Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

School of Engineering and Sciences

Process Monitoring for Quality – A Big Data and Artificial Intelligence-based Manufacturing Quality Control

A dissertation presented by

Carlos Alberto Escobar Diaz

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Engineering Sciences

Major in Machine Learning

Monterrey Nuevo León, México May 13, 2019

Dedication

I want to dedicate this accomplishment to my family. To my parents for instilling in me education as the main avenue for advancing and significantly contributing to our society. To my kids for inspiring me to be a better person and for cheering me up during difficult moments. Warm gratitude to my loved grandmother in heaven for forging my character and making me believe that I was a very capable person.

I hope this dissertation keeps the legacy and message to my kids that education is the main path for thriving and positively contributing to the development of our society.

Acknowledgements

I want to express my deepest gratitude to my mentor Dr. Ruben Morales, his support made my life-time goal possible, my PhD. His guidance always made me see the light in dark places, his demanding work and elevated expectations helped me to succeed in this program and in my professional life.

My most sincere gratitude to Dr. Jeff A. Abell for giving me the opportunity to join General Motors as a researcher in the lab under his lead (Manufacturing Systems research Lab – *MSR*) and for illuminating my research topic.

My sincere gratitude to Dr. Jorge Arinez for guiding and supporting my efforts in General Motors. Special thanks to Dr. Mike Wincek for mentoring me and pointing me out in the right research direction. Special thanks to my committee members Dr. Megan McGovern and Dr. Debejyo Chakraborty.

Finally, and very important, I want to express my special feelings to Tecnológico de Monterrey for my scholarship, specially to Dr. Neale R. Smith, the director of the PhD program.

Process Monitoring for Quality – A Big Data and Artificial Intelligence-based Manufacturing Quality

Abstract

As manufacturing companies stand on the brink of the fourth industrial revolution, *Industrial Big Data (IBD)* highlights the importance of the application of *Artificial Intelligence (AI)* in this domain. These technologies have the potential to enhance *Traditional Quality Control (TQC)* systems. Although most mature organizations generate only a few *Defects Per Million of Opportunities (DPMO)*, as in today's highly competitive global-market, customers expect perfect quality. Therefore, the detection of rare quality events represents not only a research challenge; but, also an opportunity to move quality standards forward. In this research, detection is formulated as a binary classification problem, where the main objective is to develop a predictive system that projects features into a hyper-dimensional space where those defects can be detected. Manufacturing-derived data sets for classification of quality pose two main challenges: (1) highly/ultra-unbalanced classes, (2) hyper-dimensional feature spaces, that often contain insignificant/irrelevant information. A new Big Data-Driven Manufacturing—*Process-Monitoring-for-Quality (PMQ)* philosophy has been developed. *PMQ* is a blend of *Process Monitoring (PM)* and *Quality Control (QC)*, which is founded on *Big Data and Big Models (BDBM)*. *PMQ* uses data from the process to make a real-time detection (classification). This new quality philosophy poses several theoretical challenges that must be addressed before it can be generalized across the manufacturing industry and materialize its contribution in the quality movement. This research work addresses a few of them. An analysis is presented on how *PMQ* enhances the quality movement by addressing three quality problems, which traditional *QC* techniques cannot. The predictive modeling paradigm of *Big Models (BM)* has been developed by aiming at rare quality event detection. Four *Model Selection (MS)* criteria have been developed: (1) *Penalized Maximum Probability of Correct Decision (PMPCD)*, a general criterion that can be applied to *Candidate Model (CM)* in which complexity is defined by the number of features, based on simulations, criterion induces parsimony while maintaining detection as the main driver. And a series of three-objective optimization *MS* criteria that use three of the most relevant competing attributes of each *CM* to project them into a Three-Dimensional (3D) space where the *final model* that solves the posed tradeoff between them the best is selected: (2) a *MS* criterion for the *Genetic Programming (GP)*, *3D-GP*, supported by a novel *Separability Index (SI)*, (3) a *MS* criterion for the *Support Vector Machine (SVM)*, *3D-SVM*, also supported by a novel SI, and (4) a *MS* criterion for the *Logistic Regression (LR), 3D-LR*, proposed criterion outperforms widely used criteria in highly/ultra-unbalanced data structures. A learning scheme for the L1-regularized *LR* algorithm has been developed following the *BM* paradigm, and it includes the development of two novel algorithms: (1) *Optimal Classification Threshold (OCTM)*, aimed at finding the classification threshold of a *LR*-based model, which is one of the main challenges of this learning algorithm and (2) *Hybrid Correlation and Ranking-based (HCR),* aimed at eliminating redundant features that filter feature selection methods cannot. Proposed learning scheme outperforms typical learning schemes in highly/ultra-unbalanced data structures with respect to detection and parsimony. The applicability of all theoretical developments and their capacity to solve real manufacturing complex-problems is demonstrated in three case studies, with the following detection results:

(1) *Ultrasonic Welding of Battery Tabs (UWBT)* – 100% detection, (2) *Laser Spot Welding (LSW)* – 100% detection and (3) *Sensorless Drive Diagnosis (SDD)* – 99.72% detection.

Contents

Table 1. Acronyms.

Acronym	Definition
3D	Three Dimensional
AI	Artificial Intelligence
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
BDBM	Big Data - Big Models
BIC	Bayesian Information Criterion
BM	Big Models
CEE	Cross-Entropy Error
CM	Candidate Model
CT	Classification Threshold(s)
DPMO	Defects per Million of Opportunities
DNN	Deep Neural Network
FN	False Negative
FP	False Positive
GM	General Motors
GP	Genetic Programming
HCR	Hybrid Correlation and the Ranking-based
IBD	Industrial Big Data
IIoT	Industrial Internet of Things
ISCS	Intelligent Supervisory Control Systems
KNN	K-Nearest Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LDC	Linear Discriminant Classifier
LR	Logistic Regression
LP	Learning Process
ML	Machine Learning
MLA	Machine Learning Algorithm(s)
MPCD	Maximum Probability of Correct Decision
MS	Model Selection
NB	Naïve Bayes
OCTM	Optimal Classification Threshold with respect to MPCD
PM	Process Monitoring
PMPCD	Penalized Maximum Probability of Correct Decision
PMQ	Process Monitoring for Quality
PR	Pattern Recognition
QC	Quality Control
SDD	Sensorless Drive Diagnosis
SI	Separability Index
SPC	Statistical Process Control
SQC	Statistical Quality Control
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

TQC	Traditional Quality Control
UWBT	Ultrasonic Welding of Battery Tabs
UMQ	Usage Monitoring for Quality
VS	Validation Set

Chapter 1

Introduction

In today's manufacturing world, most mature organizations have merged traditional quality philosophies to create a high conformance production environment, where their processes generate only a few *DPMO.* However, traditional techniques based on statistics have plateaued off and shown limitations in analyzing *IBD*.

In today's global competitive environment, customers expect perfect quality, and since they are using the internet and social media tools to share their experiences, leave organizations little to no flexibility to recover from their mistakes. Therefore, a single warranty event can drive a company's image and be the difference between it facing a profit or loss. A vivid example of this situation is the *GM* ignition switch, which was linked to fatal accidents [1]. This quality-related problem highlights the importance of a "zero-defect policy*". TQC* methods based on statistics do not achieve zero-defects in manufacturing.

As organizations stand on the brink of the fourth industrial revolution, *Industrial Internet of Things (IIoT)*, *IBD*, *acsensorization*, *cyberphysical systems* and *AI* are technologies propelling the new era of manufacturing, *smart manufacturing*. From the point of view of quality, these technologies have the potential to enhance traditional quality and productivity systems, in which one of the primary goals is to achieve *defect-free processes*. The main objective of this research is to apply *AI* techniques to *IBD* to develop a predictive system with the capacity to detect those very few *DPMO* – rare quality event detection – and to solve a whole new range of hitherto intractable manufacturing problems.

Today's high conformance manufacturing environment and *IBD* pose two main challenges to manufacturing-derived data sets for binary classification of quality: (1) highly/ultra-unbalanced classes, (2) hyper-dimensional feature spaces, that often contain insignificant/irrelevant information. In the context of this research, highly refers to data sets with the minority class count < 3%, and ultra to the minority class <1%.

1.1 Motivation

While manufacturing generates more data than any other sector of the economy, only few companies are harnessing it, because creating value out of *IBD* is one of today's biggest theoretical challenges posed.

More than three decades ago, in the era of Total Quality Management, the concept of "zero defects" was introduced by Mr. Philip Crosby in his book "Absolutes of Quality Management" [2]. At that time, it was only a vision; today, with the introduction of *AI* in manufacturing, this vision has become attainable. Thus, developing an *AI*-based predictive system with the ability to detect the small number of *DPMO* generated by manufacturing processes presents a unique opportunity to contribute in the path of developing *defect-free processes*.

1.2 Problem Statement and Context

Manufacturing systems are dynamic and complex in nature; therefore, *defect-free processes* rarely exist. From a customer's perspective, a *defect-free process* can be virtually accomplished either by: (1) a perfect process execution, or (2) a perfect inspection (detection) execution. The vision of this research is to create value out of *IBD* by developing a virtual *defect-free process* through perfect detection, where detection is formulated as a binary classification problem.

Table 2. Confusion matrix.

	Predicted good	Predicted defective
Good quality	True Negative (TN)	False Positive (FP)
Defective	False Negative (FN)	True Positive (TP)

Using the notation of binary classification, the problem is expressed using the confusion matrix, Table 2, where a positive result refers to a defective item, and a negative result refers to a good quality item. Since manufacturing systems only generate a few *DPMO*, binary classification of quality becomes a complex problem, because the classes tend to be highly/ultra-unbalanced.

Traditionally, statistics is applied to experimental data to develop an explanatory model or to define limits of *SPC* control charts, as shown in Figure 1. With this approach, most defects are detected (*TP*), and only a few of them are not (*FN*). If these "misses" are critical components, such as engines, transmissions, smoke detectors, etc., then these warranties could have an enormous impact, both economically and on the company's reputation.

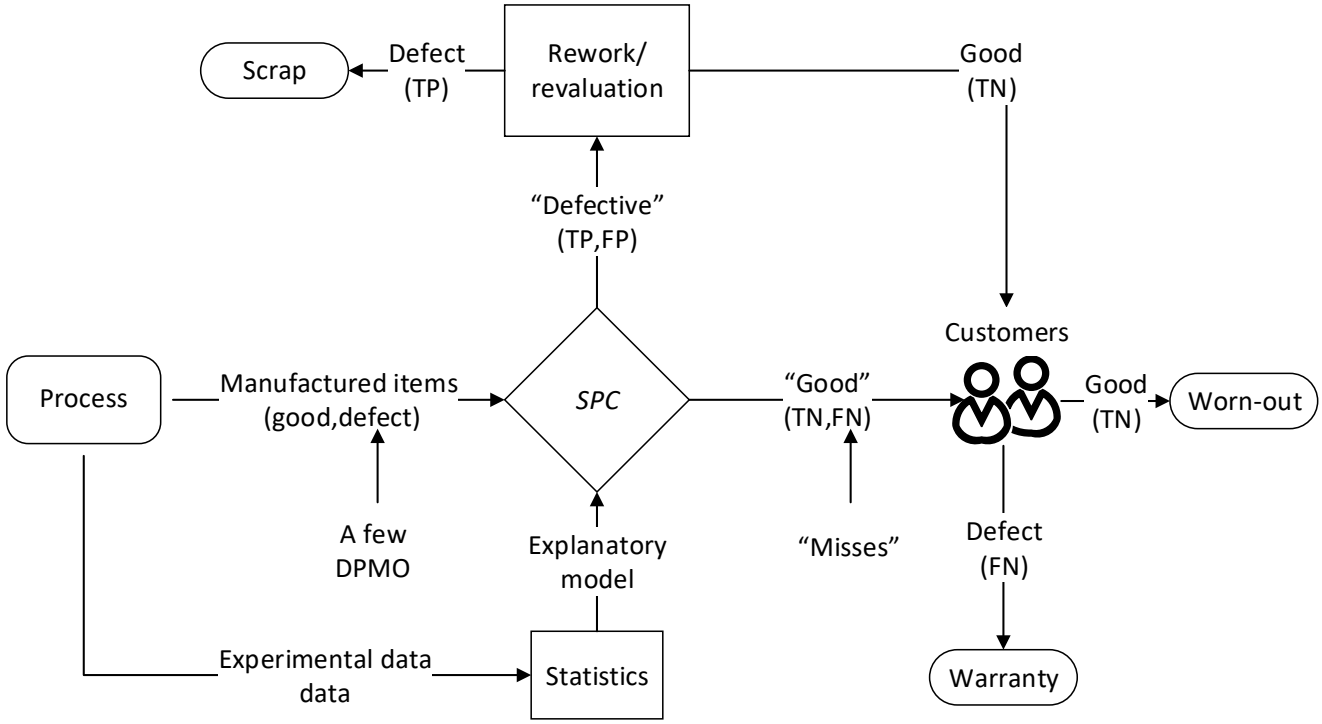


Figure 1. A traditional *QC* approach based on statistics.

The work here proposes to detect those very few *DPMO* (*FN*) that were not detected by *SPC* (rare quality event detection), as shown in Figure 1, enabling the creation of virtually *defect-free processes.*

1.3 State of the Art

Manufacturing companies are intense users of big data bases, and this industry generates/stores more data than any other [3]. Stored data becomes only useful, however, when it is properly analyzed and turned into information/action. The factory of the future is driven by manufacturing systems that exhibit fast increasing complexity, hyper-dimensional feature spaces as well as non-Gaussian, pseudo-chaotic behavior, which militate against orthodox *SPC*, but open a whole new avenue of opportunities for *AI*. A brief description of its application in manufacturing for *QC* – defect detection – is presented.

The concept of zero defects in manufacturing reemerged in 2013, when Ke-Sheng [4] presented a general framework about how to apply data mining in manufacturing. The author described the basic components of a quality monitoring system aimed at fault diagnosis or failure prognosis. A pilot study, *3D* intelligent quality inspection system, was also performed to support their framework. Three *MLA* were used to perform the quality inspection (*SVM, ANN, CT*), and k-fold cross-validation was applied to evaluate the performance of each classifier. The best classifier (*SVM*) was selected based correctness, i.e., percentage of samples correctly classified, and the *FP* rate. The proposed framework does not support the analysis of highly/ultra-unbalanced data structures, and it assumes no time effect, since cross-validation is used. Finally, no decision combination scheme aimed at improving prediction was reported.

A hybrid process monitoring and diagnosis method based on cluster analysis and the *SVM* algorithm was developed by Wuest et al. in 2014 [5]. In this work, the authors proposed a “red flag” warning system aimed at modelling the dependencies between the different states or state characteristics. The approach allows for experts to decide whether the product can still reach the final quality requirements or if it should be scrapped. Clustering analysis was applied to identify potential undesirable states by isolating extreme states and the *SVM* to determine, in quasi real-time, the process/product states at various points in the overall manufacturing system. The authors concluded that process monitoring based on clustering analysis and supervised *ML* on product state based data can potentially increase quality in manufacturing.

In 2017, Granstedt Möller [6] presented a *ML* application for quality control in Scania Engines based on the Convolutional Neural Network algorithm, where the objective was detection in highly unbalanced classes to separate bad from good. A general learning approach was followed based on k-fold cross-validation; therefore, the time-effect was assumed negligible. Preliminary prediction performances were reported in terms of type-I and type-II errors; however relative good detection was achieved at the expense of high type-I error. The author recognized that quality from the production process does not get better directly by detection; however, it increases the ability to control the production and therefore, prevents defects from reaching the customers.

Over the past decade, machine vision and *ML* have been combined and widely used across manufacturing applications to develop quality inspection systems. An automated feature extraction in industrial inspection is presented in [7], where *DNN* configurations were developed

for visual fabric defect detection. The proposed method demonstrated excellent defect detection results with low false alarm rates. An intelligent automated visual inspection system that effectively detects defective thermal fuses using *ANN* and machine vision is presented in [8]. This system not only reduces costs associated with human inspectors but also increases inspection reliability and speed. For safety considerations, all settings were aimed at eliminating the type-II error. There have also been some successful applications published on quality of fruit detection using *MLA* [9, 10], where various *MLA* were used (e.g., *DNN, ANN, CT, LDC, KNN, SVM*). Implementing a decision combination to improve prediction was identified as a future research work.

In recent years, *MLA* have been also efficiently applied for automatic defect detection in surfaces. A general framework for defect detection of hot rolled steel surface based on the *SVM* is proposed in [11]. The G-mean and F-measure were used to evaluate the detection ability of the classifiers, and cross-validation was used for hyper-parameter tuning. The authors recognized that correctness is not appropriate measure of classification performance for defect detection, and feature selection and decision combination strategies were laid out as a future work to improve the proposed framework. To distinguish discrepant from normal parts, a novel classification approach for engineering surfaces is proposed in [12], based on an ensemble that uses the *SVM* as the basic classifiers. A prediction of surface roughness by multiple regression analysis is presented in [13]. To achieve higher predictability and low computational effort, this approach included a dimensionality reduction step using principle component analysis.

To advance the state-of-the-art, Bosch held a competition in 2016, where they shared production data. The goal was to classify manufacturing failures. The winners of this competition used XGBoost and Random Forests for their algorithms [14].

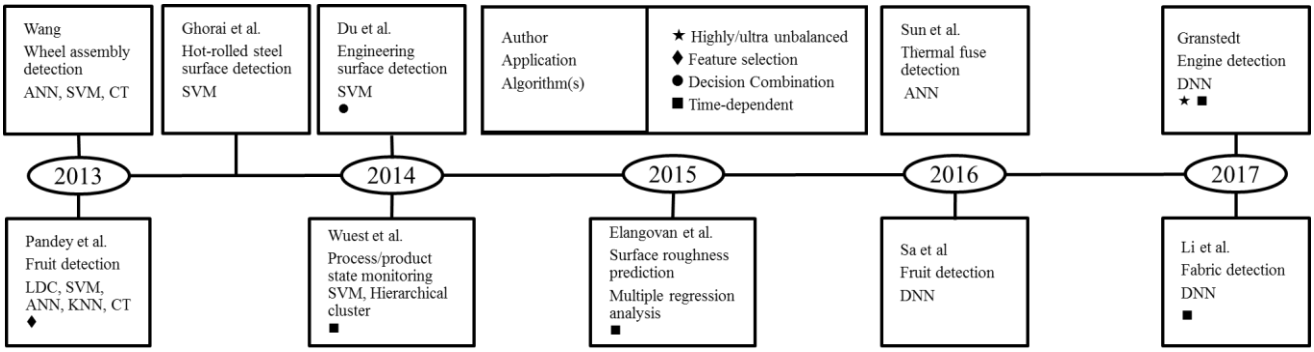


Figure 2. State of the art analysis with respect to applications and characteristics of the solution.

AI is a fast-paced technology that has shown promising results in addressing old and new challenges in manufacturing because of its ability to handle non-linear and hyper-dimensional data. However, the plethora of options of *MLA* theories and methods poses a barrier for technology adoption, and thus hinders efficient utilization if *IBD* [15]. Poor implementation of *AI* and inadequate manufacturing knowledge can lead to erroneous conclusions, low-quality solutions, and/or may require a lot of iterative work to solve a problem [5]. Since the data structures are not known in advance, there is no one best strategy to solve the detection problem. Furthermore, binary classification in manufacturing poses the following challenges:

(1) *ML*-based solutions (models) should be grounded in physics and have some level of "trust-ability" [16,17] to be deployed, (2) classes tend to be highly/ultra-unbalanced, (3) hyper-dimensional feature spaces, including relevant/redundant features [15], and (4) systems tend to be time-dependent.

Figure 2 contains a summary of the state-of-the-art. As shown in the figure, researchers have tailored different frameworks based on their applications and goals. Their solutions are presented in the context of challenges posed by typical manufacturing systems. It is observed that very little research has been reported for defect detection in cases when the classes are highly/ultra-unbalanced. Most of the reviewed papers include feature extraction and dimensionality reduction methods, which do not necessarily select/identify the most relevant features of the system [18,19]. Only one author [9] considers feature selection methods, and another [12] considers a decision combination scheme. No one has developed a solution that considers all four dimensions (denoted by the various shapes in Figure 2).

1.3.1 Statistical Vs Artificial Intelligence Quality Control

TQC philosophies highly rely on physics knowledge and statistics to create/test hypothesis, where the main goal is to build a set of facts as part of a theory of a larger body of knowledge and the focus is long-term. *SPC* is one of the primary most widely used tools for quality control in manufacturing. It is a method which employs statistical methods to monitor and control a process under a well understood cause and effect framework. Key tools used include run charts, control charts, a focus on continuous improvement, and the design of experiments. *SPC* highly relies on visual inspection and human judgement to diagnoses the process-quality properly. All the thinking, planning, and model formulation is generally done before data is generated, collected, and analyzed. The following requirements must be met for a successful deployment of *SPC* [20-23]:

(1) A comprehensive understanding of the underlying physics (cause-effect) of the process to define the product quality characteristics to be measure and their associated control limits.
(2) A feasible measure system within the temporal and physical constraints of the manufacturing plant environment.
(3) A normal distribution of the observed variables to establish the control limits. However, this is a requirement difficult to meet in practice since skewness distributions are commonly observed.

Statistical analyses are aimed at causal inference [24], where data is generated under experimental conditions and used to train explanatory models [17]. These models are then commonly validated based on p-values [25]. Data partition is not recommended, since it would decrease the power of the analysis. Data snooping is a dangerous practice to be avoided.

On the other hand, *MLA* learn information directly from data without assuming a predetermined equation, probability distribution, or model. They are primarily executed in real-time to predict pattern classification (detection), image and voice recognition, anomaly detection, etc. In the *ML* domain, generalization error is a function that measures how well a trained algorithm predicts unseen data [26], and it is used to validate/deploy a model to determine if the learning targets for a project are met.

MLA are trained using observational data [27]; therefore, *ML* models should not be used for theory building [24,17]. Instead, these models should be used to extract information and uncover patterns, associations, and correlations for hypothesis building and guiding randomized experiments. *AI* offers a new paradigm for automatic process learning and quality control. A contrastive analysis of traditional quality control paradigm based on statistics vs. *AI* is summarized in Table 3.

Table 3. Statistical quality control vs machine learning control.

	Statistical Quality Control	Total Quality Mgmt.	Six Sigma	Design for Six Sigma	Industrial Big Data	Artificial Intelligence
Contrastive Characteristics	Traditional quality analysis				Machine learning analysis	
Data	Experimental				Observational	
Feature space	Small				Big (hyper-dimensional)	
Learning	Hypothesis-based				Discovery-based	
Model	Explanatory (causal inference)				Predictive (causal discovery)	
Assumptions	Theoretical Prob. Dist., homoscedasticity				No	
Validation	Statistical significance (p-value)				Generalization	
Analysis	Real-time and asynchronous				Real-time	
Data snooping	No				Yes (iterative modeling)	
Solution	Long term				Short term	
Relearning	No				Yes (adaptive learning)	
Data partition	No (all data is used for training)				Yes (training, validation, test)	

1.4 Research Questions

Based on the discussions in the previous subsections, the following research questions are posed:

- How can *AI* (*ML*) be applied to develop a new *QC* paradigm?
- How can *AI* (*ML*) be applied for defect detection in highly/ultra-unbalanced binary data structures?
- What are the main challenges posed by manufacturing-derived data in terms of detection?
- What type of manufacturing problems can be solved by AI but not by TQC methods?

1.5 Solution Overview

The goal of this work is to address the research questions posed in Section 1.4. *PMQ,* a new *IBD* and *AI*-driven quality philosophy has been developed considering the challenges posed by manufacturing systems [28]. *PMQ* is founded on *BDBM*, Figure 3, a modeling paradigm that uses real-time observational data to virtually project each manufactured item into a hyper-dimensional space where those rare quality events can be detected. *PMQ* proposes a new problem-solving strategy: *Acsensorize, Discover, Learn, Predict (ADLP)* to guide engineers in

solving the classification problem, Figure 4. The *Learn* and *Predict* steps are solved by *BM.* As shown in Figure 5, *PMQ* can complement *SPC* to develop virtually *defect-free processes*.

