{\text{th}} \text{item Pred. bad (+)} \\ 0 & \text{if } SumPLMCS_{nci} \leq r \Rightarrow i^{\text{th}} \text{item Pred. good (-)} \end{cases} \quad (6)$$

$$MCS = \text{argmax}_{nkncr} \{MPCD\} \quad (7)$$

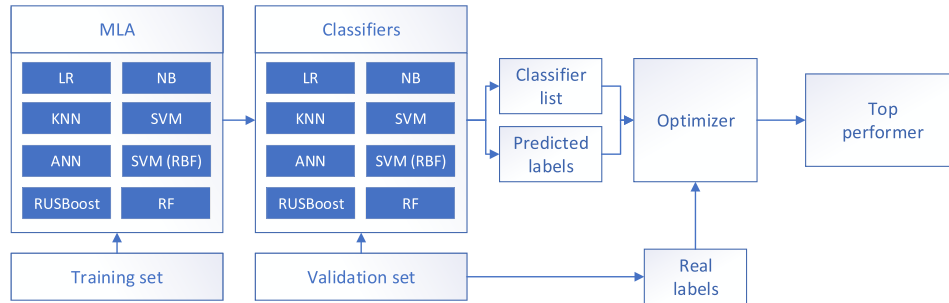
The algorithm determines an optimal fusion scheme (i.e., number of classifiers, and fusion rule) for the *MCS*, where the search space is defined by the following:

$$\sum_{k=1}^n k * C_k^n \quad (8)$$

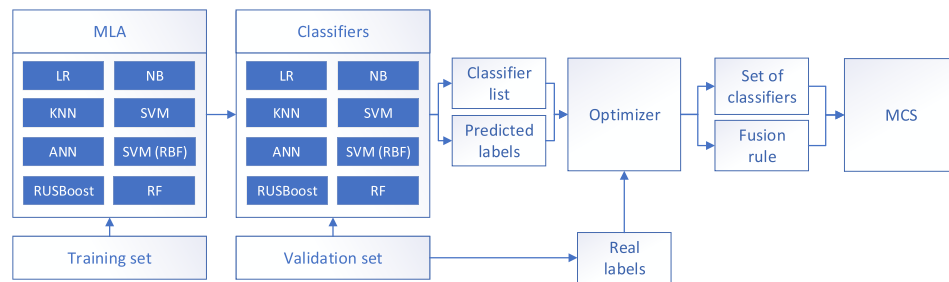
**Table 4.** Characteristics of the *MLA*.

Index	<i>MLA</i>	Linear	Nonlinear	Parametric	Nonparametric	Stable	Unstable	Gen	Dis
1	SVM	✓		✓		✓			✓
2	LR	✓		✓		✓			✓
3	NB		✓*	✓		✓		✓	
4	KNN		✓		✓	✓			✓
5	ANN		✓	✓**			✓		✓
6	SVM(RBF)		✓	✓	✓	✓			✓
7	RF		✓		✓	✓			✓
8	RUSBoost		✓		✓	✓			✓

\*with numeric features, \*\*with a set of parameters of fixed size. Gen: Generative, Dis: Discriminative.



**Figure 6.** Top performer.



**Figure 7.** Multiple classifier system.

```

1. for k = 1 to n, the total number of classifiers in Class, do
2.   generate Com, the different combinations in Ck
3.   for nc = 1 to Ck, the total number of combinations in Com, do
4.     compute SumPLMCSnc, a vector with the sum of the predicted labels of the classifiers in the combination
5.     for r = 0 to k-1, the number of fusion rules to be explored
6.       if SumPLMCSnc > r, the fusion rule
7.         PLMCSnc,r = 1 (or positive)
8.       elseif
9.         PLMCSnc,r = 0 (negative)
10.      end
11.     compute the MPCDnc by comparing PLMCSnc with RL
12.     PredictionCapacityListR(nc) = MPCDnc
13.   end
14. PredictionCapacityListNC(nc) = max(PredictionCapacityListR)
15. FusionRuleListNC(nc) = position(max(PredictionCapacityListR)) - 1
16. end
17. PredictionCapacityListK(k) = max(PredictionCapacityListNC)
18. FusionRuleListK(k) = FusionRuleListNC(position(max(PredictionCapacityListNC)))
19. SelectedCombinationListK(k) = position(max(PredictionCapacityListK))
20. end
21. PredCapacityFinal = max(PredictionCapacityListK)
22. K = position(max(PredictionCapacityListK))
23. SelectedCombinationFinal = SelectedCombinationListK(K)
24. FusionRuleFinal = FusionRuleListK(K)
25. generate the different combinations in ComFinal = Ck
26. SelectedClassifiersFinal = ComFinal(SelectedCombinationFinal, 1:end), select the final combination
27. return MCS
28. mcs = number of elements in SelectedClassifiers
29. SelectedClassifiersFinal, set of classifiers in the MCS
30. FusionRuleFinal, fusion rule
31. PredCapacityFinal, prediction capacity of the MCS
    
```

**Figure 8.** Optimizer pseudo-code.

## 5. Empirical studies

To gain insights into how the algorithm searches for *fusion rules* aimed at prediction optimization, a virtual case is validated.