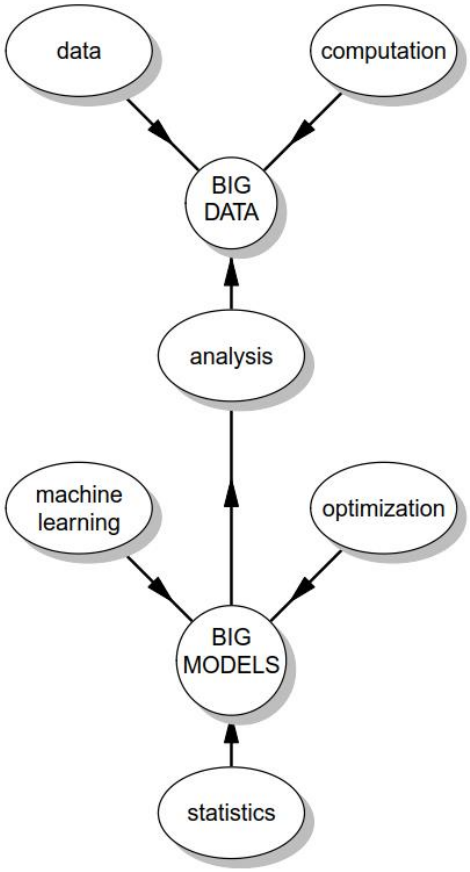


Figure 3. *BD-BM* framework [28].

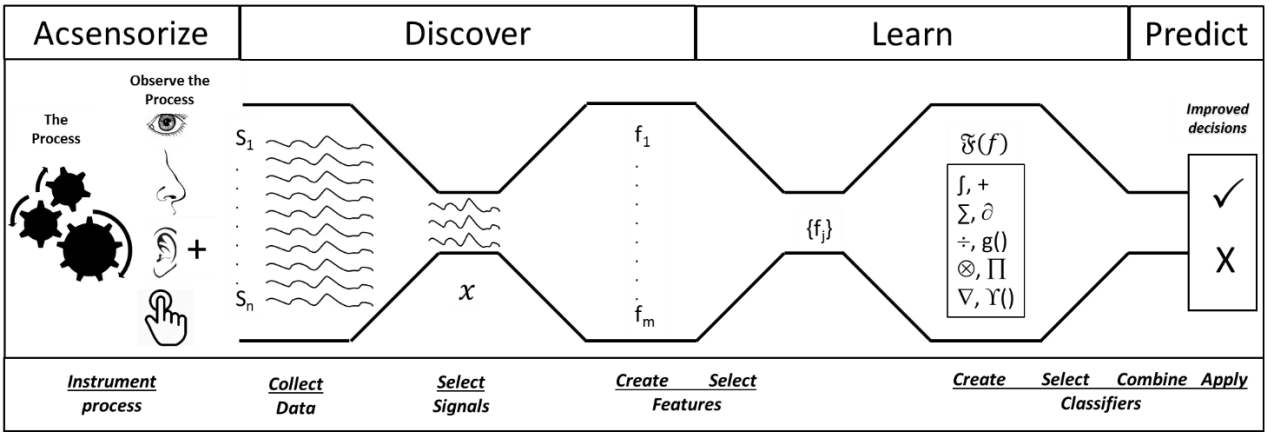


Figure 4. *PMQ* problem solving strategy [28].

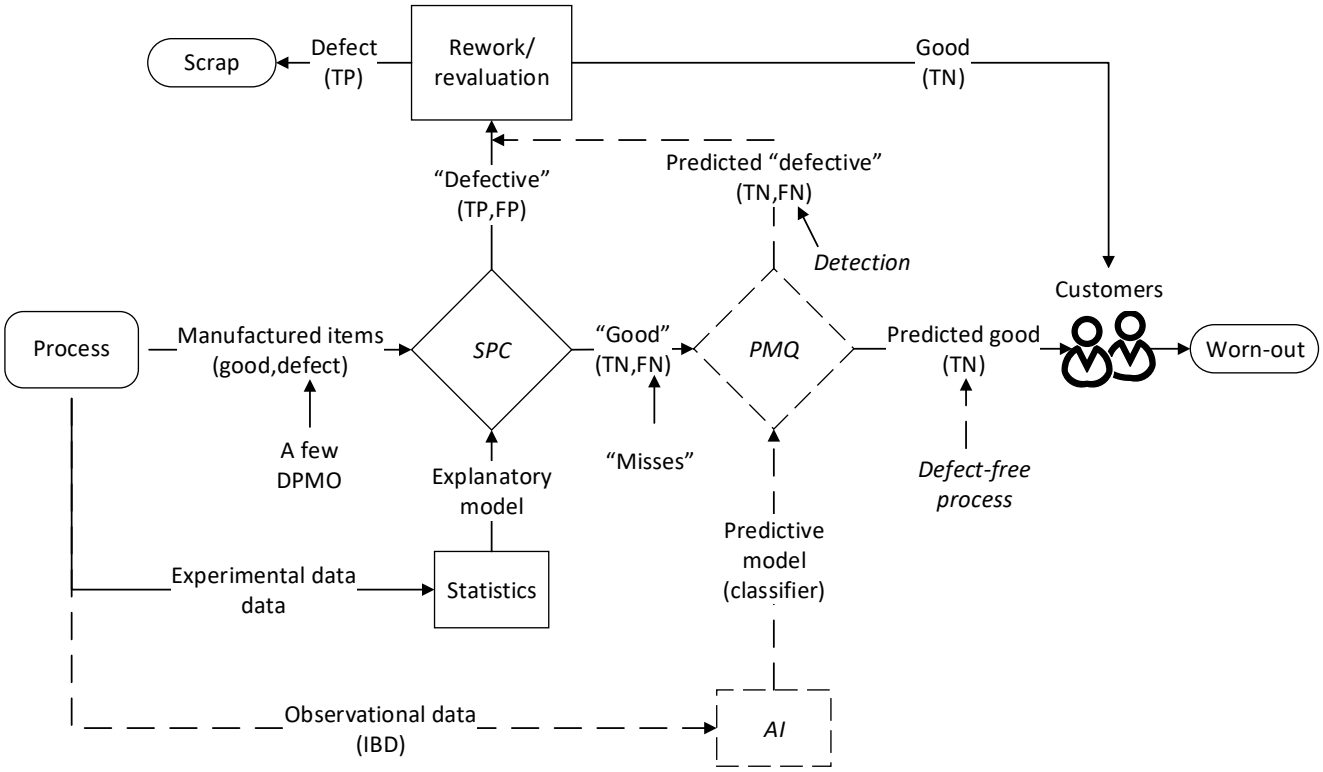


Figure 5. Conceptual framework of how *PMQ* supports SPC in the development of a virtually *defect-free process*.

The modeling paradigm of *BM* has been developed, which is based on machine learning, statistics, and optimization techniques aimed at developing a classifier with the capacity to detect rare quality events, see Figure 5.

BM includes a learning aspect that requires many models to be developed to find the *final model* or a group of classifiers (ensemble/non-ensemble); therefore, *MS* is one of the main challenges of *BM*. To facilitate information extraction and increase model's "trust-ability" and understandability, *BM* is founded on the principle of parsimony, where parsimony is induced through *MS* and *Feature Selection (FS)*. It is a practical guide for engineers to develop a manufacturing functional model for rare quality event detection. The following novel developments support the *BM* learning paradigm.

(1) A learning scheme for the L1-regularized *LR* was developed, which shows how machine learning techniques can be applied for quality control in high conformance manufacturing environment and information extraction. To improve the detection ability of this *MLA*, the learning scheme includes the development of a novel algorithm (*OCTM*) that finds the Optimal *Classification Threshold (CT)* with respect to the *Maximum Probability of Correct Decision (MPCD)*.

(2) The *HCR* filter-type feature selection algorithm eliminates redundant features based on Pearson's correlation coefficients [25] and a feature-ranking algorithm [29]. The basic idea is to keep the best feature, e.g., one which is highest ranked, from a group of two or more highly correlated features.

(3) To select the *final model*, *the PMPCD MS* criterion was developed. The proposed criterion is estimated based on the confusion matrix and the number of features; therefore, it can be widely applied as a *MS* criterion for virtually any classifier for which model complexity can be defined by the number of features (e.g., *KNN, SVM, LR, NB*). *PMPCD* is aimed at analyzing highly/ultra-unbalanced data structures, where its score values mainly indicate the detection ability of the classifier. According to simulation results, the criterion always selects the true model. Therefore, it can be applied to solve the posed tradeoff between complexity and prediction, as it discards models with extra features with little to no discriminatory information.

A novel concept for model selection criterion was introduced. It uses three of the most relevant competing attributes of each *CM* to project them into a *3D* space to select the *final model* that solves the posed tradeoff between them the best. The three associated attribute values are mapped into a *3D* space and the weighted Euclidean distance to the utopian point is computed. Then, the closest model is selected. In this context, the *utopian* point, is an ideal model that optimizes the three attribute-functions simultaneously. This concept was extended to the *GP*, *SVM* and *LR MLA*.

(4) Since *GP* is a technique free of human preconceptions or biases that highly relies on computer power to develop a good predictive model, this *MLA* is also part of the solution. However, this technique requires the creation/testing of thousands of models to find a good model. In this context, selecting the *final model* is one of the most relevant challenges. A *3D MS* criterion, *3D−GP*, based on prediction, separability, and complexity was developed. [30].

(5) Determining the amount of regularization (hyperparameter λ) is one of the main challenges of the L1-regularized *LR* algorithm, which is addressed through *MS. A 3D MS criterion, 3D−LR, based on prediction, Cross-Entropy Error (CEE), and complexity was developed* [31].

(6) The same concept was extended to the *SVM* algorithm. A 3D MS criterion, *3D−SVM*, based on prediction, separability, and complexity was developed [32].

(7) To evaluate the robustness of the detections performed by of *GP* and *SVM* models, *3D-GP* and *3D-SVM* criteria are supported by novel separability indexes, which are aimed at analyzing highly/ultra-unbalanced data structures. Their scores are computed based on the confusion matrix and the distance of each item to the *CT*. First, predicted classifications are used to populate the confusion matrix, then the separability indexes use the distances to reward for correct classifications (*TN,TP*) and to penalize for misclassifications (*FN,FP*) in the *Validation Set (VS)* [30,32].

In addition to rare quality event detection, three *SPC*-intractable applications that can be addressed by *PMQ* were identified: (1) infeasible measurement problem, (2) delayed measurement problem, and (3) uncertain characteristic problem. Common problems in which the requirements (section 1.2.1) are not satisfied. In these three cases, *PMQ* can supplement an empirical predictive framework for process control without *SPC*, as shown in Figure 6.

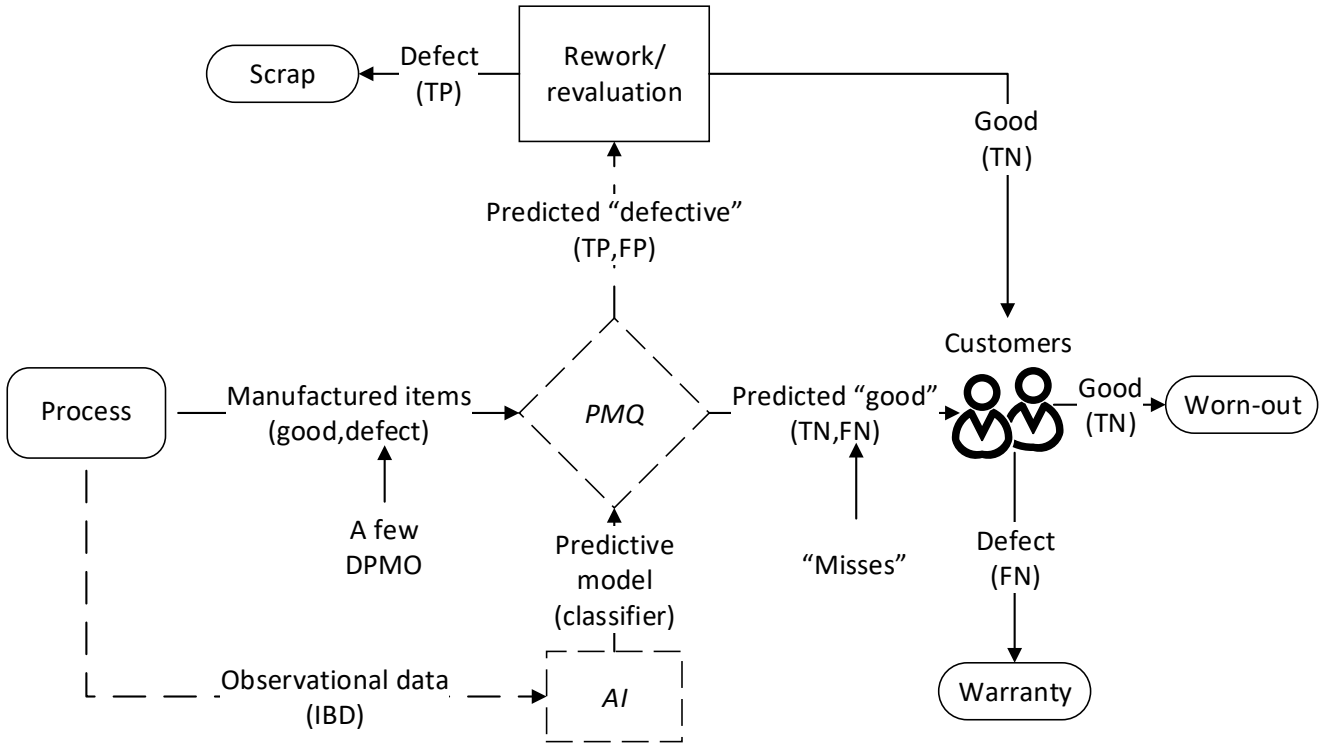


Figure 6. Application of *PMQ* without *SPC*.

1.6 Proposed Solution by Chapters

Figure 7 shows a high-level overview of the proposed solution by chapter.

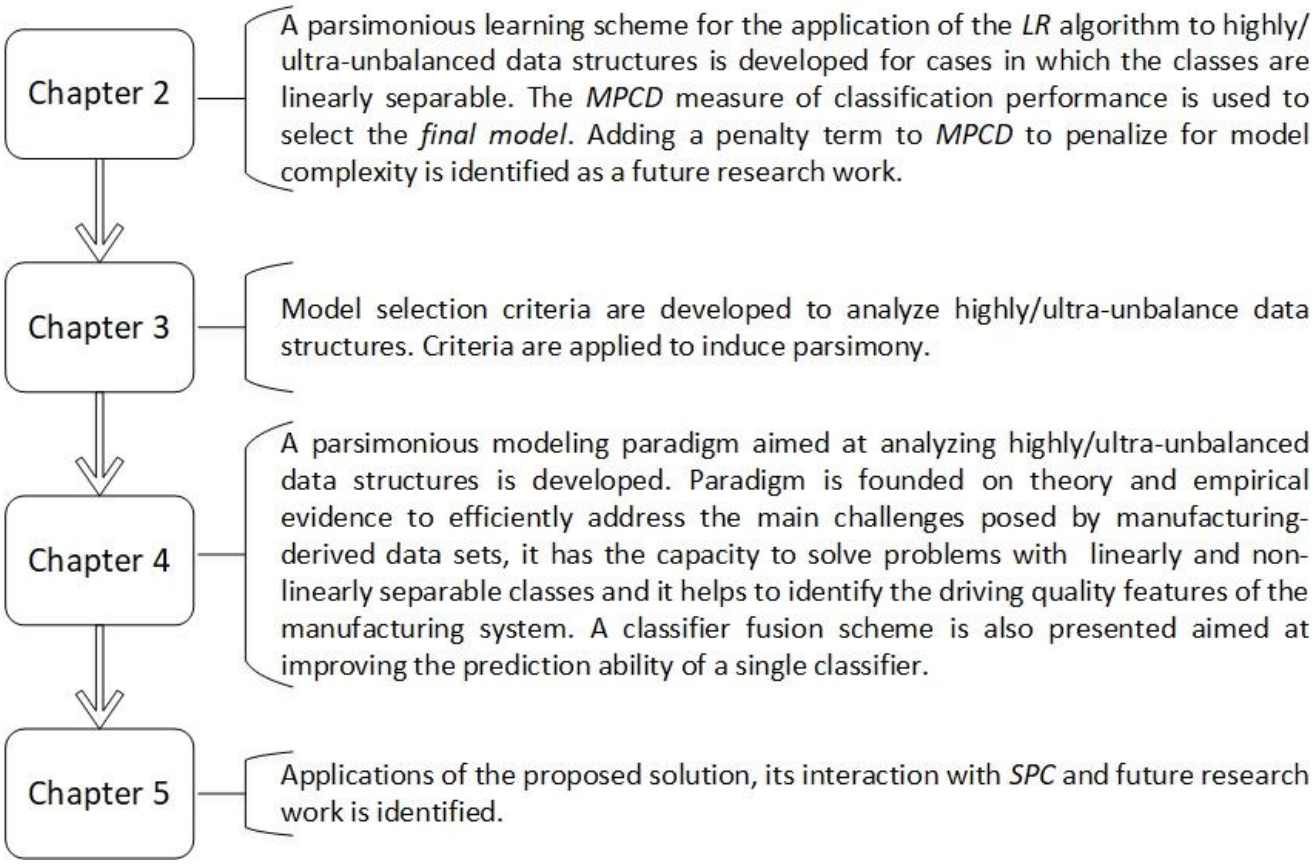


Figure 7. Proposed solution by chapters

A high-level overview (extended abstract) is included at the beginning of each chapter that describes its main content and contributions to the solution of the problem. Empirical evaluations of the theoretical developments are performed using a data set derived from the *UWBT*, an extended description of this process is presented on Appendix A. Bibliographic references of publicly available data sets used are provided at the end of each chapter.

Chapter 2

Machine learning techniques for quality control in high conformance manufacturing environment

The primary goal behind the generation and analysis of *IBD* is to achieve fault-free (defect-free) processes, through *Intelligent Supervisory Control Systems (ISCS)*. A L*earning Process (LP)* and *Pattern Recognition (PR)* strategy for a K*nowledge-Based (KB) ISCS* is presented, aimed at detecting rare quality events from manufacturing systems. The defect detection is formulated as a binary classification problem, in which the l1-regularized *LR* is used as the learning algorithm.

The outcome of the proposal is a parsimonious predictive model that contains the most relevant features. The model is validated using data derived from two automotive manufacturing systems: (1) *UWBT* from a battery assembly process and (2) *Laser Spot Welding (LSW)* subassembly components from an assembly process. Where the main objective is to detect low-quality welds (bad) from the processes. Challenges and theoretical properties of the *LR* algorithm are considered in the learning strategy.

For solving the l1-regularized LR, the *Least Absolute Shrinkage and Selection Operator (LASSO)* is an efficient method. In general, high correlations among features may hamper the *LASSO* in finding the true model. It may not be able to distinguish true features with any amount of data and any amount of regularization. Therefore, eliminating highly correlated features is one of the main challenges.

A new hybrid correlation and ranking-based, *HCR* algorithm is developed. It eliminates redundant features based on Pearson's correlation coefficients and a feature-ranking algorithm. The basic idea is to keep the best feature, i.e., highest ranked, from a set of two or more highly correlated variables.

Another challenge for the *LR* algorithm is to determine the *CT*. Since faulty events rarely occur, manufacturing-derived data sets for binary classification of quality tend to be highly/ultra-unbalanced. To address this scenario, the *MPCD* is used as a model selection criterion. Since this measure of classification performance is used to evaluate the fitness of each candidate model, an algorithm that finds the optimal *CT* with respect to *MPCD* (OCTM) is developed.

Finally, to evaluate its performance, a comparative analysis is performed following a typical modeling analysis, and results are compared and briefly discussed. Based on the comparative analysis, the models developed following the proposed *LP* and *PR* strategy exhibited better parsimony properties and good (or even better) detection capacity when compared with a typical l1-regularized *LR* analysis with three popular model selection criteria.

The proposed approach can be adapted and widely applied to manufacturing processes to boost the performance of traditional quality methods and potentially move quality standards forward, where soon virtually no defective product will reach the market. Since *MPCD* does not penalize for extra features in the model, adding a penalty term for model complexity was identified as a future research work, which is addressed in the following chapter.

Special Issue Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
2018, Vol. 10(2) 1–16

DOI: 10.1177/1687814018755519
journals.sagepub.com/home/ade
SAGE

Machine learning techniques for quality control in high conformance manufacturing environment

Carlos A Escobar[1,2] and Ruben Morales-Menendez[2]

Abstract

In today's highly competitive global market, winning requires near-perfect quality. Although most mature organizations operate their processes at very low defects per million opportunities, customers expect completely defect-free products. Therefore, the prompt detection of rare quality events has become an issue of paramount importance and an opportunity for manufacturing companies to move quality standards forward. This article presents the learning process and pattern recognition strategy for a knowledge-based intelligent supervisory system, in which the main goal is the detection of rare quality events. Defect detection is formulated as a binary classification problem. The l_1-regularized logistic regression is used as the learning algorithm for the classification task and to select the features that contain the most relevant information about the quality of the process. The proposed strategy is supported by the novelty of a hybrid feature elimination algorithm and optimal classification threshold search algorithm. According to experimental results, 100% of defects can be detected effectively.

Keywords

Manufacturing, l_1-regularized logistic regression, classification threshold algorithm, defect detection, feature elimination algorithm, model selection criterion, quality control, unbalanced data

Date received: 6 March 2017; accepted: 17 November 2017

Handling Editor: Baozhen Yao

Introduction

In today's highly competitive global market, winning requires near-perfect quality, since intense competition has led organizations to low profit margins. Consequently, a warranty event could make the difference between profit and loss. Moreover, customers use Internet and social media tools (e.g. Google product review) to share their experiences, leaving organizations little flexibility to recover from their mistakes. A single bad customer experience can immediately affect companies' reputations and customers' loyalty.

In the quality domain, most mature organizations have merged business excellence, lean production, standards conformity, six sigma, design for six sigma, and other quality-oriented philosophies to create a more coherent approach.[1] Therefore, the manufacturing processes of these organizations only generate a few defects per million of opportunities. The detection of these rare quality events represents not only a research challenge but also an opportunity to move manufacturing quality forward.

Impressive progress has been made in recent years, driven by exponential increases in computer power, database technologies, *machine learning (ML)* algorithms, optimization methods, and big data.[2]

[1]Global Research and Development, General Motors, Warren, MI, USA
[2]Dean of Graduate Studies, Tecnológico de Monterrey, Monterrey, México

Corresponding author:
Carlos A Escobar, Global Research and Development, General Motors, Warren, MI 48092, USA.
Email: Carlos.1.escobar@gm.com

From the point of view of manufacturing, the ability to efficiently capture and analyze big data has the potential to enhance traditional quality and productivity systems. The primary goal behind the generation and analysis of big data in industrial applications is to achieve fault-free (defect-free) processes,[3,4] through *intelligent supervisory control systems* (*ISCS*).[5]

A *learning process* (*LP*) and *pattern recognition* (*PR*) strategy for a knowledge-based (KB) *ISCS* is presented, aimed at detecting rare quality events from manufacturing systems. The defect detection is formulated as a binary classification problem, in which the l_1-regularized *logistic regression* (*LR*) is used as the learning algorithm. The outcome of the proposal is a parsimonious predictive model that contains the most relevant features.

The proposed strategy is validated using data derived from two automotive manufacturing systems: (1) *ultrasonic metal welding* (*UMW*) battery tabs from a battery assembly process and (2) *laser spot welding* (*LSW*) subassembly components from an assembly process. The main objective is to detect low-quality welds (bad) from the processes.

The initial idea of rare quality event detection through KB ISCS was initially introduced in Escobar and Morales-Menendez.[6] The proposal is extended—improved with respect to classification and parsimony—in this article with the introduction of two algorithms; these algorithms are aimed at addressing two of the most relevant challenges posed by the l_1-regularized *LR* algorithm. Challenges and theoretical properties are briefly discussed. To show the ability of the proposal in dealing with high-dimensional balanced data, another case study (LSW) is presented. Finally, to evaluate its performance, a comparative analysis is performed following a typical modeling analysis, and results are compared and briefly discussed.

The rest of this article is organized as follows: It starts with a review of the theoretical background in Section "LP and PR strategy" describes the proposal. Two studies in section "Case studies" followed by the "Comparative analysis." Finally, "Conclusion" and "Future work" conclude this paper.

Theoretical background

The theoretical background of this research is briefly reviewed.

ML and PR

As discussed by Ghosh,[7] "As an intrinsic part of *Artificial Intelligence (AI)*, *ML* refers to the software research area that enables algorithms to improve through self-learning from data without any human intervention." *ML* algorithms learn information directly from data without assuming a predetermined equation or model. The two most basic assumptions underlying most *ML* analyses are that the examples are independent and identically distributed, according to an unknown probability distribution. *PR* is a scientific discipline that "deals with the automatic classification of a given object into one from a number of different categories (e.g. classes)."[8]

In *ML* and *PR* domains, generalization refers to the prediction ability of a learning algorithm model on unseen data.[9] The generalization error is a function that measures well a trained algorithm generalizes.

In general, the *PR* problem can be widely broken down into three components: (1) feature space reduction, (2) classifier design and selection, and (3) classifier assessment.

Feature space reduction

In *ML* and *PR*, a feature is an individual measurable property of an observed phenomenon.[10] The prediction ability of the classifier is determined by the inherent class information available in the features.[11] In general, a feature is good if its inherent class information is relevant to one of the class labels but is not redundant to other good features. If the correlation of two variables is used as a goodness measure, a good feature should be highly correlated to one of the class labels but not highly correlated to any other features.[12,13] A feature can be considered irrelevant if the information that it contains is independent from the class label.

The world of big data is changing dramatically, and feature access has grown from tens to thousands, a trend that presents enormous challenges in the *feature selection* (*FS*) context. Empirical evidence from *FS* literature exhibits that discarding irrelevant or redundant features improves generalization, helps in understanding the system, eases data collection, reduces running time requirements, and reduces the effect of dimensionality.[12–17] This problem representation highlights the importance of finding an optimal feature subset. This task can be accomplished by *FS* or regularization.

FS. Filter-type methods select variables independently of the classification algorithm or its error criteria, they assign weights to features individually and rank them based on their relevance to the class labels. A feature is considered good if its associated weight is greater than the user-specified threshold.[12] The advantages of feature ranking algorithms are that they do not over-fit the data and are computationally faster than wrappers, and hence, they can be efficiently applied to big datasets containing many features.[13] However, most common methods—*Mutual Information*, *ReliefF*, and so on—do not help in removing redundant features, as

features are evaluated independently; therefore, as long as features contain class discriminatory information, they will be selected, even if they are highly correlated to each other.[12,18,19]

ReliefF is a well-known rank-based algorithm, and the basic idea for numerical features is to estimate the quality of each according to how well their values distinguish between instances of the same and different class labels. *ReliefF* searches for a k of its nearest neighbors from the same class, called nearest *hits*, and also k nearest neighbors from each of the different classes, called nearest *misses*; this procedure is repeated m times, which is the number of randomly selected instances. Thus, features are weighted and ranked by the average of the distances (Manhattan distance) of all *hits* and all *misses*[20] to select the most important features,[18] developing a significant threshold τ. Features with an estimated weight below τ are considered irrelevant and, therefore, eliminated. The proposed limits for τ are $0<\tau\leq 1/\sqrt{\alpha m}$,[20] where α is the probability of accepting an irrelevant feature as relevant.

Regularization. Another approach for *FS* is l_1 regularization. This method trims the hypothesis space by constraining the magnitudes of the parameters.[21] Regularization adds a penalty term to the least square function to prevent over-fitting.[22] The formulations of l_1 norm have the advantage of generating very sparse solutions while maintaining accuracy. The classifier-fitted parameters θ_i are multiplied by a coefficient λ to shrink them toward zero. This procedure effectively reduces the feature space and protects against over-fitting. Regularization methods may perform better than *FS* methods.[23]

Classifier design, selection, and assessment

A classifier is a supervised learning algorithm that analyzes the training data (e.g. data with classification class) and fits a model. The training dataset is used to train a set of candidate models using different tuning parameters.

It is important to choose an appropriate validation or cross-validation (*CV*) method to evaluate the generalization ability of each candidate model and select the *best*, according to a relevant model selection criterion.

For information-theoretic model selection approaches in the analysis of empirical data, refer to Peruggia.[24] Common performance metrics for model selection based on recognition rates—correct decisions made—can be found in Fawcett.[25]

For a data-rich analysis, the hold-out validation method is recommended, an approach in which a dataset is randomly divided into three subsets: training, validation, and testing. As an heuristic, 50% of the initial dataset is allocated to training, 25% to validation, and 25% to testing.[26]

Once the best candidate model has been selected, it is recommended that the model's generalization performance be tested on a new dataset before the model is deployed. This can also determine whether the model satisfies the learning requirement.[26] The generalization performance can be efficiently evaluated using a *confusion matrix* (*CM*).

CM. In predictive analytics, a *CM*[25] is a table with two rows and two columns that reports the number of *false positives* (*FPs*), *false negatives* (*FNs*), *true positives* (*TPs*), and *true negatives* (*TNs*). This allows more detailed analysis than just the proportion of correct guesses since it is sensitive to the recognition rate by class.

A type I error (α) may be compared with a *FP* prediction; a type II (β) error may be compared with a false *FN*.[27] They are estimated by

$$\alpha = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{1}$$

$$\beta = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}} \tag{2}$$

LR

LR, which uses a transformation of the values of a linear function, is widely used in classification problems. It is an unconstrained convex problem with a continuous differentiable objective function that can be solved either by the Newton's method or the conjugate gradient. *LR* models the probability distribution of the class label y, given a feature vector x[28]

$$P(y=1|x;\theta) = \sigma(\theta^T x) = \frac{1}{1+\exp(-\theta^T x)} \tag{3}$$

where $\theta \in \mathbb{R}^{\mathbb{N}}$ are the parameters of the *LR* model and $\sigma(\cdot)$ is the sigmoid function (logistic function) that maps values from $(-\infty,\infty)$ to $[0,1]$. The discrimination function itself is not linear, but the decision boundary is still linear.

The most common approach to estimate the parameters of a statistical model is to compute the maximum likelihood estimate (MLE). The problem of finding the MLE of the parameters θ for the unregularized LR can be defined by in terms of the negative log-likelihood (NLL)

$$\min_{\theta} \sum_{i=1}^{M} -\log p\left(y^{(i)}|\mathbf{x}^{(i)};\theta\right) \tag{4}$$

The NLL for LR is

$$\mathrm{NLL} = -\sum_{i=1}^{M}\left[y^{(i)}log\mu^{(i)} + \left(1-y^{(i)}\right)log\left(1-\mu^{(i)}\right)\right] \tag{5}$$

where $\mu^{(i)} = sigm(\theta^T x^{(i)})$. It is also called the *cross-entropy error* (*CEE*) function.[29]

Under the Laplacian prior $p(\theta) = (\lambda/2)^N \exp(-\lambda||\theta||_1)(\lambda>0)$, the *maximum a posteriori* (*MAP*) estimate of the parameters θ is

$$\min_{\theta}\sum_{i=1}^{M} -\log p\left(y^{(i)}|\mathbf{x}^{(i)};\theta\right) + \lambda||\theta||_1 \tag{6}$$

This optimization problem is referred to as l_1-regularized *LR*. This algorithm is widely applied in problems with small training sets or with high-dimensional input space. However, adding the l_1 regularization makes the optimization problem computationally more expensive. For solving the l_1-regularized *LR*,[30] the *least absolute shrinkage and selection operator* (*LASSO*) is an efficient method.

As the value of λ increases, the number of features included in the model decreases. The higher the value of λ, the lower the chance of over-fitting with too many redundant or irrelevant variables. The value of λ can be tuned through validation or *CV*.

In general, high correlations among features may hamper the LASSO in finding the true model. LASSO may not be able to distinguish true features with any amount of data and any amount of regularization.[31] Therefore, eliminating highly correlated features is one of the main challenges.

ISCS

ISCSs are computer-based decision support systems that incorporate a variety of artificial intelligence (*AI*) and non-AI techniques to monitor, control, and diagnose process variables to assist operators with the tasks of monitoring, detecting, and diagnosing process anomalies or in taking appropriate actions to control processes.[32]

There are three general solution approaches for supporting the tasks of monitoring, control, and diagnosis: (1) data driven, for which the most popular techniques are principal component analysis, Fisher discriminant analysis, and partial least-squares analysis; (2) analytical, an approach founded on first principles or other mathematical models; and (3) *KB* founded on *AI*, specifically expert systems, fuzzy logic, *ML*, and *PR*.[32,33]

Due to the explosion of industrial big data, KB ISCSs have received great attention. Since the scale of the data generated from manufacturing systems cannot be efficiently managed by traditional process monitoring and quality control methods, a *KB* scheme might be an advantageous approach.

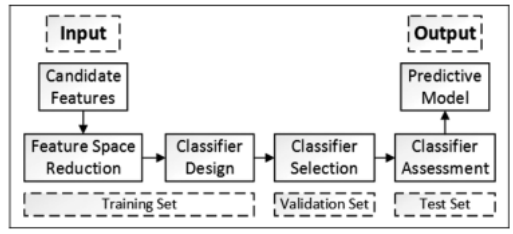


Figure 1. Learning process and pattern recognition framework.

LP and PR strategy

The proposed *LP* and *PR* strategy for a *KB ISCS* considers the l_1-regularized *LR* as the learning algorithm. Figure 1 displays the proposed strategy, Because manufacturing systems tend to be time dependent, a time-ordered hould-out data partition method should be considered (framed into a four-stage approach). The input is a set of candidate features, and the outcome is a parsimonious predictive model that contains the most relevant features to the quality of the product. This model is used to detect rare quality events in manufacturing systems. The candidate features can be derived from sensor signals following typical feature construction techniques[34] or from process physical knowledge. Due to the dynamic nature of manufacturing systems, the predictive model should be updated constantly to maintain its generalization ability.

A total of three main conditions that must be satisfied are (1) the faulty events must be generated during the manufacturing process and captured by the signals; (2) since the *LR* learning algorithm is a linear classifier, the decision boundaries between the two classes must be linear; and (3) in order for the binary classifier to properly define the classification boundary, the two classes should be well characterized, if the one class is unlabeled, not present, or not properly sampled, a one class classification—novelty detection—approach could be considered.[35–37] However, novelty detection is out of the scope of this article.

In the following subsection, the LP is presented. In which three of the most critical challenges posed by the l_1-regularized *LR* algorithm are addressed: (1) high correlations, (2) finding the classification threshold, and (3) tuning the penalty value λ (classifier selection).

LP

The first step is to eliminate irrelevant and redundant features from the analysis. For manufacturing processes, massive amounts of data and the lack of a

comprehensive physical understanding may result in the development of many irrelevant and redundant features. This problem representation highlights the importance of preprocessing the data.

The feature space reduction is performed in a two-step approach: (1) irrelevant feature elimination, in which the *ReliefF* algorithm is used to obtain the feature ranking, and the associated weight of each feature is compared with τ to eliminate the irrelevant ones, and (2) redundant feature elimination, based on a new *hybrid correlation and ranking-based* (*HCR*) algorithm. The proposed algorithm (Appendix 1) eliminates redundant features based on Pearson's correlation coefficients and a feature-ranking algorithm. The basic idea is to keep the *best* feature—highest ranked—from a set of two or more highly correlated variables. The *HCR* algorithm is a data preprosessing tool for classification problems that is simple and fast to execute.

Once the feature space has been reduced, the following step is to design the classifier and to identify which features contain the most relevant information to the quality of the product. While the classifier is aimed to detect rare quality events, the features included in the predictive model may provide valuable engineering information. Although feature interpretation is out of the scope of this approach, analyzing the data-derived predictive model from a physics perspective may support engineers in systematically discovering hidden patterns and unknown correlations that may guide them to identify root causes and solve quality problems.

The training set is used to fit n-candidate l_1-regularized *LR* models by varying the penalty value λ. It is recommended to start with the largest value of λ that gives a nonnull model (i.e. a model with the intercept only), and from that point decrease the value of λ to develop more candidate models with more features. The rationale behind this approach is that the form of the model is not known in advance; therefore, it can be approximated by generating a set of candidate models. This analysis can be computationally performed using the *LASSO* method in MATLAB or R.

Since faulty events rarely occur in manufacturing, the dataset is highly unbalanced. Therefore, the 0.5 threshold may not be the best classification threshold, and accuracy[25] may be a misleading indicator of classification performance.

To address this scenario, the concept of *maximum probability of correct decision* (*MPCD*) is used as a measure of generalization performance.[38,39] A model selection criterion tends to be very sensitive to FNs—failure to detect a quality event—in highly unbalanced data. *MPCD* is estimated by

$$\mathrm{MPCD} = (1-\alpha)(1-\beta) \tag{7}$$

Since MPCD is used as a model selection criterion, the optimal classification threshold search—with respect to MPCD—algorithm (OCTM) is developed (Appendix 2) aimed at obtaining the classification threshold. The algorithm enumerates all candidate solutions—candidate classification thresholds—and selects the one with the highest estimated *MPCD*. Candidate solutions are the mid-point values (logistic function-based conditional probabilities) between two consecutive examples.

In the context of *PR*, the primary purpose is to select the *best* candidate model with respect to generalization. Once n-candidate models have been developed, the validation dataset is used to estimate the *MPCD* of each candidate model, and the model with the highest value should be selected. In addition to *MPCD*, sparsity and *CEE* should be used as a second-level model selection criteria.

It is recommended to perform bias–variance analysis using the *CEE* to ensure that the selected model does not exhibit under-fitting or over-fitting problems.[26]

Finally, the generalization performance of the selected model is evaluated on the testing set. The classifier must be assessed without the bias induced in the validation stage. This stage ensures that the model satisfies the learning target for the project at hand.

Discussion

Although no algorithm can guarantee the best answer,[40] parsimonious modeling plays an important role in manufacturing, since model interpretation is performed to understand the system. Specifically, the l_1-regularized *LR* algorithm enjoys the following desirable properties: (1) It induces parsimony while maintaining convexity;[41] (2) it is founded on the likelihood principle, maximum likelihood provides a consistent approach to parameter estimation problems and has desirable mathematical and optimality properties;[42] (3) according to large sample theory, as the sample size tends to infinity, the sampling distribution of the MLE becomes Gaussian;[29] and (4) since many candidate models are created to approximate the true model, well-known likelihood-based model selection criterion (Akaike information criterion (AIC) or Bayesian information criterion (BIC)) can be applied (and compared) to solve the challenge posed by over-fitting due to model complexity.

In this article, the main challenges of the l_1-regularized *LR* algorithm are discussed and approached. However, the proposed framework could be generalized to other regularized algorithms (e.g. support vector machine), in which a tuning parameter procedure should be followed to induce parsimony and improve generalization.[43,44]

Case studies

Two automotive case studies are presented.

UMW

UMW is a solid-state bonding process that uses high-frequency ultrasonic vibration energy to generate oscillating shears between metal sheets clamped under pressure. It is an ideal process for bonding conductive materials such as copper, aluminum, brass, gold, and silver and for joining dissimilar materials. Recently, it has been adopted for battery tab joining in the manufacturing of vehicle battery packs. Creating reliable joints between battery tabs is critical because one low-quality connection may cause performance degradation or the failure of the entire battery pack. It is important to evaluate the quality of all joints prior to connecting the modules and assembling the battery pack.[16]

The data used for this analysis are derived from the *UMW* of battery tabs for the Chevrolet Volt,[38] an extended range electric vehicle. It is a very stable process that only generates a few defective welds per million of opportunities. However, all the welds in the battery must be good for the electric motor to function. This problem representation not only highlights the engineering intellectual challenge but also the importance of a zero-defect policy.

The collected dataset contains a binary outcome (*good*/*bad*) with 54 features derived from signals (e.g. acoustics, power, and linear variable differential transformers) following typical feature construction techniques.[34] The dataset is highly unbalanced since it contains only 36 bad batteries out of 40,000 examples (0.09%). The dataset is partitioned following the hold-out validation scheme (including *bads* in each dataset): training set (20,000), validation set (10,000), and testing set (10,000).

Feature space reduction. To eliminate irrelevant features, the dataset is initially preprocessed using the *ReliefF* algorithm. *ReliefF* is run with $k = 5$ nearest neighbors and a significance threshold of $\tau = 0.031622$ (calculated based on $1/\sqrt{\alpha m}$—$\alpha = 0.05$ and $m = 20,000$). According to the algorithm, feature 26 is the most important feature, while feature 14 is the lowest quality feature. Figure 2 summarizes the feature ranking and which features are selected based on τ. According to *ReliefF*, 45 features—out of 54—should be selected.

Redundant features from the obtained subset by *ReliefF* were eliminated by *HCR* algorithm ($\delta = 0.90$). The algorithm eliminated 13 highly correlated features. The feature space was reduced to 32 relevant variables without "high correlations."

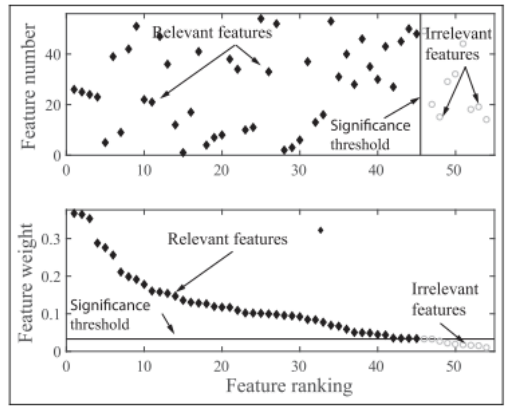


Figure 2. Feature ranking and selection using *ReliefF*.

Classifier design. The training set was used to fit 100 regularized *LR* models. The *LASSO* method was applied to estimate the fitted least-squares regression coefficients for a set of 100 regularization coefficients λ, starting with the largest value of λ that gives a nonnull model. However, the nonnull model is not included in the analysis since its estimated *MPCD* equals zero. Figure 3(a) displays each candidate model's associated value of λ, Figure 3(b) the number of features, Figure 3(c) the associated values of γ, and Figure 3(d) displays the training error (e.g. *CEE*). The number of features decreases as the value of λ increases, Figure 3(a) and (b). Selecting the right model is one of the main challenges.

OCTM. Figure 4 shows the OCTM search of candidate model 88.

Classifier selection. The goal is to select the candidate model with the highest *MPCD*. In the context of the problem that is being solved, the goal is to detect low-quality welds. Due to the relevance of failing to detect a potential defect, the type II error is the main concern of this analysis; for this reason, the *MPCD* is also used as a model selection criteria. The estimated *MPCD*, α, β, and validation error of each model are summarized in Figure 5.

According to the selection criteria, model 88 is the best candidate, with an estimated *MPCD* of 0.8805 ($\alpha = 0.0095$, $\beta = 0.1111$) and four relevant features, and varying the values of λ helped to identify the most relevant features. The coefficients are shown in Table 1. The value of γ for this model is 0.0063, meaning that any value estimated by the logistic function below this threshold will be classified as 0 (i.e. *good*) or 1 (i.e. *bad*) otherwise.

According to the bias–variance analysis, Figures 3(d) and 5(d), the first candidate models (i.e. 1–60) exhibited

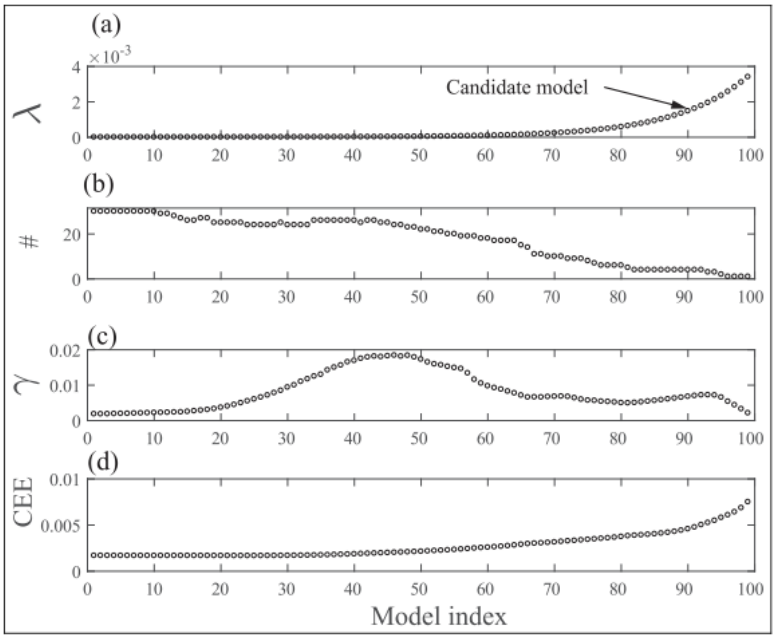


Figure 3. Candidate model information: (a) values of λ, (b) number of features, (c) optimal classification thresholds, and (d) training CEE.

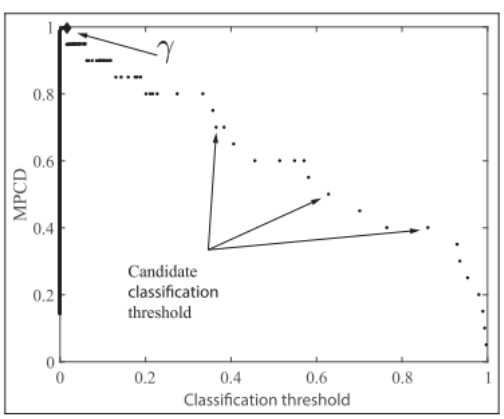


Figure 4. Optimal classification threshold search of candidate model 88.

over-fitting problems, while the last models (i.e. 91–99) exhibited under-fitting problems. Therefore, the bias–variance trade-off is efficiently overcome by this parsimonious candidate model.

A receiver operating characteristic (ROC) plot for model comparison efficiently depicts relative trade-offs between *TP* and *FP*. The best possible prediction method would be a point in the upper left corner, or coordinate 0, 1 of the *ROC* space; it would be a perfect classification. The location of the chosen model in the *ROC* plot confirms that model 88 is the best candidate, and it has the smallest estimated α from the set of candidate models with the same estimated β. Figure 6 illustrates the relative location of the model 88 in the *ROC* curve.

Classifier assessment. The importance of this final step is to assess the classifier without the induced bias in the validation stage and to ensure the model satisfies the learning target. The estimated *MPCD* of the final model on the testing data is 0.9980 ($\beta = 0$, $\alpha = 0.0020$). The testing set includes 10,000 records, with seven bad batteries. The classifier correctly classified the 7 bad units and only misclassified 20 good units. Recognition rates are summarized in Table 2.

According to model assessment results, *LR* not only shows high prediction ability but also did not commit any type II error. The graphical representation of the classification using unseen data (i.e. testing set) is shown in Figure 7.