### 5.1. Virtual case study

A sample size of 10 ( $m = 10$ ) with three bad(s) and seven good(s) is generated. The prediction of three classifiers ( $n = 3$ ) is input, and then

the optimizer is applied to search for a *fusion rule* that enhances the prediction ability of the top performer. The *RLs* and *PLs* of each classifier,  $Class(C_1, \dots, C_3)$ , are presented in Table 6, and the prediction performance of each classifier is detailed in Table 7.

Tables 6 and 7 present the results of the different combinations in  $C_3^3$ . Because there is only one classifier in each combination ( $k = 1$ ),  $r = 0$  is the only *fusion rule* to be evaluated, and  $PL = SumPLMCS = PLMCS$ . According to the prediction results,  $C_1$  is the top performer with  $MPCD = 0.7142$ .

**Table 5.** Optimizer pseudo-code.

<b>(1) Inputs</b>			
A set of classifiers and their associated predicted labels			
<ul style="list-style-type: none"> <li>Class(<math>C_1, C_2, \dots, C_n</math>), a list of classifiers, <math>n</math> represents the number of classifiers</li> <li><math>PL(pl_{ij})_{m \times n}</math>, a matrix with predicted labels (<math>pl_{ij} \in \{0, 1\}</math>) of the samples in the validation set, where <math>m</math> denotes the sample size, and the subscripts <math>i, j</math> are used to denote the <math>i^{th}</math> sample of the <math>j^{th}</math> classifier</li> <li><math>RL(r_l)_{m_s}</math>, a vector of size <math>m</math> with real labels (<math>r_l \in \{0, 1\}</math>)</li> </ul>			
<b>(2) Outputs</b>			
A Multiple Classifier System (MCS)			
<ul style="list-style-type: none"> <li>mcs the number of classifiers in the MCS</li> <li>SelectedClassifiersFinal, set of classifiers (<math>C_1, \dots, C_{mcs}</math>) in the MCS (it may include only the top performer)</li> <li>FusionRuleFinal, the fusion rule (zero is the fusion rule if only the top performer is included)</li> <li>PredCapacityFinal, estimated prediction ability</li> </ul>			
<b>(3) Initialization</b>			
Define the algorithm's lists.			
<ul style="list-style-type: none"> <li>Set <math>Com</math> as empty</li> <li>Set <math>SumLMCS</math> as empty</li> <li>Set <math>LMCS</math> as empty</li> <li>Set <math>MPCD</math> as empty</li> <li>Set <math>PredictionCapacityListR</math> as empty</li> <li>Set <math>PredictionCapacityListNC</math> as empty</li> <li>Set <math>FusionRuleListNC</math> as empty</li> <li>Set <math>PredictionCapacityListK</math> as empty</li> <li>Set <math>FusionRuleListK</math> as empty</li> <li>Set <math>SelectedCombinationListK</math> as empty</li> </ul>			

Now, the different combinations generated with  $k = 2$  ( $C_2^3$ ) are evaluated with respect to  $r = 0$  and  $1$ , where Eq. (5) is used to populate Table 8 (sumPLMCS). The classification rule described in Eq. (6) is then applied to define the PLs of each combination (PLMCS) (see Tables 9 and 10). Finally, Tables 11 and 12 present their associated predictive performances.

As per Table 11, the combination of classifiers  $C_2$  and  $C_3$  with a fusion rule  $r = 0$  perfectly separates the data ( $MPCD = 1$ ) and surpasses the benchmark set by classifier  $C_1$  ( $MPCD = 0.7142$ ). Finally,  $k = 3$  is evaluated. Because it is just one combination with the three classifiers, three rules are assessed,  $r = 0, 1$ , and  $2$ . Following the same structure, the results are presented in Tables 13, 14, 15, 16, 17, 18, and 19.

As per Table 17, the combination of classifiers  $C_1, C_2$ , and  $C_3$  with a fusion rule  $r = 1$  perfectly separates the data ( $MPCD = 1$ ). Finally, a comparative analysis combining the three classifiers with the three most common fusion rules (i.e., majority, unanimous-zero, and unanimous-one) and PLs are presented in Table 20. The classification performance is summarized in Table 21.