LSW

To show the reproducibility and flexibility of the proposal, the same *LP* and *PR* strategy is applied to a balanced dataset, derived from an *LSW* process:

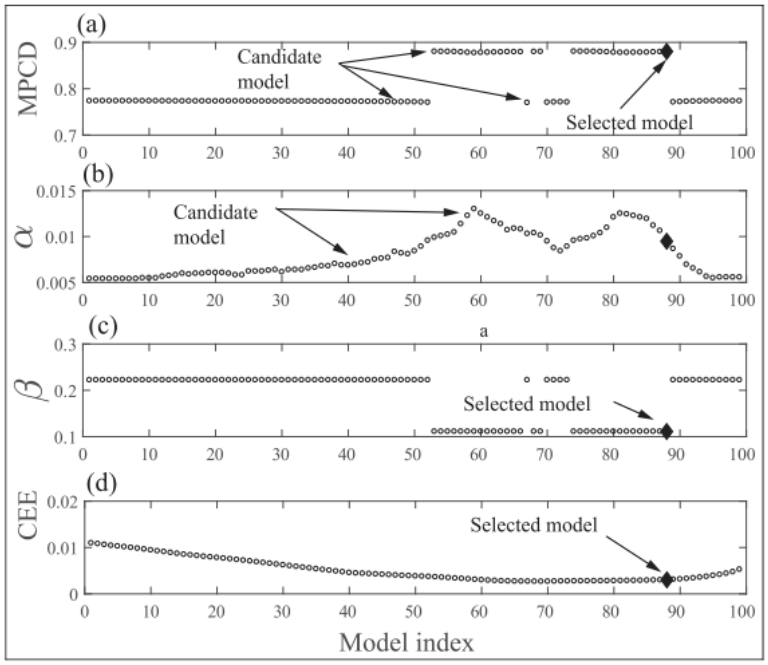


Figure 5. Generalization performance of candidate models: (a) validation MPCD, (b) validation α, (c) validation β, and (d) validation CEE.

Table 1. Coefficients of model 88.

Coefficient	Value	Coefficient	Value	Coefficient	Value
θ_0	−17.2305	θ_5	−0.0046	θ_9	31.4627
θ_{22}	8.8622	θ_{26}	0.000995		

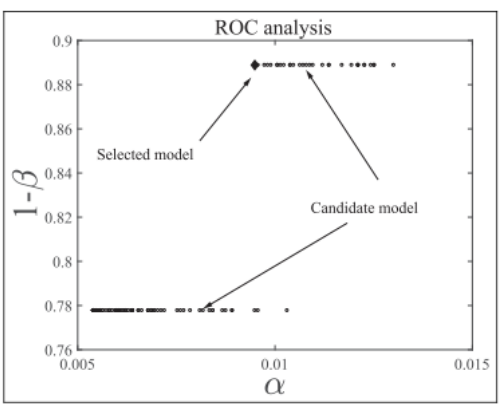


Figure 6. *ROC* curve of the candidate models.

Table 2. Confusion matrix.

	Declare good	Declare bad
Good	9973	20
Bad	0	7

Laser welding is a welding technique used to join multiple pieces of metal through the use of a laser beam. The laser welding system provides a concentrated heat source, allowing for narrow, deep welds and high welding rates. This process is used frequently in high volume welding applications, such as in the automotive industry. Laser welding in the automotive industry has applications that enable manufacturers to weld component engine parts, transmission parts, alternators, solenoids, fuel injectors, fuel filters, air conditioning equipment, and air bags, as well as many other applications.[45]

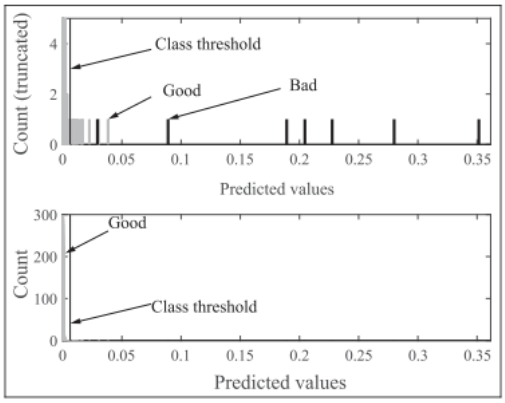


Figure 7. *LR*-based classification.

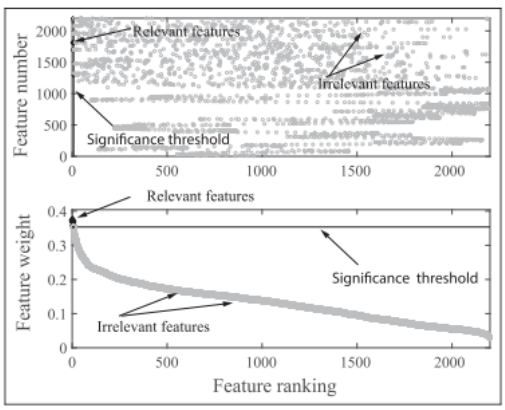


Figure 8. Feature ranking and selection using *Relief F*.

The *LSW* process is often completed in few milliseconds, it exhibits good repeatability and is easy to automate. It is an excellent option for high-productivity processes.

The dataset contains 2199 features and 317 examples (159 good, 158 bad), and it is partitioned following the hold-out validation scheme: training set (160), validation set (80), and test set (77). To maintain space efficiency, only the most relevant plots are included in this analysis.

Since the included number of bad in this training set is significantly higher than the *UMW* dataset, the *ReliefF* algorithm is run with $k = 10$, with a significance threshold of $\tau = 0.3535$. According to the *ReliefF* algorithm, feature 1812 is the most important feature, while feature 2190 is the feature with the least discriminative information. Figure 8 summarizes the feature ranking and which features are selected based on τ. According to *ReliefF*, only 13 features—out of 2199—should be selected.

Redundant features from the subset obtained by *ReliefF* are eliminated by *HCR* algorithm using $\delta = 0.90$. The algorithm eliminated nine highly correlated features. Ultimately, the feature space was reduced to four relevant variables without high correlations. Then, the l_1-regularized *LR* algorithm was used to develop 93 candidate model. Figure 9(a)–(d) shows the most relevant information (e.g. λ, number of features, γ, training *CEE*, respectively) of each candidate model.

Since there are seven candidate models that perfectly separate the data 87–93, Figure 10(a), the number of features and the validation *CEE* are used as a secondary model selection criteria. Since models 90–93 contain only one feature, model 90 is chosen, since it is the candidate model with the smallest validation *CEE*, Figure 10(d). Coefficients are shown in Table 3, and its associated classification threshold is $\gamma = 0.4375$, Figure 9(c).

The selected model perfectly separated *good* from *bad* welds in the testing set. Recognition rates are summarized in Table 4 and graphically displayed in Figure 11.

Comparative analysis

To evaluate the performance of the proposal, a comparative analysis is performed. The results of the two case studies were compared with a typical modeling analysis. The same learning algorithm was trained (with the same values of λ) without preprocessing the data and using widely known model selection approaches—*CEE, AIC*, and *BIC*.[24]

Models were mainly compared based on their detection capacity with the smallest α error possible; in addition, parsimony was also considered. Due to space constraints, only the most relevant graphs are presented.

UMW

Following the same data partition strategy, the training set is used to create the set of candidate models and to estimate the *AIC* and *BIC* scores. The associated number of features and the values of γ (obtained using the *OCTM* algorithm) of each candidate model are displayed in Figure 12. While the validation set is used to estimate the *MPCD* and *CEE* of each model. Model selection results are summarized in Figure 13.

According to the *AIC-BIC*, candidate model 81 should be selected ($AIC = 76.8137, BIC = 105.8712$), and this candidate model contains five features with an estimated *MPCD* of 0.8778. While the *CEE* criterion recommends model 69 ($CEE = 0.0025$), and this model contains 16 features with an estimated *MPCD* of 0.7719. Table 5 summarizes the generalization

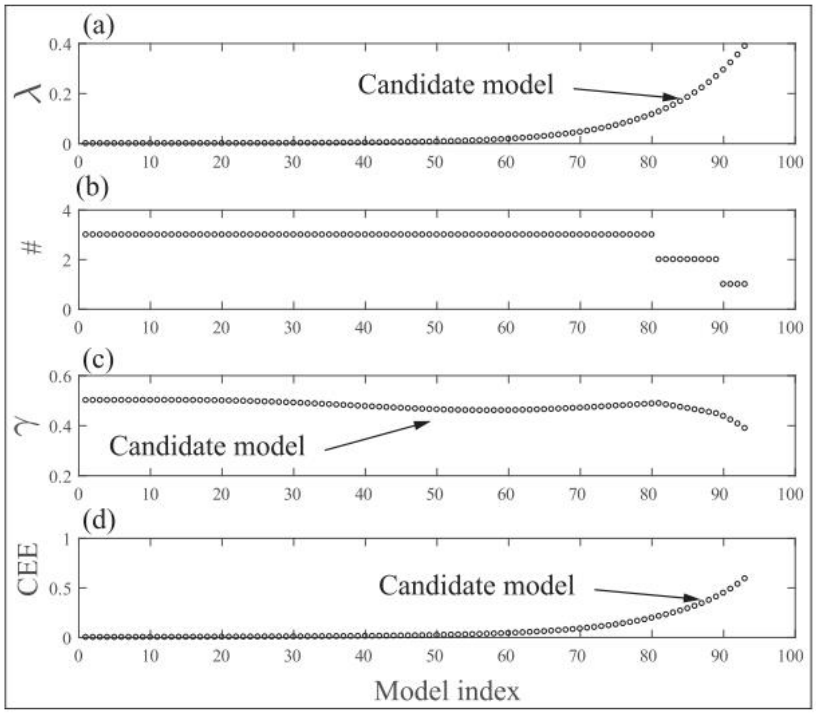


Figure 9. Candidate model information: (a) values of λ, (b) number of features, (c) optical classification threshold, and (d) training CEE.

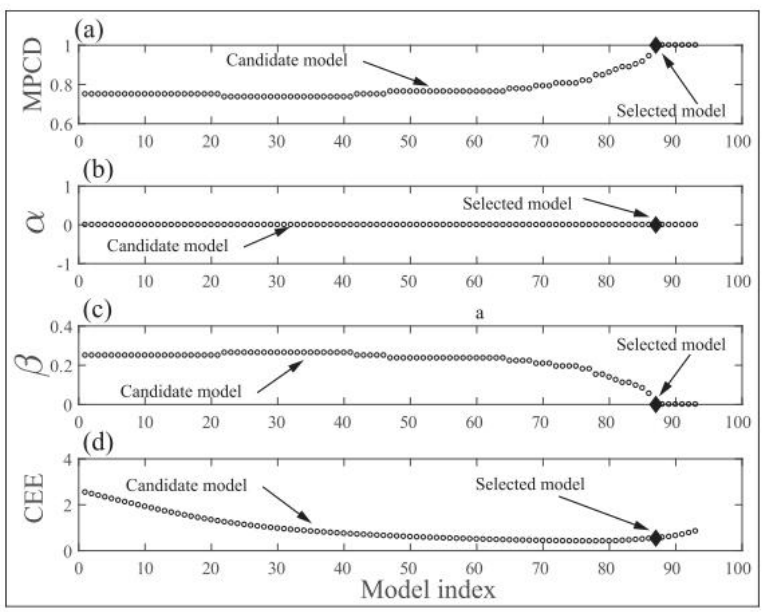


Figure 10. Generalization performance of candidate models: (a) validation MPCD, (b) validation α, (c) validation β, and (d) validation CEE.

Table 3. Coefficients of model 90.

Coefficient	Value	Coefficient	Value
θ_0	2.3689	θ_{1812}	−3.2885

Table 4. Confusion matrix.

	Declare good	Declare bad
Good	50	0
Bad	0	27

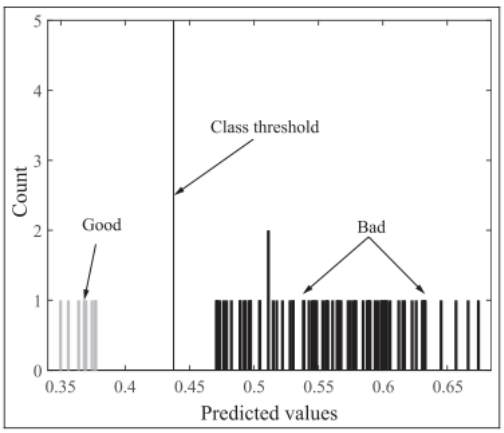


Figure 11. LR-based classification.

performance in the testing set of the three selected models developed in this section and the final model from the *UMW* case study (e.g. final).

The three models correctly classified the seven *bad* units in the testing set with a very small α. However, the final model contains only 4 features, while models 81 and 69 contain 5 and 16, respectively. From engineering perspective, it is significantly easier to interpret a model with 4 features than a model with 5 or 16.

LSW

Candidate model information is summarized in Figure 14, while the model selection criterion values are summarized in Figure 15.

According to the *AIC*, candidate model 39 should be selected ($AIC = 72.2325$), and this candidate model contains 11 features with an estimated *MPCD* of 0.8194. While the *BIC* recommends candidate model 83 ($BIC = 85.4445$) with only two features with an estimated *MPCD* of 1. Finally, the *CEE* criterion recommends model 35 ($CEE = 0.2869$), and this model

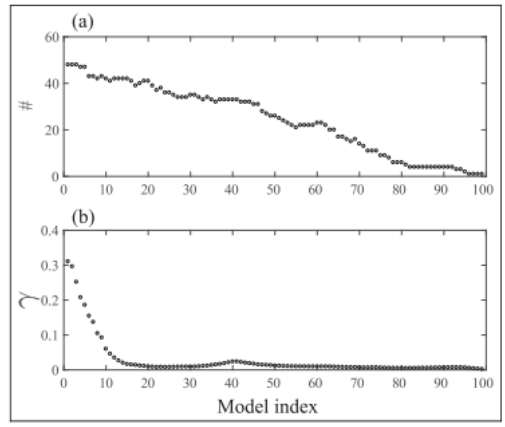


Figure 12. Candidate model information: (a) number of features and (b) optimal classification threshold.

contains 14 features with an estimated *MPCD* of 0.8056. The generalization performance in the testing set is summarized in Table 6.

In this case study, the final model outperforms the three models, although model 83 perfectly separates the classes, and this model contains two features. However, models 39 and 35 have many features and also failed to detect all the *bad* units; therefore, the *MPCD* is significantly lower.

Discussion

Based on the comparative analysis, the models developed following the proposed *LP* and *PR* strategy exhibited better parsimony properties and good (or even better) detection capacity when compared with a typical l_1-regularized *LR* analysis with three popular model selection criterion (e.g. *AIC*, *BIC*, and *CEE*).

Although l_1-regularized *LR* learning algorithm induces sparsity, the proposed strategy can boost the learning algorithm by eliminating irrelevant and redundant features.

The same approach is also being applied to different automotive manufacturing systems with promising results; however due to space constraints, they are not discussed in this article.

Conclusion

Today's business environment sustains mainly those companies committed to a zero-defect policy. This quality challenge was the main driver of this research, where an *LP* and *PR* strategy was developed for a *KB ISCS*. The proposed approach was aimed at detecting rare quality events in manufacturing systems and to identify the most relevant features to the quality of the product. The defect detection was formulated as a

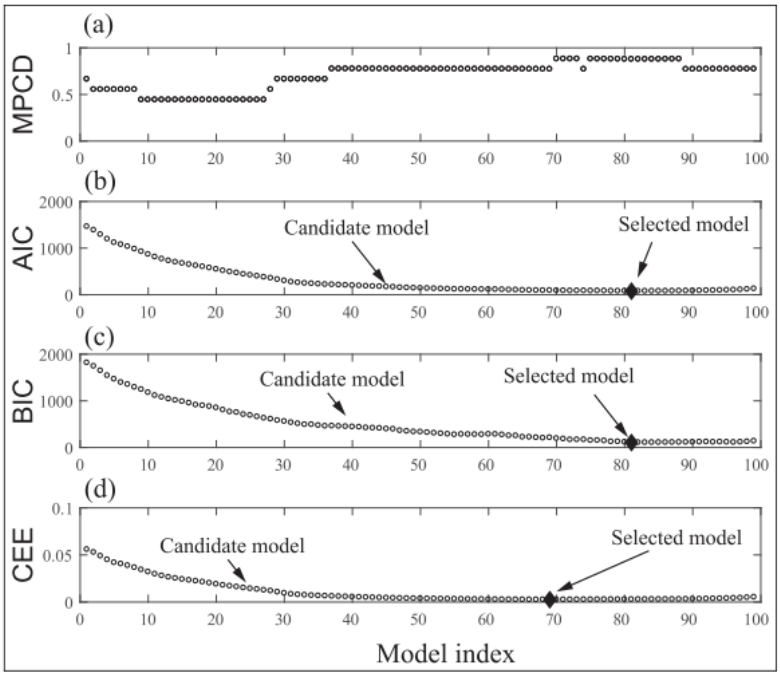


Figure 13. Model selection approaches: (a) MPCD, (b) AIC model selection criterion, (c) BIC model selection criterion, and (d) CEE model selection criterion.

Table 5. Generalization analysis of the selected models.

Model	Features	FN	FP	TN	TP	MPCD
Final	4	0	20	9973	7	0.9980
Model 81	5	0	24	9969	7	0.9976
Model 69	16	0	14	9979	7	0.9986

FN: false negative; FP: false positive; TN: true negative; TP: true positive; MPCD: maximum probability of correct decision.

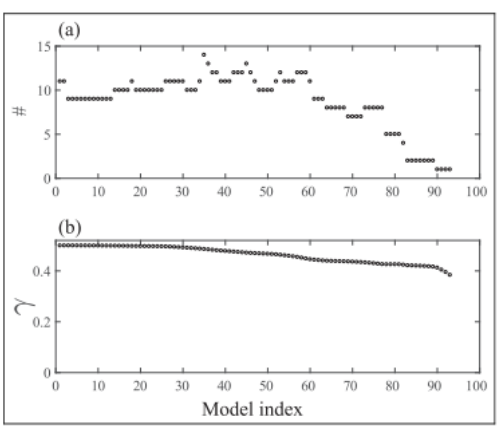


Figure 14. Candidate model information: (a) number of features and (b) optimal classification threshold.

binary classification problem and validated in two experimental datasets derived from automotive manufacturing systems: (1) *UMW* of battery tabs from a battery assembly process and (2) *LSW* sub-assembly components from an assembly process. In both cases, the main objective was to detect low-quality welds (bad) from the process.

To increase the classifier prediction ability and reduce training times, the dataset was preprocessed in a two-step approach: (1) the *ReliefF* algorithm was used to eliminate irrelevant features, and (2) the *HCR* algorithm was applied to eliminate redundant features that most filter methods cannot eliminate.

The l_1-regularized *LR* was used as the learning algorithm for the classification task and to identify the most important features. Since the form of the model was not known in advance, a set of candidate models was developed—by varying the value of λ—as an effort to

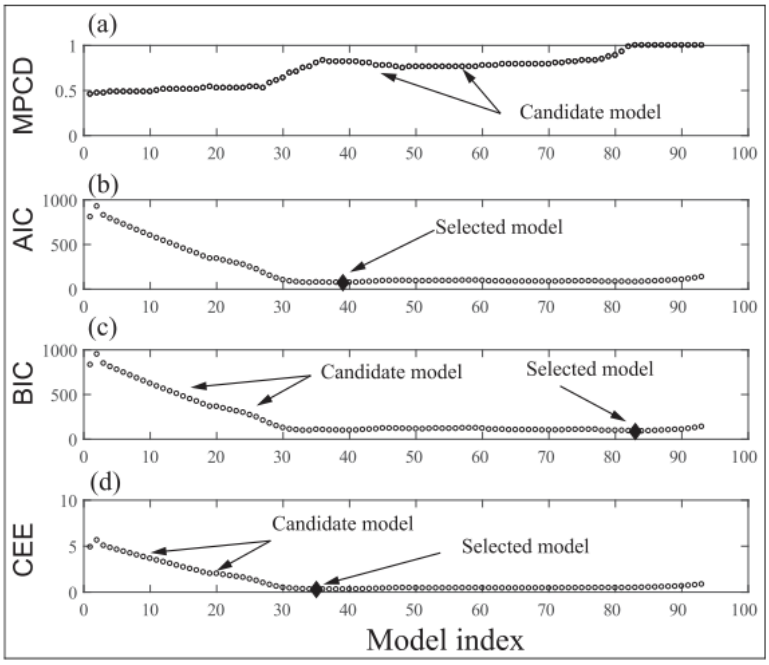


Figure 15. Model selection approaches: (a) MPCD, (b) AIC model selection criterion, (c) BIC model selection criterion, and (d) CEE model selection criterion.

Table 6. Generalization analysis of the selected models.

Model	Features	FN	FP	TN	TP	MPCD
Final	1	0	0	50	27	1
Model 39	11	5	0	50	22	0.8148
Model 83	2	0	0	50	27	1
Model 35	14	6	0	50	21	0.7778

FN: false negative; FP: false positive; TN: true negative; TP: true positive; MPCD: maximum probability of correct decision.

approximate the true model. Chosen model exhibited high capacity to detect rare quality events, since 100% of the defective units on the testing set were detected.

The proposed strategy used the *MPCD* as a model selection criterion. Therefore, the *OCTM* algorithm was developed to find γ, the optimal classification threshold with respect to *MPCD*.

The proposed approach can be adapted and widely applied to manufacturing processes to boost the performance of traditional quality methods and potentially move quality standards forward, where soon virtually no defective product will reach the market.

Future work

Since *MPCD* is founded exclusively on recognition rates, future research along this path could focus on adding a penalty term for model complexity. Although information-theoretic approaches such as *AIC* and *BIC* penalize for model complexity, they are not mainly founded on recognition rates.

Acknowledgements

We would like to express our deepest appreciation to Dr Debejyo Chakraborty, Diana Wegner, and Dr Xianfeng Hu, who helped us to complete this report. A special gratitude is given to Dr Jeffrey Abell, whose ideas and contributions illuminated this research.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially supported by the Consejo Nacional de Ciencia y Tecnologia (CONACYT; under grant 404325/215143).