**Table 6.** Predicted labels of each classifier vs real labels,  $k = 1$ .

Sample	$PL_{C1}$	$PL_{C2}$	$PL_{C3}$	RL
1	1	0	0	0
2	1	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	1	1	0	1
9	1	0	1	1
10	1	0	1	1

**Table 7.** Prediction performance ( $k = 1, r = 0$ ).

$r = 0$	$C_1$	$C_2$	$C_3$
$\alpha$	0.2857	0	0
$\beta$	0	0.6667	0.3333
MPCD	0.7142	0.3333	0.6667

According to Table 21, although the majority vote resulted in perfect separation, this is not always the case [23]. However, as shown in this study, unanimous-zero (i.e.,  $r = 0$ ) tends to be a good rule for rare event detection when the data set is highly/ultra-unbalanced. This fusion rule, however, tends to increase the  $\alpha$  error. The unanimous-one rule (i.e.,  $r = n$ ) is on the other extreme; it tends to fail to detect or significantly increase the  $\beta$  error.

Static fusion rules, such as majority or unanimous voting, often times do not determine the optimal manner to combine the decisions of the classifiers because they do not eliminate highly correlated or spurious classifiers [36, 37] that degrade the generalization performance. However, the meta-learning algorithm presented only includes classifiers that optimize prediction. This is demonstrated in the following subsection.

### 5.2. Real case studies

To exhibit the performance of PMQ-O, six highly/ultra-unbalanced data sets (five publicly available) were analyzed using the eight MLAs (see Table 4). The name and relevant information for each data set are presented in Table 22, [23, 57]. The prediction performance on the testing set of each classifier is summarized in Tables 23 and 24 presents the information of the MCS.

In data set #1, the top performer is the LR algorithm with an estimated  $MPCD = 0.8799$ ,  $\alpha = 0.0101$ , and  $\beta = 0.1111$  (see Table 23); this benchmark is improved ( $MPCD = 0.8821$ ) if LR is combined with the ANN with a fusion rule of 1. In this case, although the detection ability ( $\beta$ -error) is not improved, the FP ( $\alpha$ -error) is reduced. In data set # 2, the benchmark set by the KNN ( $MPCD = 0.9736$ ) is higher ( $MPCD = 0.9799$ ) by the MCSs based on SVM, SVM(RBF), ANN, and KNN algorithms with a fusion rule of 1. In this scenario, the detection ability is improved by 1.74% (from  $\beta = 0.0239$  to  $\beta = 0.0065$ ) with a 1.1% loss of  $\alpha$  (from 0.0026 to 0.0136). In data sets #3 and #4, the benchmarks set by RUSTBoost and ANN are not improved by any decision combination. In data set #5, the benchmark set by SVM(RBF) ( $MPCD = 0.9901$ ) is improved ( $MPCD = 0.9966$ ) if the predictions of the KNN and RF are aggregated with a fusion rule of 1. In this case, detection is slightly improved, whereas the  $\alpha$ -error remains essentially the same. Finally, in data set #6, the benchmark set by the SVM ( $MPCD = 0.8704$ ) is improved ( $MPCD = 0.8760$ ) by an MCS based on the SVM LR and ANN with a fusion rule of 1. In this case, the  $\alpha$ -error increases by 0.37% (from 0.0399 to 0.0436), whereas the  $\beta$ -error decreases by 0.94% (from 0.0935 to 0.0841).

**Table 8.** SumPLMCS, sum of the values of the combinations created with  $k = 2$ .

Sample	$\sum PL_{C1-C2}$	$\sum PL_{C1-C3}$	$\sum PL_{C2-C3}$	RL
1	1	1	0	0
2	1	1	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	2	1	1	1
9	1	2	1	1
10	1	2	1	1

**Table 9.** PLMCS, predicted labels with  $r = 0$ .

Sample	$\sum PLC_1 - C_2$	$\sum PLC_1 - C_3$	$\sum PLC_2 - C_3$	RL
1	1	1	0	0
2	1	1	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	1	1	1	1
9	1	1	1	1
10	1	1	1	1

**Table 10.** PLMCS, predicted labels with  $r = 1$ .

Sample	$\sum PLC_1 - C_2$	$\sum PLC_1 - C_3$	$\sum PLC_2 - C_3$	RL
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	1	0	0	1
9	0	1	0	1
10	0	1	0	1

**Table 11.** Prediction performance ( $k = 2, r = 0$ ).

$r = 0$	$C_1 - C_2$	$C_1 - C_3$	$C_2 - C_3$
$\alpha$	0.2857	0.2857	0
$\beta$	0	0	0
MPCD	0.7142	0.7142	1

**Table 12.** Prediction performance ( $k = 2; r = 1$ ).

$r = 1$	$C_1 - C_2$	$C_1 - C_3$	$C_2 - C_3$
$\alpha$	0	0	0
$\beta$	0.6667	0.3333	1
MPCD	0.3333	0.6667	0

**Table 13.** SumPLMCS, sum of the values of the combination created with  $k = 3$ .

Sample	$\sum PLC_1 - C_2 - C_3$	RL
1	1	0
2	1	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	2	1
9	2	1
10	2	1

**Table 14.** PLMCS, predicted labels with  $r = 0$ .

Sample	$\sum PLC_1 - C_2 - C_3$	RL
1	1	0
2	1	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	1	1
9	1	1
10	1	1

**Table 15.** Prediction performance ( $k = 3, r = 0$ ).

$r = 0$	$C_1 - C_2 - C_3$
$\alpha$	0.2857
$\beta$	0
MPCD	0.7142

**Table 16.** PLMCS, predicted labels with  $r = 1$ .

Sample	$\sum PLC_1 - C_2 - C_3$	RL
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	1	1
9	1	1
10	1	1

**Table 17.** Prediction performance ( $k = 3; r = 1$ ).

$r = 1$	$C_1 - C_2 - C_3$
$\alpha$	0
$\beta$	0
MPCD	1

**Table 18.** PLMCS, predicted labels with  $r = 2$ .

Sample	$\sum PLC_1 - C_2 - C_3$	RL
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	1
9	0	1
10	0	1



**Table 19.** Prediction performance ( $k = 3, r = 2$ ).

$r = 2$	$C_1 - C_2 - C_3$
$\alpha$	0
$\beta$	1
MPCD	0

**Table 20.** Predicted labels for comparative analysis.

Sample	PMQ-O	Majority	Unanimous		RL
			zero	one	
1	0	0	1	0	0
2	0	0	1	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	1	1	1	0	1
9	1	1	1	0	1
10	1	1	1	0	1

**Table 21.** Prediction performance of the comparative analysis.

	PMQ-O	Majority	Unanimous	
			zero	one
$\alpha$	0	0	0.2857	0
$\beta$	0	0	0	1
MPCD	1	1	0.7142	0

**Table 22.** Data sets information.

Data set	Description	Features	Training set	Validation set	Ratio (overall)
1	UMW	54	18,495 (20)	12,236 (9)	0.09 $\dagger\dagger$
2	Statlog (class 1)	36	4,435 (1,072)	2,000 (461)	23.82 $\dagger$
3	Occupancy Detection	5	6,587 (173)	1,791 (98)	3.23 $\dagger$
4	HTRU2	8	12,000 (1,484)	5,898 (155)	9.15 $\dagger$
5	Sensorless Drive	48	44,000/14,509	4,000/1,319	9.09 $\dagger$
6	Credit Card Fraud	29	200,000 (385)	84,807 (107)	0.17 $\dagger\dagger$

$\dagger$ highly unbalanced  $\dagger\dagger$ ultra unbalanced. In parentheses the count of the positive class.

The proposed list of *MLAs* support *fusion strategy*. The top performer is a different classifier in each data set, and within each data set, classifiers tend to mislabel different examples. These results support the coverage optimization problem that recommends including a diverse and complementary list of *MLAs* to reduce dependency/redundancy.

Because the proposed algorithm performs an exhaustive search to determine an optimal solution, there are certain rare cases in which the solution is not unique. In this situation, it is recommended to generate a test set to evaluate the generalization ability of *MCSs* to a new unseen data set.

### 5.3. Limitations, recommendations, and managerial insights

*PMQ* uses real-time process data, which usually are in the form of signals, warranty data, direct quality observations, or coordinates. In these cases, each of the steps of the problem solving strategy are applied to solve the pattern classification problem. Moreover, as per a recent study, most of today's problems can be more effectively solved by simple *MLAs*, rather than *deep learning* [58]. *MLAs* simplify model interpretation and understandability, enabling engineering knowledge creation aimed at process redesign and improvement. However, there are multiple image-based applications aimed at visual inspection replacement (i.e., train a *deep neural network* to identify whether a screw is present in a transmission). In these types of applications, the seven steps do not apply and only a *deep neural network* structure can be used, such as a *convolutional*.

*PMQ-O* proposes a list of eight diverse *MLAs*. However, this list can be modified (*MLAs* can be added/eliminated/changed) on the basis of user preference/intuition. Adding (or removing) an *MLA* would increase (or decrease) the number of combinations to be evaluated. The *Six Sigma* approach or reaching of the maximum possible level in an organization is a business strategy, which summarized includes financial benefits, increased productivity, and a better customer satisfaction [59].

## 6. Industry 4.0 - technologies and sustainability

Implementing *Quality 4.0* practices indicates the maturity of an organization to pursue the excellence in performance [60]. This can only be achieved with digital transformation, making technology innovation and connectivity a priority. The collection of data needed for generation of insights and propositions of value in almost real-time is one of the main pillars in *I 4.0*. For quality improvement, industries have to enable the acquisition of data and measure current processes to have a starting point to know where the optimization needs to occur. Sensors and smart sensors let this information to be interpreted. According to [61], monitoring equipment and environmental conditions in a manufacture floor, allow for diagnosis and analysis of the processes, creating competitive advantages in companies implementing these practices. As a high amount of data is required for better decision-making, intelligent transducers or smart sensors are used for this purpose. They encompass both analog and digital sensors with their corresponding signal, a microprocessor and a communication protocol for data transfer [62]. These sensors can be arranged in a network for monitoring process and performance. The cross-sensor data validation allows a high reliability in both the sensors and the network, creating an identifiable and easy detection of problems [61].