References

1. AS of Quality. *Emergence—2011 future of quality study*. Milwaukee, WI: ASQ: The Global Voice of Quality, 2011.
2. Schwab K. The fourth industrial revolution: what it means, how to respond. *World Economic Forum*, 2016, https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/
3. Yin S and Kaynak O. Big data for modern industry: challenges and trends. *Proc IEEE* 2015; 103: 143–146.
4. Yin S, Li X, Gao H, et al. Data-based techniques focused on modern industry: an overview. *IEEE T Ind Electron* 2015; 62: 657–667.
5. Venkatasubramanian V, Rengaswamy R, Kavuri S, et al. A review of process fault detection and diagnosis: part III: process history based methods. *Comput Chem Eng* 2003; 27: 327–346.
6. Escobar CA and Morales-Menendez R. Machine learning and pattern recognition techniques for information extraction to improve production control and design decisions. In: *Proceedings of the advances in data mining, ICDM 2017* (ed P Perner), New York, 12–13 July 2017, pp.285–295, Berlin: Springer.
7. Ghosh P. A comparative roundup: artificial intelligence vs. machine learning vs. deep learning, June 2016, www.dataversity.net/ai-vs-machine-learning-vs-deep-learning
8. Theodoridis S and Koutroumbas K. Pattern recognition and neural networks. In: Paliouras G, Karkaletsis V and Spyropoulos CD (eds) *Machine learning and its applications*. Berlin: Springer, 2001, pp.169–195.
9. Zhou Z. Ensemble learning. In: Li SZ and Jain AK (eds) *Encyclopedia of biometrics*. Berlin: Springer, 2009, pp.270–273.
10. Bishop C. *Pattern recognition and machine learning*. Berlin: Springer, 2006.
11. Bradley P and Mangasarian O. Feature selection via concave minimization and support vector machines. In: *Proceedings of the machine learning 15th international conference*, 24–27 July, 1998, pp.82–90. San Francisco, CA: Morgan Kaufmann.
12. Yu L and Liu H. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: T. Fawcett, & N. Mishra (eds) *Proceedings of the 20th international conference on machine learning*, Washington, DC, 21–24 August 2003, pp.856–863.
13. Hall M. Correlation-based feature selection of discrete and numeric class machine learning. In: *Proceedings of the 17th international conference on machine learning*, 29 June–2 July 2000, pp.359–366. Hamilton, New Zealand: University of Waikato.
14. Nicodemus K and Malley J. Predictor correlation impacts machine learning algorithms: implications for genomic studies. *Bioinformatics* 2009; 25: 1884–1890.
15. Wang F, Yang Y, Lv X, et al. Feature selection using feature ranking, correlation analysis and chaotic binary particle swarm optimization. In: *Proceedings 5th IEEE international conference on software engineering and service science*, Beijing, China, 23 October 2014, pp.305–309. New York: IEEE.
16. Shao C, Paynabar K, Kim T, et al. Feature selection for manufacturing process monitoring using cross-validation. *J Manuf Syst* 2013; 32: 550–555.
17. Wu S, Hu Y, Wang W, et al. Application of global optimization methods for feature selection and machine learning. *Math Probl Eng*. Epub ahead of print 2 September 2013. DOI: 10.1155/2013/241517.
18. Kira K and Rendell L. The feature selection problem: traditional methods and a new algorithm. In: *Proceedings of the 10th national conference on artificial intelligence*, San Jose, CA, 12–16 July 1992, vol. 2, pp.129–134. New York: ACM.
19. Chandrashekar G and Sahin F. A survey on feature selection methods. *Comput Electr Eng* 2014; 40: 16–28.
20. Robnik-Šikonja M and Kononenko I. Theoretical and empirical analysis of ReliefF and RReliefF. *Mach Learn* 2003; 53: 23–69.
21. Bishop C. *Neural networks for pattern recognition*. Oxford: Oxford University Press, 1995.
22. Ng A. Feature selection L1 vs L2 regularization and rotational invariance. In: *Proceedings of the 21st international conference on Machine learning*, Banff, AB, Canada, 4–8 July 2004, p. 78. New York: ACM.
23. Xing E, Jordan M and Karp R. Feature selection for high-dimensional genomic microarray data. In: *Proceedings of the 18th international conference on machine learning ICML*, Williamstown, MA, 28 June–01 July 2001, vol. 1, pp.601–608. San Francisco, CA: Morgan Kaufmann.
24. Peruggia M. Model selection and multimodel inference: a practical information-theoretic approach. *J Am Stat Assoc* 2003; 98: 778–779.
25. Fawcett T. An introduction to ROC analysis. *Pattern Recog Lett* 2006; 27: 861–874.
26. Friedman J, Hastie T and Tibshirani R. *The elements of statistical learning*, vol. 1: Berlin: Springer, 2001.
27. Devore J. *Probability and statistics for engineering and the sciences*. Boston, MA: Cengage Learning, 2015.
28. Lee S, Lee H, Abbeel P, et al. Efficient L1 regularized logistic regression. In: *Proceedings of the national conference on artificial intelligence*, Cambridge, MA, 16–20 July 2006, p. 401. Reston, VA: AIAA.
29. Murphy K. *Machine learning: a probabilistic perspective*. Cambridge, MA: The MIT Press, 2012.
30. Tibshirani R. Regression shrinkage and selection via the Lasso. *J Roy Stat Soc Ser B Met* 1996; 58: 267–288.
31. Zhao P and Yu B. On model selection consistency of Lasso. *J Mach Learn Res* 2006; 7: 2541–2563.
32. Uraikul V, Chan W and Tontiwachwuthikul P. Artificial intelligence for monitoring and supervisory control of process systems. *Eng Appl Artif Intel* 2007; 20: 115–131.

33. Chiang L, Braatz R and Russell E. *Fault detection and diagnosis in industrial systems*. Berlin: Springer Science + Business Media, 2001.
34. Huan L and Motoda H. *Feature extraction, construction and selection: a data mining perspective*. Berlin: Springer Science + Business Media, 1998.
35. Luiz H, Lorena N, André C, et al. Filter feature selection for one-class classification. *J Intell Robot Syst* 2015; 80: 227–243.
36. Khan SS and Madden MG. A survey of recent trends in one class classification. In: *Proceedings of the Irish conference on artificial intelligence and cognitive science*, Dublin, Ireland, August 2009, pp.188–197. Berlin: Springer.
37. Manevitz LM and Yousef M. One-class SVMs for document classification. *J Mach Learn Res* 2001; 2: 139–154.
38. Abell JA, Chakraborty D, Escobar CA, et al. Big data driven manufacturing—process-monitoring-for-quality philosophy. *J Manuf Sci Eng*. Epub ahead of print 31 January 2017. DOI: 10.1115/1.4036833.
39. Abell JA, Spicer JP, Wincek MA, et al. Binary classification of items of interest in a repeatable process. Patent US8757469B2, June 2014, www.google.com/patents/US20130105556
40. Wolpert DH. The lack of a priori distinctions between learning algorithms. *Neural Comput* 1996; 8: 1341–1390.
41. Tibshirani RJ. *Statistical learning with sparsity: the lasso and generalizations*, vol. 79. Boca Raton, FL: CRC Press, 2014.
42. Natrella M. NIST/SEMATECH e-handbook of statistical methods, 2010, http://www.itl.nist.gov/div898/handbook/
43. Zhu J, Rosset S, Hastie T, et al. 1-norm support vector machines. *Adv Neur In* 2004; 16: 49–56.
44. Fung G and Mangasarian O. A feature selection Newton method for support vector machine classification. *Comput Optim Appl* 2004; 28: 185–202.
45. Subhajit R. Laser welding: a new dimension in automotive industry. *OEM Update*, December 2016, www.oemupdate.com/industry-update/*-industry

Appendix 1

The HCR algorithm has three components, Figure 16:

1. *Inputs*: F, list of features in descending order (i.e. top-ranked feature in column 1); FC, a feature pairwise correlation matrix; and *delta* (δ), the high-correlation threshold. To obtain F, it is necessary to rank the features according to their relevance to the target class. Once all features have been ranked, the ordered correlation matrix FC is obtained. δ is a user-specified threshold for a pair of features to be considered highly correlated.
2. *Initialization*: defines the three sets used by the algorithm: sorted and uncorrelated feature list, *SUFL*, which stores the features evaluated and selected by the algorithm; *EliminatedList*, which stores the highly correlated features that have been already evaluated and eliminated; and *TabuList*, which is the union of the first two lists. In addition, *TabuList* is used for the algorithm to check whether feature i has been previously evaluated (i.e. either selected or eliminated).
3. *Output*: $F_{reduced}$, subset of not highly correlated features sorted from highest ranking (line 14).

Inputs: $F(F1,\ F2, \ldots, Fn\ )$: *list of features ordered by descending (top ranked feature in column one)*
$FC(fc_{ij})_{nxn}$: *nxn ordered feature pairwise correlation matrix*
Delta: high − correlation threshold
Output: $F_{reduced}$: *subset of not highly − correlated features and sorted from highest ranking*
Initialization: *Set TabuList as empty,*
Set SUFL as empty
Set EliminatedList as empty

1. *for i = 1 to n do begin*
2. *find the features whose Correlation with feature i is larger than the threshold Delta and set the feature set as CorrFeat*
3. *if #(CorrFeat) == 1 and feature i is not in TabuList*
4. *add feature i to set SUFL and add feature i to TabuList*
5. *elseif #(CorrFeat) > 1 and feature i is not in TabuList*
6. *add feature i to set SUFL and*
7. *add features larger than i in corrFeat to EliminatedList and*
8. *keep uniqueness of the elements in EliminatedList and*
9. *set TabuList as the union of SUFL and EliminatedList*
10. *else*
11. *keep the three sets unchanged*
12. *end*
13. *end*
14. *return* $F_{reduced}$ *= SUFL*

Figure 16. Pseudo-code of the *HCR* algorithm.

The algorithm performs n iterations (lines 1, 2, 12, and 13) to find which features are highly correlated to feature i, and the *CorrFeat* variable is updated and evaluated at each iteration, with three possible scenarios: (1) when feature i does not have any correlated features and has not been previously evaluated, that feature is added to the *SUFL* and *TabuList* (lines 3 and 4); (2) when feature i has one or more highly correlated feature(s) and is not in the *TabuList*, that feature is added to the *SUFL*, while the highly correlated features larger than i are added to the *EliminatedList*, maintaining the uniqueness of the elements in the *EliminatedList* while updating the *TabuList* (lines 5–9); and (3) otherwise, the three sets are unchanged (lines 10 and 11).

Since the best values of k—for *ReliefF*—and δ—for *HCR*—are not known in advance, they can be tuned with respect to prediction. The value of k can be set based on the number of bad units in the training, and δ can be heuristically set and evaluated between 0.50 and 0.95.

Appendix 2

The *OCTM* algorithm has three components, Figure 17:

1. *Inputs*: *CP*, list of the conditional probabilities of each example—estimated using the logistic

```
Input: CP(CP_1, CP_2, ... CP_m):
  List of conditional probabilities ordered
Output: γ
  Optimal classification threshold
Initialization: set CCTL as empty
  List of MPCD values associated to each candidate classification threshold
    1.  For i = 1 to m − 1 do begin
    2.          CCT_i = (CP_i + CP_i+1)/2
    3.          estimate MPCD_i at each CCT_i
    4.          add MPCD_i to CCTL
    5.  end
    6.  Find p, the position of the max (CCTL)
    7.  return γ = CCT_p
```

Figure 17. Pseudo-code of the *OCTM* algorithm.

function—ordered by either ascending or descending.

2. *Initialization*: defines the vector *CCTL* that stores the estimated *MPCD* values associated with each candidate classification threshold.
3. *Output*: γ.

The algorithm performs $m - 1$ iterations (lines 1 and 5) to find the candidate classification thresholds (line 2), CCT_i, $i = 1, \ldots, m - 1$. The $MPCD_i$ is estimated at each CCT_i (line 3). The candidate classification threshold list is $CCTL = \{MPCD_i\}_{i=1}^{m-1}$ (line 4). Find the position p of the maximum value of *CCTL*. Finally, γ is the value of CCT_p (lines 6 and 7).

Chapter 3

Process-Monitoring-for-Quality – A Model Selection Criterion

Due to the “no free lunch” principle, there is no a priori distinction between learning algorithms; therefore, many different *MLA* should be trained and tested to solve the pattern classification problem effectively. In this context, *MS* becomes a critical aspect to develop a manufacturing-functional model for defect detection.

The new *IBD* driven manufacturing quality philosophy, *PMQ*, proposes *BDBM*, a new modeling paradigm that includes a learning process that often requires many models (classifiers) to be created to find the final one. Selecting the best single model is one of the most important challenges. *MS* methods are based on the principle of parsimony. Therefore, the best model can be defined as the one that efficiently solves the tradeoff between complexity, i.e., number of features, and prediction ability.

In manufacturing, model interpretation is very important, since the learned characteristics of the system can be used to plan/design experiments to find optimal levels of process/product parameter. This situation highlights the importance of parsimonious modeling. While the *Akaike Information Criterion (AIC)* and *Bayesian Information Criterion (BIC)* from information and likelihood theory are well known *MS* criterion, they would not be generalized well to nonparametric (non-likelihood-based) models. To evaluate the discriminative capacity of a feature in a high conformance manufacturing environment, the *MPCD* was developed. However, since it does not penalize for extra parameters, adding a penalty term was the research challenge posed, and subsequently solved.

The *PMPCD* is presented as a *MS* criterion for binary classifiers in highly unbalanced data structures for virtually any classifier where is complexity is defined by the number of features in the model. This criterion solves the tradeoff between complexity and prediction ability. Since prediction is more important than parsimony, the criterion is mainly driven by prediction. Its score value indicates the ability to correctly classify the class with the smaller count. The criterion is broken down into two components: the first one rewards the prediction capacity based on *MPCD* and second one is a penalty function that induces parsimony by decreasing the *PMPCD* value based on the extra features.

The main objective of this research is to find the right amount of penalty and to keep prediction as the main driver while inducing parsimony. The challenge is to find a scaling factor for the penalty term. Experimental conditions to replicate manufacturing-derived data sets are defined and simulated; results exhibited the best scaling factor. *PMQ* proposes a new process learning paradigm based on *BM* that is aimed at improving process control and part quality; however, selecting the best single model with respect to parsimony and prediction is one of the main challenges. A new *MS* criterion was developed, *PMPCD*, for highly unbalanced data structures. In contrast with likelihood-based MS criterion, *PMPCD* is estimated based on the confusion matrix and the number of features; therefore, it can be widely applied as a *MS* criterion for virtually any classifier.

The main contribution of this work is the approach to find a penalty function of *MPCD* that efficiently solves the tradeoff between model complexity and prediction ability. The penalty function was tuned based on its ability to select the *perfect model,* i.e., that which separates the data. *PMPCD* showed high precision in selecting the *perfect model* in the simulated experimental conditions. According to simulation and experimental results, the model selection criterion induces parsimony by selecting the model with the minimum number of features needed for an effective/efficient defect detection. In this chapter, a new *AI*-based modeling paradigm is presented that uses *PMPCD* as a model selection criterion to induce parsimony.

Manufacturing Letters 15PA (2018) 55–58

Contents lists available at ScienceDirect

Manufacturing Letters

journal homepage: www.elsevier.com/locate/mfglet

Letters

Process-monitoring-for-quality — A model selection criterion

Carlos A. Escobar [a,b,*], Ruben Morales-Menendez [b]

[a] Global Research & Development, General Motors, Warren, MI, USA
[b] Tecnológico de Monterrey, Mexico

ARTICLE INFO

Article history:
Received 11 September 2017
Received in revised form 10 January 2018
Accepted 10 January 2018
Available online 11 January 2018

Keywords:
Model selection criterion
Process monitoring for quality
Binary classification
Highly unbalanced data structures
Big data driven manufacturing

ABSTRACT

The new big data driven manufacturing quality philosophy, *Process Monitoring for Quality (PMQ)*, proposes Big Data — Big Models, a new modeling paradigm that includes a big data-driven learning process that requires many models to be created to find the final one. Since many candidates are created, one of the main challenges is to select the model that efficiently solves the tradeoff between complexity and prediction. Most mature manufacturing organizations generate only a few *Defects Per Million of Opportunities (DPMO)*; therefore, manufacturing-derived data sets for classification of quality tend to be highly unbalanced. The *Penalized Maximum Probability of Correct Decision (PMPCD)* is developed to solve the posed tradeoff. According to simulation and experimental results, the model selection criterion induces parsimony by selecting the model with the minimum number of features needed for an effective/efficient defect detection.

1. Introduction

Manufacturing companies are intense users of *big data* bases, this industry generates/stores more data than any other [1]. *Machine Learning (ML)* techniques are applied to improve process monitoring, control and part quality [2–4]. Classification of quality is one of the most important applications [5,6], where relevant characteristics of the process/product are related to an ordinal or binary output aimed at detecting defects [7,8]. Most mature manufacturing organizations generate only a few *DPMO*; then, manufacturing-derived data sets for binary classification of quality are highly unbalanced.

The new big data driven manufacturing quality philosophy, *PMQ*, proposes Big Data – Big Models, a new modeling paradigm that includes a big data-driven learning process that often requires many models (classifiers) to be created to find the final one. Selecting the *best* single model is one of the most important challenges [7]. *Model Selection (MS)* methods are based on the principle of parsimony [9,10]. Modeling and *MS* are essentially concerned with the art of approximation. Therefore, the *best* model can be defined as the one that efficiently solves the tradeoff between complexity – number of features – and prediction ability. In contrast with other industries, where prediction is the main goal, model interpretation is very important, since the learned characteristics of the system can be used to plan/design experiments to find optimal levels of process/product parameters, this situation highlights the importance of parsimonious modeling.

Due to the lack of priori distinction between learning algorithms [11], different learning structures should be applied to develop the Big Models, followed by a *MS* criterion to select the final one. While the *Akaike Information Criterion (AIC)* and *Bayesian Information Criterion (BIC)* from information and likelihood theory are well known *MS* criterion [12,13], they would not be generalized well to nonparametric (non-likelihood-based) models.

To evaluate the discriminative capacity of a feature in a high conformance manufacturing environment, the *Maximum Probability of Correct Decision (MPCD)* was developed [7,14]. Since *MPCD* does not penalize for extra parameters, adding a penalty term was proposed as a future research [15,16].

The *PMPCD* is presented as a *MS* criterion for binary classifiers in highly unbalanced data structures (i.e., 3% of defects or less) for virtually any classifier. This criterion solves the tradeoff between complexity and prediction ability.

This letter is organized as follows. A brief theoretical background is in Section 2. Section 3 describes the *MS* criterion, results are in Section 4. A case study using a manufacturing-derived data set is presented in Section 5 and conclusions in Section 6.

2. Theoretical background

In predictive analytics, a confusion matrix [17] is a table with two rows and two columns that reports the number of *False*

* Corresponding author at: Global Research & Development, General Motors, Warren, MI, USA.
E-mail address: carlos.1.escobar@gm.com (C.A. Escobar).

https://doi.org/10.1016/j.mfglet.2018.01.001
2213-8463/

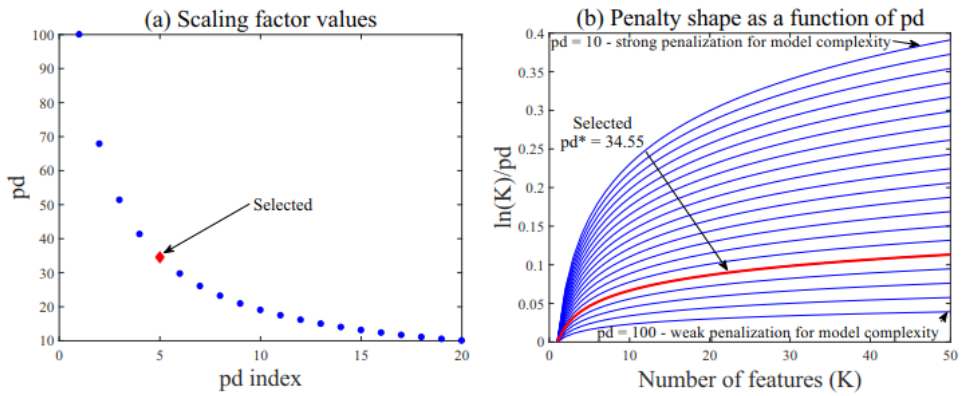


Fig. 1. Information of candidate scaling factors (pd).

Positives (FP), *False Negatives (FN)*, *True Positives (TP)*, and *True Negatives (TN)*. This allows more detailed analysis than just the proportion of correct guesses since it is sensitive to the recognition rate by class.

A type-I error (α) may be compared with a *FP* prediction; a type-II (β) error may be compared with a false *FN* [18]. They are defined as:

$$\alpha = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}, \quad \beta = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}}. \tag{1}$$

where α is the incorrect rejection of a true null hypothesis, and β is the incorrect retain of a false null hypothesis.

The *MPCD* is a probabilistic-based measure of classification performance [7,14]. In a very high conformance production rate, the *MPCD* is more sensitive to the recognition rate by class than just the proportion of correct guesses. Its estimated value is mainly driven by its capacity to correctly classify the class with the defective items. The α, and β errors are combined to estimate $MPCD \in [0, 1]$:

$$\mathrm{MPCD} = (1-\alpha)(1-\beta) \tag{2}$$

where higher score indicates better classification performance.

The *Support Vector Machine (SVM)* is one of the *ML* algorithms that can be applied to improve process control and part quality through binary classification. It is a discriminative classifier formally defined by a separating decision surface or hyperplane [19,20]. The algorithm outputs an optimal hyperplane founded on the large margin principle [21].

3. The model selection criterion

A *MS* criterion for binary classifiers trained using highly unbalanced data structures was developed: *PMPCD*. Since prediction is more important than parsimony [22], the criterion is mainly driven by prediction. Its score value indicates the ability to correctly classify the class with the smaller count. To induce parsimony, a penalty term based on the natural logarithm function – similar to *BIC* [12] – is added:

$$PMPCD = (1-\alpha)(1-\beta) - ln(K)/pd^{*} \tag{3}$$

where K is the number of features, and the model with the highest estimated value on the validation set [15,23,24] is the preferred one. Since training/testing an algorithm on the same data yields an overoptimistic result [25].

The term $(1-\alpha)(1-\beta)$ rewards the prediction capacity, while the penalty function $ln(K)/pd^{*}$ induces parsimony by decreasing the *PMPCD* value based on the extra features. Since the natural logarithm is a monotonically increasing function, the penalty values follow the same pattern. The main challenge is to find the right amount of penalty – pd^{*} scaling factor – to keep prediction as the main driver while inducing parsimony.

A random-based simulation was developed aimed at determining pd^{*} for modeling analysis with 50 features or less. The simulation randomly selects a number of *good* features $\in [2, 50]$ to create a *SVM* that perfectly separates the data. Then, n more candidate models – with no more than 50 features in total – are created with a fraction of the *good* features $\in [20\%, 100\%]$ plus a number of randomly selected *noisy* features. At the end of each run, there are models with more and lesser number of features than the *perfect* model with the full range of *MPCD* values.

The experimental space for *pd* was [10, 100], Fig. 1(a) shows the values of the 20 *pd* candidate values, and Fig. 1(b) shows their associated penalty shape in the range of interest. With $pd = 10$ the *MS* criterion will be dominated by the number of features, whereas with $pd = 100$ will be dominated by prediction.

To replicate the data structure of a typical manufacturing-derived data set, the simulations were run with a random number of observations $\in [2000, 20000]$, and a random number of bad units

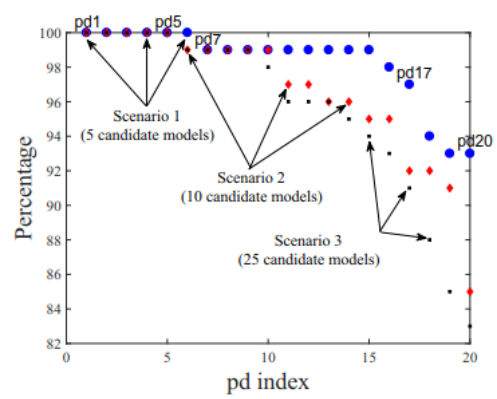


Fig. 2. Percentage of times the *perfect* model was selected with respect to *pd*.

$\in [0.1, 3]\%$. The simulation was replicated 100 times at each *pd* candidate value for consistency evaluation.

4. Results

To find $pd*$ three scenarios were analyzed, Fig. 2. According to results of scenario 1, the first 6 *pd* candidates, selected 100% of the times (replications) the *perfect* model. Whereas the ability to select the *perfect* model started to decline at *pd*-7. Therefore, $pd^* = 29.69$ (*pd*-6). In scenarios 2 and 3, the first 5 *pd* candidates, selected 100% of the times the *perfect* model, and the inflection point is at *pd*-6; therefore, the $pd^* = 34.55$ (*pd*-5). The ability of selecting the *perfect* model with respect to *pd* in the three scenarios is summarized in Fig. 2.

The results exhibited the best scaling factor to induce parsimony, while consistently maintaining prediction, as the main driver is *pd*-5. If $pd^* \in [pd\text{-}1, pd\text{-}4]$ the *PMPCD* will be mainly driven by prediction, whereas if $pd^* \in [pd\text{-}6, pd\text{-}20]$, it will be driven by complexity and its ability to select the *perfect* model will be hampered.

$$PMPCD = (1-\alpha)(1-\beta) - ln(K)/34.55 \quad (4)$$

The *PMPCD* score can take negative values, e.g., if the estimated $MPCD = 0$ and the model contains more than 1 feature. $PMPCD = 0$ indicates a single-feature model with an estimated $MPCD = 0$. In general, a model with an estimated $MPCD = 0$ indicates a failure to recognize all the examples of the class with the smaller count. The maximum value is 1, which indicates a single-feature model that perfectly separates the data.

5. Case study

The l_1-regularized logistic regression [26] is used as the learning algorithm for the classification task (e.g, defect detection). A set of 99 candidate models are created by varying the regularization value (λ). As the value of λ increases, the number of features included in the model decreases. In the context of Big Models, *PMPCD* scores are used to select the *best* single model, the complete learning process herein presented can be found in [16].

The data used for this analysis is derived from the ultrasonic metal welding [27] of battery tabs for the Chevrolet Volt [7] – same project in which the *MPCD* was developed. It is a solid state bonding process that uses high frequency ultrasonic vibration energy to generate oscillating shears between metal sheets clamped under pressure. It is an ideal process for bonding conductive materials such as copper, aluminum, brass, gold, and silver, and for joining dissimilar materials. Recently, it has been adopted for battery tab joining in the manufacturing of vehicle battery packs. A very stable process, that only generates a few defective welds per million of opportunities.

The collected data set contains a binary outcome (*good*/*bad*) with 54 features derived from signals (e.g., acoustics, power, and linear variable differential transformers) following typical feature construction techniques [28]. The data set is highly unbalanced, since it contains only 36 bad batteries out of 40,231 examples (0.09%). The data set is partitioned following the hold-out validation [24] scheme: training set (18,495 – including 20 *bads*), validation set (12,236 – 9 *bads*), and testing set (9500 – 7 *bads*).

Fig. 3(a),(b) shows the associated values of λ and number of features of each candidate model, respectively. Fig. 3(c) shows the estimated *MPCD*, while Fig. 3(d) shows the *PMPCD*. According to the criterion, model 88 should be selected, this model contains 4 features with an estimated $MPCD = 0.8805$. Although models 70 (14 features) and 73 (11 features) have an estimated $MPCD = 0.8818$, their complexity is significantly higher; therefore they were not selected by the criterion. Selected model correctly (effectively) detected the seven bads in the testing set ($MPCD = 0.9979, \beta = 0, \alpha = 0.0021$). Based on the testing performance, it is clear that only four features are needed for an efficient – low α – defect detection.

6. Conclusions

PMQ proposes a new process learning paradigm based on Big Models that is aimed at improving process control and part

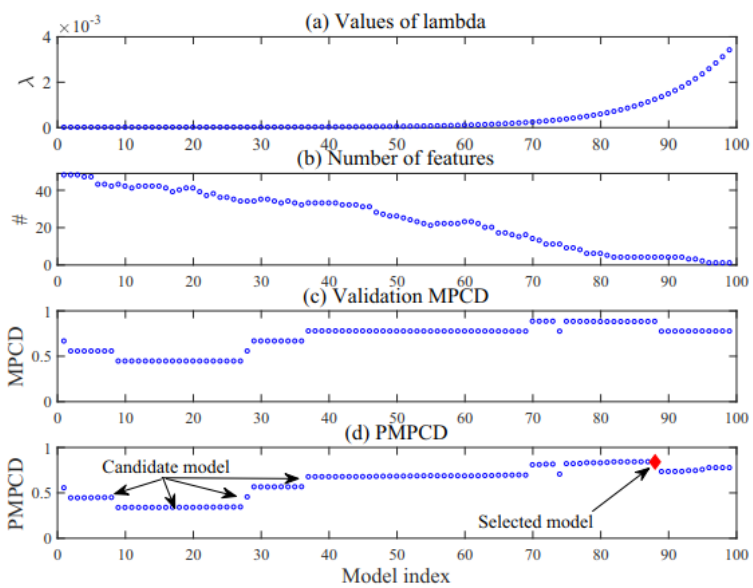


Fig. 3. Candidate model information.

quality; however, selecting the *best* single model with respect to parsimony and prediction is one of the main challenges. A new *MS* criterion was developed, *PMPCD*, for highly unbalanced data structures. In contrast with likelihood-based *MS* criterion *PMPCD* is estimated based on the confusion matrix and the number of features; therefore it can be widely applied as a *MS* criterion for virtually any classifier.

The main contribution was the approach to find a penalty function of *MPCD* that efficiently solves the tradeoff between model complexity and prediction ability. The penalty function was tuned based on its ability in selecting the *perfect* model. *PMPCD* showed high precision in selecting the *perfect* model in the simulated experimental conditions. Empirical results exhibited the criterion's ability to select a parsimonious model with high prediction ability.

Future research could focus on replacing *MPCD* with other confusion matrix-based measures of classification performance to develop other *MS* criterion for different data structures.

References

[1] Baily M, Manyka J. Is Manufacturing'Cool'Again. McKinsey Global Institute; 2013.
[2] Ghorai S, Mukherjee A, Gangadaran M, Dutta PK. Automatic defect detection on hot-rolled flat steel products. IEEE Trans Instrum Measure 2013;62(3):612–21.
[3] Yin S, Li X, Gao H, Kaynak O. Data-based techniques focused on modern industry: an overview. IEEE Trans Ind Electron 2015;62(1):657–67.
[4] Yin S, Kaynak O. Big data for modern industry: challenges and trends [Point of View]. Proc IEEE 2015;103(2):143–6.
[5] Malhotra R. A systematic review of machine learning techniques for software fault prediction. Appl Soft Comput 2015;27:504–18.
[6] Köksal G, Batmaz I, Testik MC. A review of data mining applications for quality improvement in manufacturing industry. Expert Syst Appl 2011;38(10):13448–67.
[7] Abell J, Chakraborty D, Escobar C, Kee H, Wegner D, Wincek M. Big data driven manufacturing — Process-monitoring-for-quality philosophy. ASME J Manuf Sci Eng Data Sci-Enhanced Manuf 2017;139(10).
[8] Uraikul V, Chan W, Tontiwachwuthikul P. Artificial Intelligence for Monitoring and Supervisory Control of Process Systems. Eng Appl Artif Intell 2007;20(2):115–31.
[9] Breiman L. The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. J Am Stat Assoc 1992;87(419):738–54.
[10] Zhang P. On the choice of penalty term in generalized FPE criterion. In: Selecting Models from Data. Springer; 1994. p. 41–9.
[11] Wolpert DH. The lack of a priori distinctions between learning algorithms. Neural Comput 1996;8(7):1341–90.
[12] Burnham KP, Anderson DR. Model selection and multimodel inference: a practical information-theoretic approach. Springer Science & Business Media; 2003.
[13] De Brabanter J, Pelckmans K, Suykens JA, De Moor B, Vandewalle J. Robust complexity criteria for nonlinear regression in NARX models. In: Proc of the 13th System Identification Symp. p. 79–84.
[14] Abell JA, Spicer JP, Wincek MA, Wang H, Chakraborty D, Binary Classification of Items of Interest in a Repeatable Process, US Patent, no. US8757469B2, June 2014. [Online]. Available:www.google.com/patents/US20130105556.
[15] Escobar CA, Morales-Menendez R. Machine learning and pattern recognition techniques for information extraction to improve production control and design decisions. In: Perner Advances in Data Mining, ICDM. Springer Verlag; 2017. p. 285–95.
[16] Escobar CA, Morales-Menendez R. Machine learning techniques for quality control in high conformance manufacturing environment. Adv. Mech. Eng. 2018;10(2):1–16.
[17] Fawcett T. An Introduction to ROC Analysis. Pattern Recogn Lett 2006;27(8):861–74.
[18] Devore J. Probability and statistics for engineering and the sciences. Cengage Learning; 2015.
[19] Burges CJ. A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 1998;2(2):121–67.
[20] Ng A, Support Vector Machines, Machine Learning, 2008, lecture note. [Online]. Available:http://cs229.stanford.edu/notes/cs229-Notes3.pdf.
[21] Murphy K. Machine learning: a probabilistic perspective. MIT press; 2012.
[22] Shibata R, Statistical Aspects of Model Selection, From Data to Model, 1989, 215–40.
[23] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Stat Surveys 2010;4:40–79.
[24] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning, vol. 1. Berlin: Statistics Springer; 2001.
[25] Larson S. The shrinkage of the coefficient of multiple correlation. J Educ Psychol 1931;22(1):45.
[26] Tibshirani R. Regression shrinkage and selection via the LASSO. J R Stat Soc Ser B (Methodological) 1996:267–88.
[27] Shao C, Paynabar K, Kim T, Jin J, Hu S, Spicer J, et al. Feature selection for manufacturing process monitoring using cross-validation. J Manuf Syst 2013;10.
[28] Liu H, Motoda H. Feature extraction, construction and selection: a data mining perspective. Springer Science & Business Media, vol. 453, 1998.

Chapter 4

Process-Monitoring-for-Quality – Big Models

PMQ is a *IBD*-driven quality philosophy aimed at defect detection (through binary classification). It is founded on *BM*, a predictive modeling paradigm based on machine learning, statistics and optimization, that includes a learning aspect that requires many models to be developed to find the *final model*. The term *BM* was coined to describe a learning paradigm that requires many models to be developed to find the *final model*, or a model combination scheme aimed at optimizing the classification performance of a single classifier through a decision combination rule. *PMQ* proposed a four-step problem solving strategy (acsensorize, discover, learn, and predict) to solve the pattern recognition problem, where the learn and predict steps are addressed by *BM.* In big data-based analyses, the data structure is not known in advance; therefore, there is no a priori distinction between *MLA*. Unless a single-feature model perfectly separates the two classes, various algorithms should be included in the analysis as an effort to improve the prediction ability on unseen data generalization of a single classifier. Pattern classification is a broad topic with no universal best approach, finding a superior quality solution requires machine learning and domain knowledge.

A learning approach is proposed based on seven *MLA* and ad hoc model selection/validation tools with the capacity to solve a wide spectrum of binary classification problems. The learning process is developed considering the main challenges posed by manufacturing-derived data sets: (1) many features, including relevant, irrelevant, and redundant; and (2) highly unbalanced classes. It is broken down into five steps supported by theory and empirical evidence, in which the outcome is a parsimonious predictive system. The proposed learning scheme is a good starting point supported by theory and empirical evidence. *BM* use features defined following typical feature construction techniques or from process physical knowledge. They are derived from sensor signals obtained from a well acsensorized process.

Since manufacturing systems tend to be time-dependent, model development is performed following a time-ordered hold-out data partition strategy. An approach in which a data set is divided into three subsets: training, validation, and testing.

Since *BM* are founded on the principle of parsimony, feature selection is highly recommended. Depending on the data structure of the problem, BM can be developed using either one, a few, or all the reviewed *MLA*. From generalization perspective, the main objective of creating many models is to prevent overfitting or underfitting the data set by hyperparameter tuning. Developing Big Models is a multi-step approach: (1) data preprocessing, (2) model creation, (3) model validation/selection, (4) prediction optimization, and (5) model testing. High level knowledge can be extracted through *BDBM* which can be used for redesigning manufacturing processes.

Two cases studies were performed on real data to validate the proposed learning framework. With no prior knowledge about the underlying system that generated the data sets, the pattern recognition problem was effectively solved by the *BM* learning paradigm. In the first case study, the pattern was better approximated by linear classifiers and the prediction was optimized through a non-ensemble combination. In the second case study, the pattern was better recognized by nonlinear algorithms and no decision combination outperformed a single model.

The case studies exhibited the capacity of *BM* to solve a wide spectrum of binary classification problems. Parsimonious modeling helped to identify the most relevant features.

Binary classification is a very broad topic, with a plethora of options. Poor domain knowledge can lead to a low-quality solution. A good starting point based on theory and empirical evidence was presented here. This paper addressed the learning problem of *PMQ*, which is solved by *BM*. A learning paradigm that requires many models to be developed to find the *final model*. Manufacturing pattern recognition problems pose several challenges. The theoretical aspect of modeling these data sets was reviewed. A learning scheme was developed based on seven *MLA*, which have the capacity to solve a wide spectrum of binary classification problems. *BM* can mine massive amounts of data to learn manufacturing systems. Predictive model and derived knowledge, have the capacity to solve complex manufacturing problems that cannot be efficiently solved by traditional quality methods. And therefore, potentially move the process learning curve off the plateau of no progress. Once the ability of *BM* to detect rare quality events was demonstrated, the next research challenge was to determine how this *QC* paradigm can be applied to solve *SPC*-intractable problems, results of the study are presented in this chapter.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 26 (2018) 1167–1179

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

North American Manufacturing Research Conference

Process-Monitoring-for-Quality — Big Models

Carlos A. Escobar[a,b,*], Jeffrey A. Abell[b], Marcela Hernández-de-Menéndez[a], Ruben Morales-Menendez[a]

[a]*Tecnológico de Monterrey, Av E Garza Sada 2501, Monterrey NL, 64849 México*
[b]*Global Research & Development, General Motors, Warren, MI, USA*

Abstract

Process Monitoring for Quality (PMQ) is a big data-driven quality philosophy aimed at defect detection (through binary classification) and empirical knowledge discovery. It was originally developed to solve a complex manufacturing quality problem. It is founded on *Big Models*, a predictive modeling paradigm based on machine learning, statistics and optimization, that includes a learning aspect that requires many models to be developed to find the *final model*. When dealing with big data, the data structure is not known in advance; therefore, there is no a priori distinction between learning algorithms, and a plethora of options to choose from. The learning scheme of *Big Models* is described, which is based on several well known learning algorithms with the capacity to effectively solve a wide spectrum of binary classification problems. The main challenges of manufacturing pattern recognition problems are discussed and addressed to provide a strong foundation to the *Big Models* learning paradigm. Finally, two defect detection case studies are presented with highly unbalanced data derived from real manufacturing systems to validate the proposal.

Peer-review under responsibility of the scientific committee of the 4th International Conference on System-Integrated Intelligence.

Keywords: Manufacturing; Quality; Big data; Big Models; Process learning; Binary classification; Predictive modeling.

1. Introduction

Process Monitoring for Quality (PMQ) is a big data-driven quality philosophy aimed at defect detection and empirical knowledge discovery [1]. It is a blend of process monitoring and quality control founded on *Big Data-Big Models*, Fig. 1, catalysts in shaping the new era of manufacturing. Its first application led to the development of an initial quality monitoring process for the ultrasonic welding of battery tabs in the Chevrolet Volt. The defect detection was formulated as a binary classification (i.e., good, suspect). Classification of quality is one of the most important applications [2,3], where relevant characteristics of the process or product are observed and related to an ordinal or binary output aimed at detecting defects [4]. *PMQ* has the potential to solve a whole range of hitherto intractable manufacturing problems [5].

PMQ adapted the traditional manufacturing quality problem solving strategies [1,6], and proposed a four-step approach – acsensorize, discover, learn, predict – to solve the pattern recognition (classification) problem, Fig. 2, in which the learning step is solved by *Big Models*. A predictive modeling paradigm based on machine learning, statistics and optimization that includes a learning aspect that requires many models to be developed to find the *final model* (classifier) [7], a group of homogeneous models to create an ensemble [8] or a group of heterogeneous models to create a *Multiple Classifier System (MCS)* (non-ensemble) [9,10], the last two aimed at optimizing the classification performance of a single classifier – through a decision combination rule. *Big Models* are mainly used for prediction, but they can be used for "relational" or "explanatory" purposes too.

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.

E-mail address: carlos.1.escobar@gm.com

2351-9789 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 4th International Conference on System-Integrated Intelligence.
10.1016/j.promfg.2018.07.153

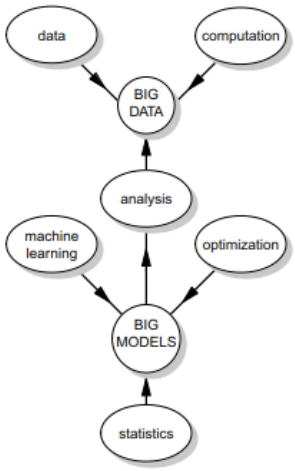


Fig. 1. Big Data-Big Models [1].

In big data-based analyses, the data structure is not known in advance; therefore, there is no a priori distinction between *Machine Learning Algorithms (MLA)* [11]. For example, a single-neuron perceptron can solve linearly separable patterns (*AND*, *OR* gates), while it cannot solve linearly inseparable patterns (*XOR* gate) [12]. Unless a single-feature model perfectly separates the two classes, various algorithms should be included in the analysis as an effort to improve the prediction ability on unseen data (generalization) of a single classifier.

Pattern classification is a broad topic with no universal *best* approach, finding a good quality solution requires machine learning and domain knowledge. A learning approach is proposed based on seven *MLA* and *ad hoc* model selection/validation tools with the capacity to solve a wide spectrum of binary classification problems. The learning process is developed considering the main challenges posed by manufacturing-derived data sets: (1) many features, including relevant, irrelevant, and redundant; and (2) highly unbalanced classes [13]. And it is broken down into five steps supported by theory and empirical evidence, in which the outcome is a parsimonious predictive system.

The rest of this paper is organized as follows. It starts with a brief review of the most common classification scenarios and relevant *MLA* in section 2. Section 3 addresses the learning step (predictive modeling), followed by two case studies in section 4. Finally, section 5 concludes the paper.

Table 1. Acronyms Table

Acronym	Definition
ANN	Artificial Neural Network
CT	Classification Tree
FN	False Negatives
FP	False Positives
HCR	Hybrid Correlation- and Ranking-based
KNN	K Nearest Neighbors
LASSO	Least Abs. Shrinkage and Sel. Op.
LR	Logistic Regression
MCS	Multiple Classifier System
MLA	Machine Learning Algorithm(s)
MLE	Maximum Likelihood Estimate
MPCD	Maximum Prob. of Correct Decision
NB	Naive Bayes
NLL	Negative Log-Likelihood
OCTM	Optimal Classification Threshold
PMPCD	Penalized Max. Prob. of Correct Dec.
PMQ	Process Monitoring for Quality
SVM	Support Vector Machine
TN	True Negatives
TP	True Positives

2. Background

The objective of this review is to create a list – while not exhaustive – of diverse supervised *MLA*, Table 2. Proposed list was developed to include linear and non-linear [12], parametric and non-parametric [14], stable and unstable [15], generative and discriminative [16] algorithms, so that they together can solve a wide spectrum of binary classification problems[1].

For graphical purposes, the last column of Table 2 describes some common binary classification scenarios that can be effectively solved by each *MLA*. For graphical feasibility, scenarios are presented in two dimensional plots, Fig. 3. Described match should not be viewed as definitive and unique.

There are dozens of more algorithms that could be listed, some of them can be quite effective in some specific scenarios. However, most of them are some adaptation of the algorithms in this list, which provides a strong foundation for *Big Models*. A brief review of the recommended *MLA* is included.

[1] Authors acknowledge that some algorithms can change their taxonomy (e.g., from parametric to non-parametric) depending upon their definition.

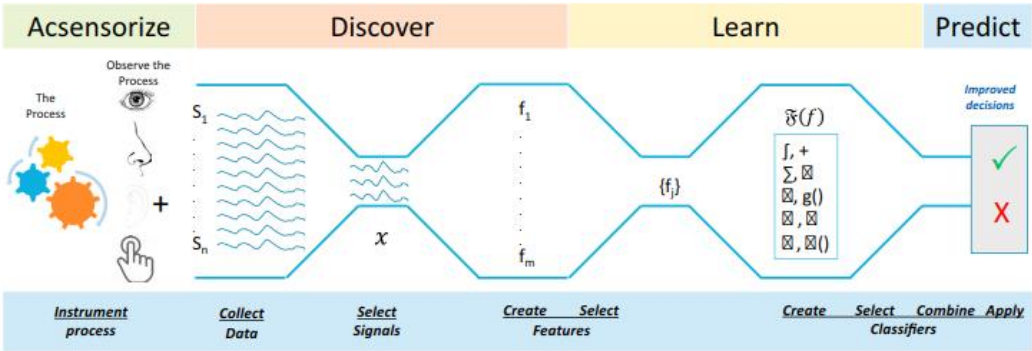


Fig. 2. *PMQ* problem solving strategy [1].

Table 2. Characteristics of the *MLA*

MLA	Linear	Non-linear	Parametric	Non-parametric	Stable	Un-stable	Gene-rative	Discri-minative	Scenario
SVM	✓		✓		✓			✓	1
LR	✓		✓		✓			✓	1
NB		✓*	✓		✓		✓		5
KNN		✓		✓	✓			✓	3
ANN		✓	✓**			✓		✓	1,2,3,4,5,6
SVM (RBF)		✓	✓	✓	✓			✓	2
CT(ensemble)		✓		✓	✓			✓	4,6

*with numeric features. **with a set of parameters of fixed size.

2.1. Support Vector Machine

Support Vector Machine (SVM) is used in both linear and non-linear classification. In a binary classification problem, the data are mapped and a hyperplane that maximizes the distance between selected data points is developed to serve as a boundary that separates the classes [17]. The support vectors are the closest points to the boundary line; these are the only data taken in consideration for determining the best position of the hyperplane [18]. When a new instance enters into the system, this is assigned and mapped in the corresponding side of the hyperplane. When the linear condition is not met, a kernel function such as polynomial or (*Gaussian*) *Radial Basis Function (RBF)* kernels [19] is used to transform the data and convert it into a linearly separable structure[20]. *SVM* has a low training speed [21] and does not work well when classes share space [17]. For the mathematical formulation and penalized (regularized) options refer to [22–25]. To create a *SVM*-based ensemble refer to [26].

2.2. Logistic Regression

The *Logistic Regression (LR)* is a parametric algorithm that uses a transformation of the values of a linear function. It models the probability distribution of the class label y, given a feature vector x [27]:

$$P(y = 1|x;\theta) = \sigma(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}. \quad (1)$$

where $\theta \in \mathbb{R}^N$ are the parameters of the *LR* model and $\sigma(\cdot)$ is the sigmoid function (logistic function) that maps values from $(-\infty, \infty)$ to $[0, 1]$. The discrimination function itself is not linear, but the decision boundary is still linear.

The most common approach to estimate the parameters of a statistical model is to compute the *Maximum Likelihood Estimate (MLE)*. The problem of finding the *MLE* of the parameters θ for the unregularized *LR* can be defined by in terms of the *Negative Log-Likelihood (NLL)*:

$$\min_{\theta} \sum_{i=1}^{M} -\log p(y^{(i)}|\mathbf{x}^{(i)};\theta). \quad (2)$$

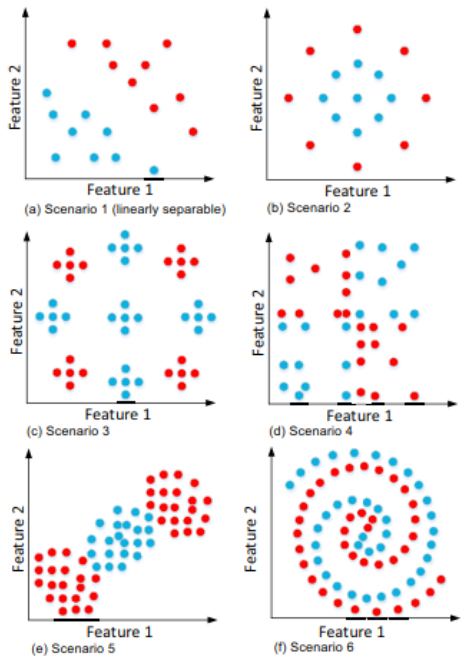


Fig. 3. Different classification scenarios (classes differentiated by color).

The *NLL* for *LR* is:

$$NLL = -\sum_{i=1}^{M}[y^{(i)}log\mu^{(i)} + (1 - y^{(i)})log(1 - \mu^{(i)})].(3)$$

where $\mu^{(i)} = sigm(\theta^T x^{(i)})$. It is also called the *Cross-Entropy Error* function [14]. Under the Laplacian prior $p(\theta) = (\lambda/2)^N \exp(-\lambda\|\theta\|_1)(\lambda > 0)$, the *Maximum A Posteriori* estimate of the parameters θ is:

$$\min_{\theta} \sum_{i=1}^{M} -\log p(y^{(i)}|\mathbf{x}^{(i)};\theta) + \lambda\|\theta\|_1. \quad (4)$$

This optimization problem is referred to as l_1-regularized *LR*. Adding the l_1-regularization makes the optimization problem computationally more expensive. For solving the l_1-regularized *LR* [28], the *Least Absolute Shrinkage and Selection Operator (LASSO)* is an efficient method.

As the value of λ increases, the number of features included in the model decreases. The higher the value of λ, the lower the chance of overfitting with too many redundant or irrelevant variables, but too high value of λ would underfit the data. To solve the bias/variance tradeoff, λ should be tuned. To create a *LR*-based ensemble refer to [29].