As per [63], quality and the implementation of *Six Sigma* or the corresponding sigma level are directly related to manufacturing processes, although *Six Sigma* is a generic improvement methodology. Competitiveness is on a rise, and moving from *Four Sigma* to *Six Sigma* would mean improvements to be integrated into normal operations. The future benefits would include financial and technological development, standardization, and safety measure implementation. The variation in processes should decrease over time, and therefore, creating more robust systems, which would mean less maintenance and investments for upgrades. Certain *key performance indicators (KPIs)* in a successful implementation [64] include the following: (1) efficiency, (2) cost reduction, (3) time to deliver, (4) quality of the service, (5) customer satisfaction, (6) employee satisfaction, (7) reduced variation, and (8) financial benefits.

The following step from obtaining the necessary data is to communicate it and process it in almost real-time to prevent failures or detect defects. This *big data* approach and *ML* usage are the current standard for detecting gaps in the process. The business value of implementing real-time processing is the ability for instantaneous streaming of information and reaction [65] and this value has to prevail for a company to keep investing and develop new products and services.

**Table 23.** Prediction ability (testing set) of each classifier by data set.

MLA		Data set					
		1	2	3	4	5	6
SVM	$\alpha$	0.001	0.015	0.029	0.010	0.017	0.040
	$\beta$	0.222	0.013	0.031	0.103	0.055	0.094
	M	0.777	0.972	0.941	0.888	0.929	0.8704
LR	$\alpha$	0.010	0.030	0.033	0.022	0.136	0.000
	$\beta$	0.111	0.022	0.010	0.077	0.124	0.252
	M	0.88	0.949	0.958	0.902	0.758	0.748
NB	$\alpha$	0.006	0.138	0.033	0.025	0.089	0.003
	$\beta$	0.222	0.135	0.010	0.090	0.049	0.187
	M	0.773	0.746	0.958	0.887	0.867	0.811
KNN	$\alpha$	0.000	0.003	0.023	0.009	0.004	0.000
	$\beta$	0.778	0.024	0.316	0.271	0.021	0.262
	M	0.222	0.974	0.668	0.723	0.975	0.738
ANN	$\alpha$	0.011	0.012	0.033	0.027	0.011	0.009
	$\beta$	0.111	0.035	0.010	0.052	0.017	0.150
	M	0.879	0.954	0.958	0.92	0.972	0.843
SVM RBF	$\alpha$	0.001	0.016	0.015	0.011	0.003	0.000
	$\beta$	0.778	0.017	0.327	0.090	0.007	0.402
	M	0.222	0.967	0.663	0.899	0.99	0.598
RF	$\alpha$	0.000	0.014	0.026	0.008	0.003	0.000
	$\beta$	0.889	0.050	0.174	0.168	0.019	0.280
	M	0.111	0.937	0.805	0.826	0.978	0.720
RUST Boost	$\alpha$	0.006	0.287	0.032	0.019	0.079	0.038
	$\beta$	0.222	0.089	0.000	0.142	0.030	0.150
	M	0.773	0.649	0.968	0.842	0.893	0.818

Top performer in bold, M means MPCD.

The selection of the adequate solutions highly depend on the practitioners leading the engineering teams. As generations pass by, new challenges are found when training new technicians to handle new technologies and models. According to [66], in *I 4.0*, engineers must be able to understand mobile technology, embedded systems, sensors, network technology, machine-to-machine communication, robotics, AI, bionics, and safety competencies, along with other managerial skills such as leadership, financial analysis, and critical thinking. The value of these abilities will change the educational system to meet current and future global demands in the industry [67]. To close the gap between generations, education needs to transform and enhance the benefits that technology can bring to the industry. Some of the main changes in the workplace have been the flexibility to work without fixed hours or places, unconventional projects or opportunities, abroad programs, experimental jobs and the constant use of technology [68].