2.3. K Nearest Neighbors

The *K Nearest Neighbors (KNN)* is a non-parametric algorithm based on the closest training examples in the feature space. It is one of the most simplest techniques that requires little or no knowledge about the distribution of the data. This algorithm classify new examples considering K surrounding neighbors, proximity is defined by a distance metric, [30]. If a pattern exists, the performance of *KNN* is primarily determined by the choice of K and the type of distance metric used. Intuitively, *KNN* tries to approximate a locally smooth function; a small value of K will lead to a large variance, whereas large values may lead to a large bias. The main challenge is to find an optimal value for K that achieves the right trade off between the bias and the variance of the model. Because manufacturing derived data sets tend to be highly unbalanced, exhaustive search of feasible odds numbers can be exploited, $\{K|3, ..., b$, and $2 \nmid K\}$, where b is the number of defects in the training set. For alternative methods refer to [31,32]. To create a *KNN*-based ensemble refer to [33].

2.4. Classification Tree

Classification Tree (*CT*) is a non-parametric algorithm [19] in which consecutive related questions are asked to predict the class of a given instance. *CT* is composed of a root node, branches, and terminal or non-terminal nodes. A node represents a variable; the root node is the starting point, where all the learning data will began to be divided in subsets. The branches are the possible solutions of the different variables. In the nonterminal nodes a comparison of a numerical or nominal attribute value with a constant is performed. A terminal or leaf node is not divided anymore and here is where a class is defined [34] [35]. When a new instance enters to the tree, it starts in the root node. At this and each subsequent levels a specific test is developed and, depending on the result, the instance goes to the corresponding branch. This process is repeated until the data reach the terminal nodes. There are measures for determining which variable best divide the input data including information gain, gain ratio and *Gini* index [36]. One important step in developing a *CT* is pruning it to reduce its size and avoid overfitting, [37] [38]. Some popular *CT*s include *ID3*, *CART*, *J48*, *ADTree* [39] and C4.5 [38]. For feature selection via regularized trees refer to [40].

The main advantage of this method is that it is easy to visualize and understand [41]. Also, it is not greatly affected by outliers or missing values and can be used to explore a problem in order to define significant variables. The main disadvantage of *CT*s is the tendency to overfitting [42]. This issue can be approached by constructing ensembles of *CT*s; bagging, boosting, and randomization [8,43,44]. For an ensemble aggregation method in highly unbalanced data refer to [13].

2.5. Naive Bayes

Naive Bayes (*NB*) is a parametric algorithm based on *Bayes* theorem of conditional probabilities. The basic classification process consists on determining a score based on the training data values. In a simple binary classification problem, a high score is associated with one class and a small score is related to the other class. The result is compared with a threshold to determine the final class [37]. *NB* is fast calculating the needed probabilities as it only performs one scan to the data [45]. *NB* has a strong independence of variables assumption [19]. Another assumption of *NB* is that numerical values always have a normal distribution. *NB* is easy to develop [37] and its classification process is easy to understand as well. It is not affected by missing values as it omits them. In this sense, *NB* is suitable for working with high amount of data [37]. *NB* cannot remove irrelevant attributes and its performance is highly dependent on the attribute selection procedure used. Finally, this algorithm is very affected by irrelevant features [45].

2.6. Artificial Neural Network

Artificial Neural Network (*ANN*) non-linear algorithm design is a replica of the human neural structure. It is arranged in a determined number of nodes ordered in layers. The most popular *ANN* method is the multilayer perceptron which has been successfully used in a wide range of applications [46]. For developing the learning process in a supervised learning approach, the system is trained with examples of inputs and outputs so a model is developed to predict new instances [47].

It has a good performance in presence of multicollinearity [38] and works well with great amount of features and noisy data. However, it requires lots of computational resources (e.g. memory usage), is affected by missing data [38]. To control the effective complexity of the network and therefore prevent overfitting a Bayesian regularization method can be used to train the network [12,48].

3. Big Models

The term *Big Models* was coined to describe a learning paradigm that requires many models to be developed to find the *final model*, or a model combination scheme aimed at optimizing prediction. *PMQ* proposed a four-step problem solving strategy – acsensorize, discover, learn, predict –, Fig. 2, to solve the pattern recognition problem. A brief overview of the learning paradigm is presented.

Depending on the data structure of the problem, *Big Models* can be developed using either one, a few, or all the reviewed *MLA*. From generalization perspective, the main objective of creating many models is to prevent overfitting or underfitting the data set by hyperparameter [49] tuning. Overfitting is often the result of excessively complicated models, they show low bias, but high variance [50,51]. Whereas underfitting is often a result of an excessively simple model [14]. Both overfitting and underfitting lead to poor generalization. However, modest overfitting is less damaging than underfitting [52].

Many models have important parameters which cannot be directly estimated from the data, for example, in the *KNN*, there is no analytical formula available to calculate an appropriate value [49].

Big Models use features defined following typical feature construction techniques [53] or from process physical knowledge. They are derived from sensor signals obtained from a well acsensorized process [1]. Because manufacturing systems tend to be time-dependent, model development is performed following a time-ordered hold-out data partition strategy. An approach in which a data set is divided into three subsets: training, validation, and testing. Since *Big Models* are founded on the principle of parsimony [54], feature selection is highly recommended.

Although each learning algorithm may require a different set of assumptions about the data to be able to classify, the two basic assumptions underlying most of *MLA* are that the training examples are independent and identically distributed (iid) according to an unknown probability distribution. These two assumptions can be verified if the selected model generalize well to the testing set. *Big Models* require a feature space with a pattern and a good ratio between number of features and number of examples [55,56], to avoid the curse of dimensionality [57], $ft > n$, where ft (features) and n (examples).

Developing *Big Models* is a multi-step approach aimed at solving the bias and variance tradeoff, and at

obtaining an unbiased estimation of generalization performance of the predictive system; information used to determine – under uncertainty – if *Big Models* have the capacity to solve the problem at hand. The proposed steps: (1) data preprocessing, (2) model creation, (3) model validation/selection, (4) prediction optimization, and (5) model testing. Figure 4 shows the *Big Models* learning scheme.

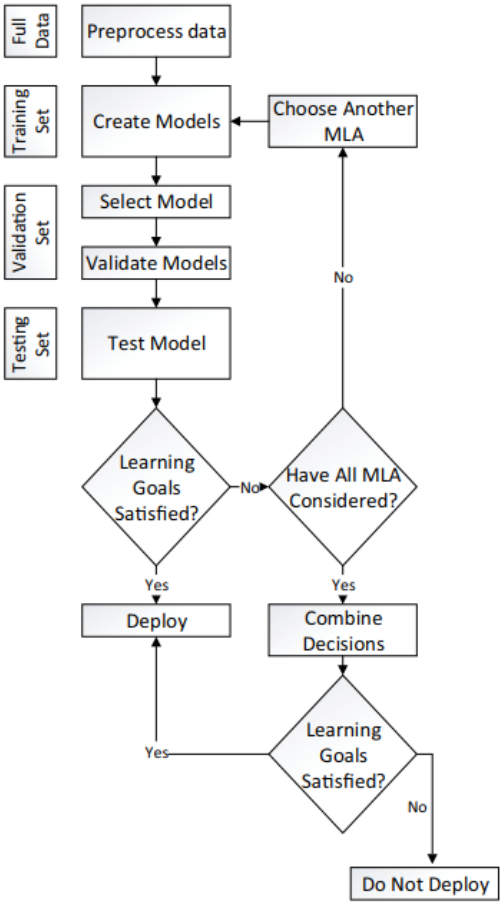


Fig. 4. *Big Models* learning scheme.

3.1. Data preprocessing

Cleaning and preparing data is a critical first step in *Big Models*. The basic idea is to extract the most information available in the data set. In manufacturing, due to sensors can fail at any time, data sets often have missing values. Handling missing data is important as many *MLA* do not support data with missing values. If the data set has a small percentage of missing values, imputation methods [58] have been shown to work well. Otherwise row, or column elimination may be a better option.

Due to acsensorization – adding a multitude of dissimilar sensors –, the range of values of raw data varies widely, in some *MLA*, objective functions will not work properly without standardization/scaling. Feature scaling improves the convergence of steepest descent/ascent-based optimization algorithms [59], which do not possess the property of scale invariance (e.g., *SVM*, *LR*, *ANN*). Moreover, for the *KNN*, if one of the features has a broad range of values, the distance will be governed by this particular feature. Standardization methods can be found in [60].

When a new manufacturing process is initially deployed, it often occurs that engineers do not fully understand the physics of the process and the huge amount of information is used to create tens, hundreds or even thousands of features, which frequently include relevant, irrelevant and redundant ones. Theoretical analysis and empirical evidence show that the last two are not helpful in solving pattern recognition problems: (1) they may have negative effect on the classification performance because of the mutual effect between the features; (2) they may significantly increase computing time; and (3) it is more difficult to extract high-level knowledge from the analysis [57,61–64]. Moreover, analyses with many irrelevant features and small sample size, can potentially lead to the development of spurious models [65]. Irrelevant and redundant features can be eliminated by filtering algorithms [66].

Filtering algorithms select variables independently of the classification algorithm or its error criteria, they assign weights to features individually and rank them based on their relevance to the class labels. A feature is considered good if its associated weight is greater than the user-specified threshold [57]. The advantages of feature ranking algorithms are that they do not over-fit the data and are computationally faster than wrappers, and hence they can be efficiently applied to big data sets containing many features [61].

3.2. Model creation

The training set is used to create many candidate models by varying the hyperparameters of the selected *MLA*. Regularization and/or wrapper feature selection algorithms [66] may boost parsimonious modeling. It is a good idea to create a scatter plot to visualize the pattern before a *MLA* is chosen. Most of the times graphical analysis are unfeasible, and because the data structure is not known in advance, there is no a priori dis-

tinction between *MLA*. Unless a single-feature model – well understood from physics perspective – perfectly separates the two classes, all the reviewed *MLA* listed in section 2 should be included in the analysis.

Due to the iterative and often ad hoc process of searching the data for patterns [1,54], spurious models can be developed. Performing model selection, and prediction evaluation using the training set, carries two serious problems; model bias and model uncertainty [54].

3.3. Model selection/validation

Model selection/validation approaches are generally based on the confusion matrix [67]. Model selection induces parsimony, and essentially leads to the model(s) with the smallest number of parameters for adequate representation of the data. It is a solution for the tradeoff between model complexity and prediction ability. The *Penalized Maximum Probability of Correct Decision (PMPCD)* can be used to solve this tradeoff [68]. The model selection scores are estimated on the validation set.

Big Models paradigm does not encourage tests of null hypotheses leading to the arbitrary classification (dichotomies) of *significant* versus *not significant* for model validation. Since there is no statistical theory that supports the notion that hypothesis testing with a fixed $alpha(\alpha)$ level is the basis. Hypothesis testing may be uninformative with respect to evaluating the classification ability of a model; models are validated based on their ability to meet the learning targets for the project.

If the classes are highly unbalanced, the *Maximum Probability of Correct Decision (MPCD)* should be used to evaluate the models' ability to detect those rare events [1,7,69], since it tends to be very sensitive to β error in these data structures, otherwise other methods of classification performance can also be used [70]. If the competing models have the same number of features, *MPCD* or any other relevant measures of classification performance can be used as a means for model selection. Relevance is determined based on the data structure, or based on the importance of committing α and β errors. The *final model* of each *MLA* is called classifier.

3.4. Prediction optimization

The classification performance of a single classifier, can be optimized by combining the decision of two or more classifiers. Ensemble methods [8,10] and non-ensemble methods [71,72] are powerful solutions to complex classification problems. Classification optimization refers to the prediction step of Fig. 2.

The decision combination rule is developed in this stage. Although there is no general superiority of a particular combination scheme, for binary classifiers a simple majority vote rule can be implemented. This simplification allows easy integration of all different kinds of classifiers architectures [10]. For decision combination rules see [72–74].

3.5. Model testing

An unbiased generalization performance of the predictive system can be estimated on the testing set. Only the *final model* or decision combination rule (ensemble or non-ensemble) is assessed. This step eliminates (or lessen) the bias induced by data snooping in the training and validation stages. Generalization results are compared with the learning targets to determine if the predictive system can be deployed. If the prediction performance needs to be improved, machine learning diagnosis [75] may help to troubleshoot the pattern recognition problem, and therefore improve it. This diagnosis should be performed before requesting more data or concluding that the learning goals cannot be satisfied.

3.6. Discussion

The underlying theory of *Big Models* was briefly introduced in this section. The proposed learning scheme is a good starting point supported by theory and empirical evidence that addresses the main challenges of manufacturing-derived data sets. The importance of each of the steps as well as the choice of tools/methods is illustrated in the following section.

4. Case Studies

Based on two real case studies the proposal will be exemplified and discussed.

4.1. Ultrasonically Metal Welding Case

The used data for this analysis is derived from ultrasonically metal welding [64] of battery tabs for the Chevrolet Volt [1], an extended range electric vehicle. A very stable process, that only generates a few defective welds per million of opportunities. However, all the welds in the battery must be good for the electric motor to function. This problem representation not only highlights the engineering intellectual challenge, but also the importance of a zero-defects policy.

The collected data set contains a binary outcome (*good*/*bad*) with 54 features derived from signals (e.g.,

acoustics, power, and linear variable differential transformers) following typical feature construction techniques [53]. The data set is highly unbalanced, since it contains only 36 bad batteries out of 40,231 examples (0.09%). The data set is partitioned following the time-ordered hold-out validation scheme: training set (18,495 - including 20 *bads*), validation set (12,236 - 9 *bads*), and testing set (9,500 - 7 *bads*). An extended learning process herein presented, can be found in [76].

Because a single classifier cannot perfectly separate the classes, a non-ensemble prediction approach is developed. Due to space limitations, only a detailed description of the learning process of the *LR* is reported, which shows the relevance of each step.

The solution process should be viewed as an illustrative example, more than a definitive best way to solve the pattern recognition problem. Following the five steps of the proposal:

4.1.1. Step 1: Data preprocessing

To eliminate irrelevant features, the data set is initially preprocessed using the *ReliefF* algorithm [77]. *ReliefF* is run with $k = 10$ nearest neighbors and a feature significance threshold of $\tau = 0.0223$, (calculated based on $1/\sqrt{\alpha m}$, $\alpha = 0.05$, and $m = 40,231$), according to the algorithm, feature 26 is the most important feature, while feature 14 is the lowest quality feature. Figure 5 summarizes the feature ranking and which features are selected based τ. According to *ReliefF*, 43 features –out of 54– should be selected.

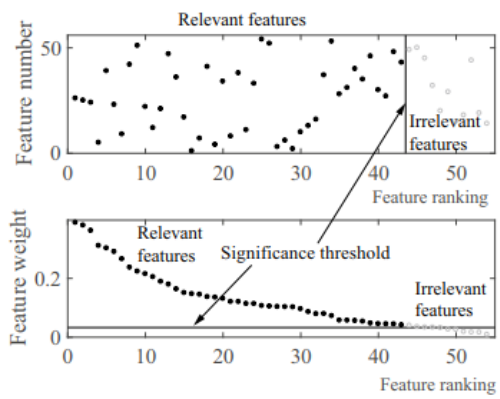


Fig. 5. Feature ranking and selection using *ReliefF*.

Redundant features are eliminated by the *HCR* algorithm [76] with a high-correlation threshold of $\delta = 0.80$, and using the *ReliefF* ranking to keep the most relevant feature from a set of highly correlated ones. The algorithm eliminated 20 highly-correlated features. The feature space was reduced to 23 relevant features without high-correlations.

4.1.2. Step 2: Model creation

The training set is used to fit 100 regularized *LR* models. The *LASSO* method was applied to estimate the fitted least-squares regression coefficients for a set of 100 regularization coefficients λ, starting with the largest value of λ that gives a non-null model. However, the non-null model is not included in the analysis since its estimated *MPCD* equals zero. Figure 6a displays each candidate model's associated value of λ, Fig. 6b the number of features, and Fig. 6c γ, the associated classification threshold – value obtained with respect to *MPCD*, Fig. 6d shows this search using the optimal classification threshold algorithm, *OCTM* [76].

The number of features decreases as the value of λ increases. Selecting the right model is one of the main challenges.

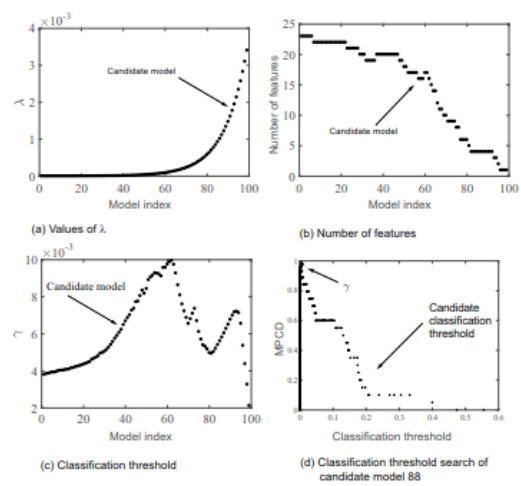
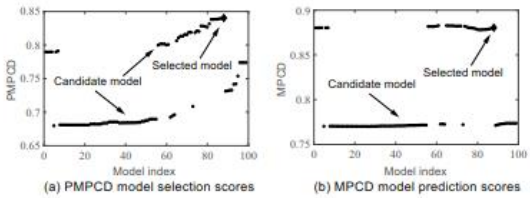


Fig. 6. Candidate model information.

4.1.3. Step 3: Model selection/validation

The validation set is used to estimate the *PMPCD* and *MPCD*. According to *PMPCD*, model 88 – together with its associated classification threshold – is the best candidate (classifier), with an estimated score of 0.8403, Fig. 7a, and *MPCD* of 0.8805 ($\alpha = 0.0095, \beta = 0.1111$), Fig. 7b, and 4-relevant features. The value of γ for this model is 0.0063, meaning that any value estimated by the logistic function below this threshold will be classified as 0 (i.e., *good*), or 1 (i.e., *bad*) otherwise. Varying the values of λ helped to identify the most relevant features.



(a) PMPCD model selection scores (b) MPCD model prediction scores

Fig. 7. Model selection and validation.

4.1.4. Step 4: Prediction optimization

For the non-ensemble notation, the *final model* is called $Classifier(C_i)$, $C_i \in \{0,1\}$ and $i = 1,...,7$, the binary outcome is called $Label(L_{f(i,...,o)})$, where $L_{f(i,...,o)} \in \{0,1\}$, $i,...,o \in \{1,...,7\}$ and $f(i,...,o)$ defines the decision combination rule. The prediction ability on the validation set of the C_is of each of the seven *MLA* considered by *Big Models* is summarized in Table 3; however, due to space limitations, the learning process of the remaining 6 is not reported. Table 4 summarizes the defect detection by C_i.

Table 3. Prediction ability analysis of the selected models (*validated models)

Classifier	*MLA*	*FN*	*FP*	*TN*	*TP*	*MPCD*
C_1	*SVM*	1	37	12190	8	0.8862*
C_2	*ANN*	1	82	12145	8	0.8829*
C_3	*LR*	1	116	12111	8	0.8805*
C_4	*NB*	2	38	12189	7	0.7754
C_5	*RF*	2	64	12163	7	0.7737
C_6	*KNN*	7	0	12227	2	0.2222
C_7	*SVM(RBF)*	7	7	12220	2	0.2221

Table 4. Detection by classifier.

Classifier	$bad_\#$								
	1	2	3	4	5	6	7	8	9
C_1	X	✓	✓	✓	✓	✓	✓	✓	✓
C_2	X	✓	✓	✓	✓	✓	✓	✓	✓
C_3	X	✓	✓	✓	✓	✓	✓	✓	✓
C_4	X	✓	✓	✓	✓	✓	✓	X	✓
C_5	X	✓	✓	✓	✓	✓	✓	X	✓
C_6	X	X	✓	X	X	✓	X	X	X
C_7	X	X	✓	✓	X	X	X	X	X

All of them failed to detect the bad_1, whereas C_1-C_3 detected the same defects – bad_2-bad_9. Since C_4-C_7 showed lower detection performance, and they failed to detect the bad_1, they are not considered in the non-ensemble. For which, the main goal is to reduce the α error through a decision combination rule. All possible model combinations are evaluated: (1) for a 2-classifier combination $min(C_i, C_j)$ rule is followed, both classifiers must call a unit *bad* to be classified as *bad*; otherwise it is classified *good*, eqn. 5; and (2) for a 3-classifier combination a simple majority vote $(C_i + C_j + C_k) > 1$, eqn. 6, Table 5.

$$L_{ij} = min(C_i, C_j) \tag{5}$$

$$L_{ijk} = \begin{cases} 1 & (C_i + C_j + C_k) > 1 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Table 5. *MPCD* by non-ensemble option.

C_1-C_2	C_1-C_3	C_2-C_3	C_1-C_2-C_3
0.8870	0.8869	0.8839	0.8837

Based on *MPCD*, Table 5, only classifiers C_1-C_2 should be included in the non-ensemble, with a *min* decision combination rule. Although the C_1 has similar performance, it is outperformed by the non-ensemble, and from prediction perspective, a multimodel-based prediction, tends to be more robust [54]. If any of the lower performer classifiers (C_4-C_7) would have detected the bad_1, a decision combination rule such as *max* would have the potential to improve detection at the expense of α.

4.1.5. Step 5: Model testing

The importance of this final step is to assess the non-ensemble without the induced bias in the training and validation stages, and to ensure that the learning targets are satisfied, and therefore, decide if the predictive system can be deployed. Table 6 shows the confusion matrix of the non-ensemble.

Table 6. *MCS* confusion matrix.

	Declare good	Declare bad
good	9490	3
bad	0	7

The *MCS* correctly detected the seven *bad* units in the testing set, with only three *FP*, *MPCD* = 0.9997. From this analysis, it can be concluded that the two classes are virtually separable, and therefore, the non-ensemble can deployed with high confidence.

4.1.6. Big Data-Big Models to engineering

High level knowledge can be extracted through *Big Data-Big Models* which can be used for redesigning manufacturing processes[2]. The most relevant information from engineering perspective, was provided by fea-

[2] Due to confidentiality reasons, this subsection is addressed in a high level.

ture 26, which was ranked as the one with the most discriminative power by *ReliefF*, Fig. 5. This feature represented the slope of the power curve at the beginning of the welding process, due to signal plotting (and feature relevance analysis), it was found that low quality welds tend to have slow slope, which most of the times resulted in a bad quality weld. A power booster was an option to increase the slope of welds with slow slope at the beginning of the process. Although model interpretation is not the main focus of this paper, the herein presented preliminary feature/model analysis, illustrates how *Big Data-Big Models* may provide valuable information for defect root cause analysis and process redesign.

4.2. Sensorless Drive Diagnosis Case

To evaluate the proposed approach from machine learning perspective and reproducibility purposes, another case study is presented using a public data set – Sensorless Drive Diagnosis [78]. The data set contains 48 numerical features (plus the class label), which are extracted from motor current [79], the motor has intact and defective components. This results in 11 different classes with different conditions. The goal of this study is to detect only class one. This data set is highly unbalanced too (58509 instances - including 5319 class 1). *Big Models* are constructed following a hold-out validation method: training set (34,309 - including 3,119), validation set (12,100 - 1,100), and testing set (12,100 - 1,100). Since the dataset does not provide specific information about the meaning/name of each feature, they are referred to as feature 1,2,...48.

The data is preprocessed using the *ReliefF* and *HCR* algorithms, feature relevance is obtained with $k = 10$ and $\tau = 0.0185$, ($\alpha = 0.05$, and $m = 58,509$), highly correlated features are eliminated based on $\delta = 0.90$, results are shown in Table 7.

Table 7. Feature selection information.

Feature description	Features
Irrelevant	1-6,13-18, 25-48
Redundant	8,7,12,10,20,19,24,23,22
Selected	9,11,21

Because irrelevant and redundant features are not helpful for classification, only relevant and not highly correlated features 9, 11 and 21 (selected - Table 7) are used in the modeling stage. For each *MLA* selected features are used to create many models to tune their associated hyperparameter (e.g., number of neurons, neighbors, trees, λ, etc). The *MPCD* and *PMPCD* are used as a measure of classification performance and model selection criterion respectively. The prediction ability on the validation set of the C_is of each of the seven *MLA* is summarized in Table 8; however, due to space limitations, their learning process is not reported.