Even though training and the possession of technology are important, the value of an organization lays in its ability to sustain itself. The majority of companies are not prepared to tackle problems that could happen three or more years in the future, which can lead to poor

**Table 24.** PMQ-O results by data set.

Data set	Classifier(s)	$\alpha$	$\beta$	MPCD	Fusion rule
1	LR,ANN	0.0076	0.1111	0.8821	1
2	SVM,SVM(RBF),ANN,KNN	0.0136	0.0065	0.9799	1
3	RUSBoost	0.0319	0	0.9681	0
4	ANN	0.0273	0.0516	0.9225	0
5	SVM(RBF),KNN,RF	3.790e-04	0.0030	0.9966	1
6	SVM,LR,ANN	0.0436	0.0841	0.8760	0

implementation of *Quality 4.0* initiatives, as they only try to optimize operational efficiency immediately [69]. The steps in the path for sustainability, according to the North Highland Consulting Company, are: (1) To recognize transformation is required, (2) Envision the future and build the case, (3) Create the strategy and success measures, (4) Socialize and align, (5) Prioritize and organize, (6) Execute and implement. The implementation of these steps will ensure an excellent execution and a constant attention for customer and company needs.

### 7. Conclusions

In today's manufacturing world, most mature organizations have merged traditional quality-control methods to create high-conformance production environments. Process-monitoring charts are being applied to improve the process capability index, at the industrial benchmark of *sigma level four*, where only a few *DPMOs* are generated. Detecting these defects to move manufacturing processes to the *next sigma level* is one of the primary intellectual challenges posed to the application of AI to manufacturing processes.

*PMQ* is a *Quality 4.0* initiative aimed at defect detection, where rare quality event detection is its primary application. Detection is formulated as a binary classification problem. Because manufacturing-derived data sets for binary classification of quality tend to be highly/ultra-unbalanced, *PMQ-O*, an *MCS* with the capacity to effectively analyze these data structures, was developed.

*PMQ-O* is founded on a list of eight diverse *MLAs*, an *ad hoc* fitness function (measure of classification performance), and a novel meta-learning algorithm. This algorithm takes the prediction from a set of classifiers as input, and determines which classifiers are reliable and which are not, in predicting the final quality class. As per several publicly available data sets, the *meta-learning algorithm* outperforms widely used *fusion rules* because they do not adapt to the characteristics of the data.

Because *PMQ-O* improves prediction by determining the *optimal fusion rule* from a set of classifiers, this development is a step forward in the path of moving manufacturing processes to *the next sigma level*.

Future research along this path can focus on evaluating different fitness functions for *PMQ-O*. This would allow for the proposed method to generalize to other types of data structures and regression problems.

## Declarations

### Author contribution statement

Carlos A. Escobar: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Daniela Macias: Performed the experiments.

Ruben Morales-Menendez: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

### Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

### Data availability statement

Data included in article.

### Declaration of interests statement

The authors declare no conflict of interest.

### Additional information

No additional information is available for this paper.

## Acknowledgements

Authors would like to thank Dr. Seog- Chan Oh and Dr. Megan McGovern for providing valuable feedback in the development of this analytical tool.

## References

- [1] J. Bughin, J. Seong, J. Manyika, M. Chui, R. Joshi, Notes from the AI Frontier: Modeling the Impact of AI on the World Economy, McKinsey Global Institute, Sep. 2018 [Online]. Available: [www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-modeling-the-impact-of-ai-on-the-world-economy](http://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-modeling-the-impact-of-ai-on-the-world-economy).
- [2] E. Of Encyclopaedia Britannica, "Industrial Revolution," Online, May 2021 [Online]. Available: [www.britannica.com/event/Industrial-Revolution](http://www.britannica.com/event/Industrial-Revolution).
- [3] R.E. Scott, The manufacturing footprint and the importance of US manufacturing jobs, in: Economic Policy Institute Briefing Paper, 388, 2015.
- [4] W. D, L. Ch (Eds.), Global Manufacturing Scorecard: How the US Compares to 18 Other Nations, BROOKINGS, Jul. 2018 [Online]. Available: [www.brookings.edu/research/global-manufacturing-scorecard-how-the-us-compares-to-18-other-nations/](http://www.brookings.edu/research/global-manufacturing-scorecard-how-the-us-compares-to-18-other-nations/).
- [5] I. W, E. O, D. A, "GDP Ranked by Country 2021," World Population Review, 2019, Apr. 2021.
- [6] A. Sreenivasulu Reddy, Y. Sunil, and G. Madhavi Reddy, "Study on Application of Six Sigma in Shoe Manufacturing Industry."
- [7] G. Sharma, P.S. Rao, B.S. Babu, Process capability improvement through dmaic for aluminum alloy wheel machining, J. Indust. Engin. Int. 14 (2) (2018) 213–226.
- [8] S. Deniz, M. C. imen, Barriers of six sigma in healthcare organizations, Manag. Sci. Letter. 8 (9) (2018) 885–890.
- [9] N.V. Fursule, S.V. Bansod, S.N. Fursule, Understanding the benefits and limitations of six sigma methodology, Int J. Scient. Res. Publ. 2 (1) (2012) 1–9.
- [10] C.A. Escobar, J. Arinez, R. Morales-Menendez, "Process-Monitoring- for-quality—a step forward in the zero defects vision, SAE Int. J. Manufact. Mater. (2020), 2020-01-1302.
- [11] R.D. Moen, C.L. Norman, Circling back, Qual. Prog. 43 (11) (2010) 22.
- [12] C. Escobar, M. McGovern, R. Morales-Menendez, Quality 4.0: a Re- view of big data challenges in manufacturing, J. Intell. Manuf. (2021) 1–16.
- [13] J.A. Abell, D. Chakraborty, C.A. Escobar, K.H. Im, D.M. Wegner, M.A. Wincek, "Big data driven manufacturing — process-monitoring- for-quality philosophy, ASME J. Manufactur. Sci. Eng. Data Sci. Enhanc. Manufactur. 139 (10) (2017).
- [14] D.H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. 8 (7) (1996) 1341–1390.
- [15] M. Wozniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.
- [16] D. Ruano-Ordás, I. Yevseyeva, V.B. Fernandes, J.R. Méndez, M.T. Emmerich, Improving the drug discovery process by using multiple classifier systems, Expert Syst. Appl. 121 (2019) 292–303.
- [17] G. Webb, L. Lee, F. Petitjean, B. Goethals, Understanding concept drift, arXiv preprint arXiv:1704.00362 (2017).
- [18] H. Wang, Z. Abraham, Concept drift detection for streaming data, in: Int J Conf on Neural Networks, IEEE, 2015, pp. 1–9.
- [19] J. Mount, "Don't Use Classification Rules for Classification Problems, The R Project for Statistical Computing, Aug. 2020.
- [20] J. Vanschoren, Meta-learning: a survey, arXiv preprint arXiv:1810.03548 (2018).
- [21] F. Roli, G. Giacinto, "Design of Multiple Classifier Systems," in *Hybrid Methods In Pattern Recognition*, World Scientific, 2002, pp. 199–226.
- [22] T.K. Ho, Complexity of classification problems and comparative advantages of combined classifiers, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 97–106.
- [23] C.A. Escobar, J.A. Abell, M. Hernández-de-Menéndez, R. Morales- Menendez, "Process-Monitoring-for-Quality — big models, Procedia Manufacturing 26 (2018) 1167–1179.
- [24] J. Devore, Probability and Statistics for Engineering and the Sciences, Cengage Learning, 2015.
- [25] A. Pannu, Artificial intelligence and its application in different areas, Artif. Intell. 4 (2015) 79–84.
- [26] Z. Ge, Z. Song, S. Ding, B. Huang, Data mining and analytics in the process industry: the role of machine learning, IEEE Access (5) (2017), 20 590–20 616.
- [27] Y. Oh, K. Ransikarbum, M. Busogi, D. Kwon, N. Kim, Adaptive SVM-based real-time quality assessment for primer-sealer dispensing process of sunroof assembly line, Reliab. Eng. Syst. Saf. 184 (2019) 202–212.
- [28] Y. Oh, M. Busogi, K. Ransikarbum, D. Shin, D. Kwon, N. Kim, Real-time quality monitoring and control system using an integrated cost-effective support vector machine, J. Mech. Sci. Technol. 33 (12) (2019) 6009–6020.
- [29] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach. Learn. 40 (2) (2000) 139–157.
- [30] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
- [31] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
- [32] Z. Zhou, Ensemble Learning, *Encyclopedia of Biometrics*, 2015, pp. 411–416.
- [33] S. Tulyakov, S. Jaeger, V. Govindaraju, D. Doermann, "Review of Classifier Combination Methods," in *Machine Learning In Document Analysis And Recognition*, Springer, 2008, pp. 361–386.
- [34] R.T. Clemen, Combining forecasts: a review and annotated bibliography, Int. J. Forecast. 5 (4) (1989) 559–583.
- [35] K. Tumer, J. Ghosh, Analysis of decision boundaries in linearly combined neural classifiers, Pattern Recogn. 29 (2) (1996) 341–348.
- [36] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, in: European Conf on Machine Learning, Springer, 2001, pp. 576–587.
- [37] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, in: Advances in Neural Information Processing Systems, 1995, pp. 231–238.
- [38] T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey, Meta-learning in neural networks: a survey, arXiv preprint arXiv:2004.05439 (2020).
- [39] J. Brownlee, What is meta-learning in machine learning? Tech. Rep. (Apr. 2021).
- [40] L. Rokach, Pattern classification using ensemble methods, World Scientific 75 (2010).
- [41] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.
- [42] Z. Xu, L. Cao, X. Chen, Meta-learning via weighted gradient update, IEEE Access 7 (2019), 110 846–110 855.
- [43] C. Xu, J. Shen, X. Du, A method of few-shot network intrusion detection based on meta-learning framework, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3540–3552.
- [44] A. Raghu, M. Raghu, S. Bengio, O. Vinyals, Rapid learning or feature reuse? towards understanding the effectiveness of maml, arXiv preprint arXiv:1909.09157 (2019).
- [45] S. Ray, Understanding Support Vector Machine Algorithm from Examples (Along with Code), July 7 2017 [Online]. Available: [www.analyticsvidhya.com/blog/2017/5/10/understaing-support-vector-machine-example-code](http://www.analyticsvidhya.com/blog/2017/5/10/understaing-support-vector-machine-example-code).
- [46] P. Valente Klaine, M. Ali Imran, O. Onireti, R. Demo Souza, A sur vey of machine learning techniques applied to self-organizing cellular networks, *IEEE Comm Surveys & Tutorials* (2017) 1.
- [47] X. Wu, V. Kumar, Q. Ross, J. Ghosh, Q. Yang, H. Motoda, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (2008) 1–37.
- [48] S.B. Imandoust, M. Bolandraftar, Application of K-nearest neighbor (KNN) approach for predicting economic events: theoretical background, Int. J. Eng. Res. Appl. 3 (5) (2013) 605–610.
- [49] H.B. Demuth, M.H. Beale, O. De Jess, M.T. Hagan, Neural Network Design, Martin Hagan, 2014.
- [50] K. Murphy, Machine Learning: A Probabilistic Perspective, MIT press, 2012.

- [51] I. Davidson, An ensemble technique for stable learners with performance bounds, in: AAAI, 2004, 2004, pp. 330–335.
- [52] S. Permission, Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression, 2005.
- [53] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: a hybrid approach to alleviating class imbalance, *IEEE Trans on Systems, Man, and Cybernetics-Part A: Systems and Humans* 40 (1) (2009) 185–197.
- [54] Y. Freund, R.E. Schapire, et al., Experiments with a new boosting algorithm, in: *Icml*, 96, Citeseer, 1996, pp. 148–156.
- [55] J. Friedman, T. Hastie, R. Tibshirani, *The Elements of Statistical Learning*, Statistics Springer, Berlin, 2001 vol. 1.
- [56] J. A. Abell, J. P. Spicer, M. A. Wincek, H. Wang, and D. Chakraborty, “Binary Classification of Items of Interest in a Repeatable Process,” US Patent , no. US8757469B2, June 2014. [Online]. Available: [www.google.com/patents/US20130105556](http://www.google.com/patents/US20130105556).
- [57] C.A. Escobar, R. Morales-Menendez, “Process-monitoring-for- quality—a model selection criterion for regularized logistic regression, *Procedia Manufacturing* 34 (2019) 832–839.
- [58] Y. Bathaee, “The artificial intelligence black box and the failure of intent and causation,” *harv, J. Technol.* 31 (2017) 889.
- [59] G. Goncalves, J. Reis, R. Pinto, M. Alves, J. Correia, A step forward on intelligent factories: a smart sensor-oriented approach, in: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation, (ETFA)*, 2014, pp. 1–8.
- [60] K. Lee, Ieee 1451: a standard in support of smart transducer networking, in: *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference [Cat. No. 00CH37066]*, vol. 2, IEEE, 2000, pp. 525–528.
- [61] R. Hoerl, One perspective on the future of six-sigma, *Int. J. Six Sigma Compet. Advant.* 1 (1) (2004) 112–119.
- [62] A. Chakraborty, K.C. Tan, The current state of six sigma application in services, *Manag. Serv. Qual.: Int. J.* (2007).
- [63] J. Antony, Six sigma in the UK service organizations: results from a pilot survey, *Manag. Audit J.* (2004).
- [64] N. Radziwill, *Connected, Intelligent, Automated: the Definitive Guide to Digital Transformation and Quality 4.0*, Quality Press, 2020 [Online]. Available: <https://books.google.com.mx/books?id=ISzsDwAAQBAJ>.
- [65] A. L'heureux, K. Grolinger, H.F. Elyamany, M.A. Capretz, Machine learning with big data: challenges and approaches, *IEEE Access* 5 (2017) 7776–7797.
- [66] M. Hernandez-de Menendez, R. Morales-Menendez, C.A. Escobar, M. McGovern, Competencies for industry 4.0, *Int. J. Interact. Des. Manuf.* 14 (4) (2020) 1511–1524.
- [67] M. Hernandez-de Menendez, C. Escobar Diaz, R. Morales-Menendez, Technologies for the future of learning: state of the art, *Int. J. Interact. Des. Manuf.* 14 (2) (2020) 683–695.
- [68] M. Hernandez-de Menendez, C.A.E. Diaz, R. Morales-Menendez, Educational experiences with generation z, *Int. J. Interact. Des. Manuf.* 14 (3) (2020) 847–859.
- [69] P. Zairi, *Leading Into The Future through Excellence: an Assessment Guide*, Ser. Quality 4. 0 Series. Independently Published, 2019 [Online]. Available: <https://books.google.com.mx/books?id=HIWBwgEACAAJ>.