Table 8. Prediction ability analysis of the selected models (* Validated models, + RBF)

Classifier	*MLA*	*FN*	*FP*	*TN*	*TP*	*MPCD*
C_1	*KNN*	7	29	10971	1093	0.991*
C_2	*SVM+*	25	301	10699	1075	0.951*
C_3	*ANN*	15	668	10332	1085	0.927*
C_4	*RF*	36	896	10104	1064	0.889
C_5	*NB*	42	1007	9993	1058	0.874
C_6	*LR*	56	2016	8984	1044	0.775
C_7	*SVM*	305	521	10479	795	0.689

As shown in Table 8, the *KNN* algorithm is the one with the highest prediction performance, $MPCD = 0.9910$. For this particular algorithm, $k = 1, 3, ..., 49$ were explored, where $k = 45$ shows the highest value.

In contrast with the previous case study, different classifiers committed different miss-classifications, therefore, more decision combinations are evaluated for prediction optimization. Predicted labels – validation set – of classifiers C_1-C_3 and decision combination rules described in Eqs. 5, 6, 7, 8, 9 are explored. Since no decision combination shows higher prediction performance than the *KNN* algorithm, this model is used as the classifier, its unbiased generalization ability on the testing set is reported in Table 9, $MPCD = 0.9948$.

$$L_{ijk} = \begin{cases} 1 & (C_i + C_j + C_k) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

$$L_{ijk} = \begin{cases} 1 & (C_i + C_j + C_k) > 2 \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

$$L_{ij} = max(C_i, C_j) \tag{9}$$

Table 9. *KNN* confusion matrix.

	Declare good	Declare bad
good	10973	27
bad	3	1097

The analyzed data set used for this case study is highly unbalanced with many irrelevant and redundant features. With no prior knowledge about the underlying system that generated the Sensorless Drive Diagnosis data, the pattern recognition problem was effectively solved by the *Big Models* learning paradigm. Three relevant inferences can be drawn about the system: (1) the pattern can be efficiently recognized by only three features; (2) the system is not linear, since the non-linear

algorithms show significantly better performance; and (3) since the *KNN* is a very stable algorithm, it can be deployed with $k = 45$ to detect class 1, this algorithm with its associated hyperparameter can virtually separate the classes.

4.3. Discussion

In both case studies the *Big Models* learning paradigm efficiently solved the pattern recognition problem. In the first case study, the pattern was better approximated by linear classifiers and the prediction was optimized through a non-ensemble combination, whereas in the second case study, the pattern was better recognized by non-linear algorithms and no decision combination outperformed a single model. Case studies exhibited the capacity of *Big Models* to solve a wide spectrum of binary classification problems. Parsimonious modeling helped to identify the most relevant features, – although out of the scope of this paper – information that can be used to extract high level knowledge about the system.

5. Conclusions

Binary classification is a very broad topic, with a plethora of options. Poor domain knowledge can lead to a low quality solution. A good starting point based on theory and empirical evidence was presented here.

This paper addressed the learning problem of *PMQ*, which is solved by *Big Models*. A learning paradigm that requires many models to be developed to find the *final model*. Manufacturing pattern recognition problems pose several particular challenges. The theoretical aspect of modeling these data sets was reviewed. A learning scheme was developed based on seven *MLA*, which have the capacity to solve a wide spectrum of binary classification problems.

Big Models follow an iterative process of searching the data for patterns. The pattern recognition problem can be solved by the right choice of the learning algorithm and its associated hyperparameter, and prediction can be optimized through a decision combination scheme. Since it is not known in advance which algorithm (and hyperparameter value) is the best, many models need to be created. However, this iterative model creation approach can lead to the development of spurious models. The proposed learning scheme takes into consideration this potential risk, to avoid this problem in the *final model*. And therefore, if the target goals are met, the predictive system can be deployed with confidence.

Due to the relevance of model interpretation in manufacturing, the learning scheme is founded in parsimonious modeling; with the ability to deal with many features – including irrelevant and redundant – and highly unbalanced data.

Big Models have the ability to mine massive amounts of data to learn manufacturing systems. Predictive model and derived knowledge, have the capacity to solve complex manufacturing problems that cannot be efficiently solved by traditional quality methods. And therefore, potentially move the process learning curve off the plateau of no progress.

5.1. Future work

This paper included a basic decision combination analysis to solve the prediction stage of the *PMQ* problem solving strategy, in which the objective was to find the optimal combination of classifiers with respect to prediction *MPCD* using different rules. However, the complexity (number of features) associated to each classifier should be considered to induce parsimony at a decision combination level. A mathematical formulation that considers prediction and complexity when creating a non-ensemble can be a future research along this path.

The relearning problem poses another challenge. Since manufacturing systems are dynamic in nature, *Big Models* need to be constantly updated, but the research challenge is to develop a monitoring system to detect when the classifiers have lost their prediction capacity.

Deep Learning comprises a set of algorithms that automatically discover significant features/representations to develop a detection or classification task. In some applications, defects can be detected through image recognition, in this context, this learning technique is also a future work to be considered.

References

[1] J. A. Abell, D. Chakraborty, C. A. Escobar, K. H. Im, D. M. Wegner, and M. A. Wincek, "Big Data Driven Manufacturing — Process-Monitoring-for-Quality Philosophy," *ASME J of Manufacturing Science and Eng on Data Science-Enhanced Manufacturing*, vol. 139, no. 10, 2017.

[2] R. Malhotra, "A Systematic Review of Machine Learning Techniques for Software Fault Prediction," *Applied Soft Computing*, vol. 27, pp. 504–518, 2015.

[3] G. Köksal, İ. Batmaz, and M. C. Testik, "A Review of Data Mining Applications for Quality Improvement in Manufacturing Industry," *Expert systems with Applications*, vol. 38, no. 10, pp. 13 448–13 467, 2011.

[4] V. Uraikul, W. Chan, and P. Tontiwachwuthikul, "Artificial Intelligence for Monitoring and Supervisory Control of Process Systems," *Eng Applications of Artificial Intelligence*, vol. 20, no. 2, pp. 115–131, 2007.

[5] C. A. Escobar, M. A. Wincek, D. Chakraborty, and R. Morales-Menendez, "Process-monitoring-for-qualityapplications," *Manufacturing Letters*, vol. 16, pp. 14–17, 2018.

[6] R. D. Moen and C. L. Norman, "Circling Back," *Quality Progress*, vol. 43, no. 11, p. 22, 2010.

[7] C. A. Escobar and R. Morales-Menendez, "Machine Learning and Pattern Recognition Techniques for Information Extraction to Improve Production Control and Design Decisions," in *P. Perner Advances in Data Mining, ICDM*. Springer Verlag, 2017, pp. 285–295.

[8] T. G. Dietterich, "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.

[9] Z. Zhou, "Ensemble Learning," *Encyclopedia of Biometrics*, pp. 411–416, 2015.

[10] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann, "Review of Classifier Combination Methods," in *Machine Learning in Document Analysis and Recognition*. Springer, 2008, pp. 361–386.

[11] D. H. Wolpert, "The Lack of a Priori Distinctions Between Learning Algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996.

[12] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design*. Martin Hagan, 2014.

[13] J. De Brabanter, K. Pelckmans, J. A. Suykens, B. De Moor, and J. Vandewalle, "Robust Complexity Criteria for Nonlinear Regression in NARX Models," in *Proc of the* 13th *System Identification Symp*, 2003, pp. 79–84.

[14] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[15] I. Davidson, "An Ensemble Technique for Stable Learners with Performance Bounds," in *AAAI*, vol. 2004, 2004, pp. 330–335.

[16] R. Michalski, J. Carbonell, and T. Mitchell, *Machine Learning: An Artificial Approach — Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression*. Berlin Heidelberg: Springer-Verlag, 2013.

[17] S. Ray, "Understanding Support Vector Machine Algorithm From Examples (Along With Code)," July 7 2017. [Online]. Available: www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code

[18] R. Moraes, J. Valiati, and N. Gavião, "Document-Level Sentiment Classification: An Empirical Comparison Between SVM and ANN," *Expert Systems with Applications*, vol. 40, pp. 621–633, 2013.

[19] P. Valente Klaine, M. Ali Imran, O. Onireti, and R. Demo Souza, "A Survey of Machine Learning Techniques Applied to Self Organizing Cellular Networks," *IEEE Comm Surveys & Tutorials*, p. 1, 2017.

[20] S. Ferdowsi, S. Voloshynovskiy, M. Gabryel, and M. Korytkowski, "Multi-class Classi cation: A Coding Based Space Partitioning," in *Int Conf on Artificial Intelligence and Soft Computing*, 2014, pp. 593–604.

[21] S. Kotsiantis, I. Zaharakis, and P. Pintelas, "Machine Learning: A Review of Classification and Combining Techniques." *Artificial Intelligence Review*, vol. 26, pp. 159–190, 2006.

[22] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm Support Vector Machines," *Advances in NIPS*, vol. 16, no. 1, pp. 49–56, 2004.

[23] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature Selection for SVMs," in *NIPS*, vol. 12, 2000, pp. 668–674.

[24] G. Fung and O. Mangasarian, "A Feature Selection Newton Method for Support Vector Machine Classification," *Computational Optimization and Applications*, vol. 28, no. 2, pp. 185–202, 2004.

[25] P. Bradley and O. Mangasarian, "Feature Selection via Concave Minimization and Support Vector Machines," in *ICML*, vol. 98, 1998, pp. 82–90.

[26] S. Du, C. Liu, and L. Xi, "A Selective Multiclass Support Vector Machine Ensemble Classifier for Engineering Surface Classification using High Definition Metrology," *J. of Manufacturing Science and Eng*, vol. 137, no. 1, p. 011003, 2015.

[27] S. Lee, H. Lee, P. Abbeel, and A. Ng, "Efficient L_1 Regularized Logistic Regression," in *Proc of the National Conf on Artificial Intelligence*, vol. 21, no. 1. Cambridge, MA, 2006, p. 401.

[28] R. Tibshirani, "Regression Shrinkage and Selection via the LASSO," *J. of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[29] R. Zakharov and P. Dupont, "Ensemble Logistic Regression for Feature Selection," *Pattern Recognition in Bioinformatics*, pp. 133–144, 2011.

[30] S. B. Imandoust and M. Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background," *Int. J of Eng Research and Applications*, vol. 3, no. 5, pp. 605–610, 2013.

[31] A. G. Karegowda, M. Jayaram, and A. Manjunath, "Combining Akaikes Information Criterion (AIC) and the Golden-Section Search Technique to Find Optimal Numbers of K-Nearest Neighbors," *Int J of Computers and Applications*, vol. 2, pp. 80–87, 2010.

[32] L. Li, D. M. Umbach, P. Terry, and J. A. Taylor, "Application of the GA/KNN Method to SELDI Proteomics Data," *Bioinformatics*, vol. 20, no. 10, pp. 1638–1640, 2004.

[33] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[34] M. Aly, "Survey on Multiclass Classification Methods," *Neural Nets*, vol. 19, 2005.

[35] A. Izenman, *Modern Multivariate Statistical Techniques. Regression, Classification, and Manifold Learning*, G. Casella, S. Fienberg, and I. Olkin, Eds. Springer Science+Business Media, 2013.

[36] S. Sharma, J. Agrawal, S. Agarwal, and S. Sharma, "Machine Learning Techniques for Data Mining: A Survey." in *Int Conf on Computational Intelligence and Computing Research*, 2013, pp. 1–6.

[37] X. Wu, V. Kumar, Q. Ross, J. Ghosh, Q. Yang, H. Motoda, and D. Steinberg, "Top 10 Algorithms in Data Mining," *Knowledge and Information Systems*, vol. 14, pp. 1–37, 2008.

[38] S. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[39] Y. Liu and M. Yuan, "A Comparative Study of Classification Algorithms for Spam Email Data Analysis," *Int J on Computer Science and Eng*, pp. 1890–1895, 2011.

[40] H. Deng and G. Runger, "Feature Selection via Regularized Trees," in *Int Joint Conf on Neural Networks*, 2012, pp. 1–8.

[41] A. K. B. Baharudin, L. H. Lee, and K. Khan, "A Review of Machine Learning Algorithms for Text-Documents Classification." *J of Advances in Information Technology*, vol. 4, pp. 1–1, 2010.

[42] Analytics Vidhya Content Team, "A Complete Tutorial on Tree Based Modeling from Scratch (in R and Python) ," 2017. [Online]. Available: www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python

[43] T. G. Dietterich *et al.*, "Ensemble Methods in Machine Learning," *Multiple classifier systems*, vol. 1857, pp. 1–15, 2000.

[44] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[45] K. Al-Aidaroos, A. A. Bakar, and Z. Othman, "Naive Bayes Variants in Classification Learning," in *Int Conf on Information Retrieval and Knowledge Management: Exploring the Invisible World*, 2010, pp. 276–281.

[46] D. Pham and A. Afify, "Machine-Learning Techniques and Their Applications in Manufacturing," in *In Proc of the Institution of Mechanical Engineers*, vol. 219, 2005, pp. 395–412.

[47] A. F. Amato nd A. López, E. P. na Méndez, P. Vaăhara, A. Hampl, and J. Havel, "Artificial Neural Networks in Medical Diagnosis ," *J of Applied Biomedicine*, vol. 11, no. 2, pp. 47–58, 2013.

[48] F. Burden and D. Winkler, "Bayesian Regularization of Neural Networks," *Artificial Neural Networks: Methods and Applications*, pp. 23–42, 2009.

[49] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer, 2013, vol. 810.

[50] S. Arlot and A. Celisse, "A Survey of Cross-Validation Procedures for Model Selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[51] T. Hastie, J. Friedman, and R. Tibshirani, "Model Assessment and Selection," in *The elements of statistical Learning*. Springer, 2001, pp. 193–224.

[52] R. Shibata, "Statistical Aspects of Model Selection," *From Data to Model*, pp. 215–240, 1989.

[53] L. Huan and H. Motoda, "Feature Extraction, Construction and Selection: A Data Mining Perspective," 1998.

[54] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference: a Practical Information-Theoretic Approach*. Springer Science & Business Media, 2003.

[55] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty, "Optimal Number of Features as a Function of Sample Size for Various Classification Rules," *Bioinformatics*, vol. 21, no. 8, pp. 1509–1515, 2004.

[56] H. E. Tinsley and S. D. Brown, *Handbook of Applied Multivariate Statistics and Mathematical Modeling*. Academic Press, 2000.

[57] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-based Filter Solution," in *ICML*, vol. 3, 2003, pp. 856–863.

[58] Y. Liu and V. Gopalakrishnan, "An Overview and Evaluation of Recent Machine Learning Imputation Methods Using Cardiac Imaging Data," *Data*, vol. 2, no. 1, p. 8, 2017.

[59] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Int Conf on Machine Learning*, 2015, pp. 448–456.

[60] I. B. Mohamad and D. Usman, "Standardization and its Effects on K-means Clustering Algorithm," *Research J of Applied Sciences, Eng and Technology*, vol. 6, no. 17, pp. 3299–3303, 2013.

[61] M. Hall, "Correlation-based Feature Selection of Discrete and Numeric Class Machine Learning," in *Proc of the* 17th *Int Conf on Machine Learning*. University of Waikato, 2000, pp. 359–366.

[62] K. Nicodemus and J. Malley, "Predictor Correlation Impacts Machine Learning Algorithms: Implications for Genomic Studies," *Bioinformatics*, vol. 25, no. 15, pp. 1884–1890, 2009.

[63] F. Wang, Y. Yang, X. Lv, J. Xu, and L. Li, "Feature Selection using Feature Ranking, Correlation Analysis and Chaotic Binary Particle Swarm Optimization," in 5th *Int Conf on Software Eng and Service Science*, 2014, pp. 305–309.

[64] C. Shao, K. Paynabar, T. Kim, J. Jin, S. Hu, J. Spicer, H. Wang, and J. Abell, "Feature Selection for Manufacturing Process Monitoring using Cross-Validation," *J. of Manufacturing Systems*, vol. 10, 2013.

[65] P. M. Lukacs, K. P. Burnham, and D. R. Anderson, "Model Selection Bias and Freedmans Paradox," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 117–125, 2010.

[66] G. Chandrashekar and F. Sahin, "A Survey on Feature Selection Methods," *Computers & Electrical Eng*, vol. 40, no. 1, pp. 16–28, 2014.

[67] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[68] C. A. Escobar and R. Morales-Menendez, "Process-Monitoring-for-Quality — A Model Selection Criterion," *Manufacturing Letters*, vol. 15, pp. 55–58, 2018.

[69] 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

[70] D. M. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation," *J of Machine Learning Technologies*, vol. 2, pp. 37–63, 2011.

[71] Z. Zhou, "Ensemble Learning," in *Encyclopedia of Biometrics*. Springer, 2009, pp. 270–273.

[72] H. Kang and J. Kim, "A Probabilistic Framework for Combining Multiple Classifiers at Abstract Level," in *Proc of the* 4th *Int Conf on Document Analysis and Recognition*, vol. 2, 1997, pp. 870–874.

[73] H. J. Kang and J. H. Kim, "Dependency Relationship based Decision Combination in Multiple Classifier Systems," in *Proc of* 14th *Int Joint Conf on Artificial Intelligence*, vol. 2, Montreal, Canada, 1995, pp. 1130–1136.

[74] T. Ho, J. Hull, and S. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[75] P. Domingos, "A Few Useful Things to Know about Machine Learning," *Comm of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[76] C. A. Escobar and R. Morales-Menendez, "Machine Learning Techniques for Quality Control in High Conformance Manufacturing Environment," *Advances in Mechanical Eng*, vol. 10, no. 2, 2018.

[77] M. Robnik-Šikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[78] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[79] F. Paschke, C. Bayer, M. Bator, U. Mönks, A. Dicks, O. Enge-Rosenblatt, and V. Lohweg, "Sensorlose Zustandsüberwachung an Synchronmotoren," in *Proc 23th. Workshop Computational Intelligence, Dortmund, Germany 5*, 2013, p. 211.

Chapter 5

Process-Monitoring-for-Quality – Applications

PMQ makes a limited statement about the quality of a manufactured item when a direct measurement of the quality is not practical or not possible. *PMQ* uses real time manufacturing process data to declare an item as either ‘‘good” or ‘‘suspect/bad”. Traditional quality techniques, such as *SPC,* require first principle understanding of the process and a mature understanding of the product and a methodology for verifying the quality of each manufactured item. When these requirements are not met, as shown in this paper, the process cannot be effectively controlled.

Though the context of the development and application of *PMQ* were very specific, the potential applications are broader with the capacity to solve *SPC*-intractable problems. *PMQ* enhances and extends standard *SPC* with an empirical predictive framework by addressing three incomplete background knowledge situations that *SPC* or traditional quality control cannot: (1) *infeasible measurement problem*, (2) *the delayed measurement problem,* and (3) *the uncertain charactersitc problem.* The infeasible measurement problem occurs when a physical characteristic exists and is measurable; however, it may not be feasible to measure it within the time constraints of the manufacturing process. The delayed measurement problem occurs when there is a delay in the time of measurement and the buffer has a large capacity. All items in the buffer are suspect when a problem is detected at the measurement station. Extra time and expense are required to test and possibly repair or scrap the items. Finally, the uncertain characteristic problem occurs when the quality control procedure cannot even begin because the quality characteristic is ‘‘uncertain”.

This study also illuminated three specific applications that are worth of future investigation: (1) *multiple testing*, (2) *learning catalyst*, and (3) *usage monitoring for quality* (UMQ). For the first, the *PMQ* rule can be used with the traditional quality test to form a new test comprised of two sub-tests. In the second, *PMQ* can be divorced from the immediate process and be used for continuous learning about the process. For the third, *SPC* is often thought of as only an intra-plant activity, but the *BDBM* environment demonstrates that it is also an extra-plant activity because a manufactured item’s performance data can be (and often is) collected on the item during its service life. These three applications have also the potential to enhance the quality movement. *PMQ* is an addition to *SPC*, since it does not rely on the first principle understanding of the process though it is equipped to exploit any such knowledge. Modern plants generate large volumes of data from processes that are complex and not necessarily understood from the first principle perspective. In such situations, *PMQ* and its potential extensions give new meaning to and opportunities for continuous improvement and control in manufacturing.

Manufacturing Letters 16 (2018) 14–17

Contents lists available at ScienceDirect

Manufacturing Letters

journal homepage: www.elsevier.com/locate/mfglet

Letters

Process-Monitoring-for-Quality—Applications

Carlos A. Escobar [a,b,*], Michael A. Wincek [a], Debejyo Chakraborty [a], Ruben Morales-Menendez [b]

[a] Global Research & Development, General Motors, Warren, MI, USA
[b] Tecnológico de Monterrey, Monterrey, NL, Mexico

ARTICLE INFO

Article history:
Received 16 September 2017
Received in revised form 23 January 2018
Accepted 3 February 2018

Keywords:
Manufacturing
Big Data
PMQ
Process Monitoring
Quality

ABSTRACT

Innovation and the marketplace have been pushing *Statistical Process Control (SPC)* outside its comfort zone, which requires a mature understanding of the product and process and a methodology for verifying the quality of each manufactured item. Especially when a new technology is proven to work and customer interest is high, companies want manufacturing to respond to the uncertainties. *Process Monitoring for Quality (PMQ)* is a strategy, based on the empirical learning and data gather capabilities of the Big Data environment, that addresses this challenge while verifiably producing quality product. *PMQ* offers opportunities for learning and quality improvement: it enhances the quality movement by addressing three quality problems that SPC or traditional quality control techniques cannot; and by illuminating future applications.

1. Introduction

Big data [1], Industrial Internet of Things [2], acsensorization [3], artificial intelligence [4], machine learning [5], and cyber-physical systems [6] are propelling smart manufacturing. Many technical and organizational challenges of smart manufacturing must be addressed in order to realize gains over the entire value network [1,7]. This letter identifies some challenges to traditional quality control that can now be addressed by *PMQ*, which was first introduced in [3].

PMQ is a big data-driven quality philosophy that makes a limited statement about the quality of a manufactured item when a direct measurement of the quality is not practical or not possible. The strategy originated from the *Big Data–Big Model* (*BDBM*) point of view described in [3] that was used to develop an initial quality monitoring process for the ultrasonic welding of battery tabs in the Chevrolet® Volt. The strategy uses real time manufacturing process data to declare an item as either "good" or "suspect". Though the context of the development and application of PMQ were very specific, the potential applications are broader. Application of standard *SPC* has three requirements: a mature understanding of the process, an observable relevant quality characteristic with associated quality criteria, and a strategy and process to verify the quality criteria at the plant in real time. Today's innovative manufacturing environment and competitive business environment sometimes force the launching of a product even though the above three requirements for an *SPC* program cannot be satisfied. Fig. 1 shows how standard *SPC* is "brittle"; it "breaks" if not all steps are possible or not all steps are successfully completed. The purpose of this letter is to indicate how *PMQ* enhances and extends standard *SPC* by addressing three incomplete background knowledge situations that *SPC* or *traditional quality control* [8] cannot.

2. Background

SPC uses first principle knowledge, engineering technology, and statistical tools to control a manufacturing process under a well-understood cause and effect framework. When this framework is missing or incomplete, *PMQ* supplements it with an empirical predictive framework based on statistics, machine learning, and optimization. *SPC* requires known product quality characteristics that are measurable within the temporal and physical constraints of the manufacturing plant environment. Fig. 2 provides a taxonomy of quality features in manufacturing and relates them in a path diagram, which is list of the numbered nodes separated by an arrow, "→". The path that describes a conventional quality control initiative is $(1 \rightarrow \{3,4\} \rightarrow 6 \rightarrow 9 \rightarrow 12)$. This path relies on a quality characteristic that is known and physically observable either by

Abbreviations: BDBM, Big Data–Big Models; NDE, non-destructive evaluation; PMQ, process monitoring for quality; SPC, statistical process control; UMQ, usage monitoring for quality.

* Corresponding author at: Global Research & Development, General Motors, Warren, MI, USA.
E-mail address: carlos.1.escobar@gm.com (C.A. Escobar).

https://doi.org/10.1016/j.mfglet.2018.02.004
2213-8463/

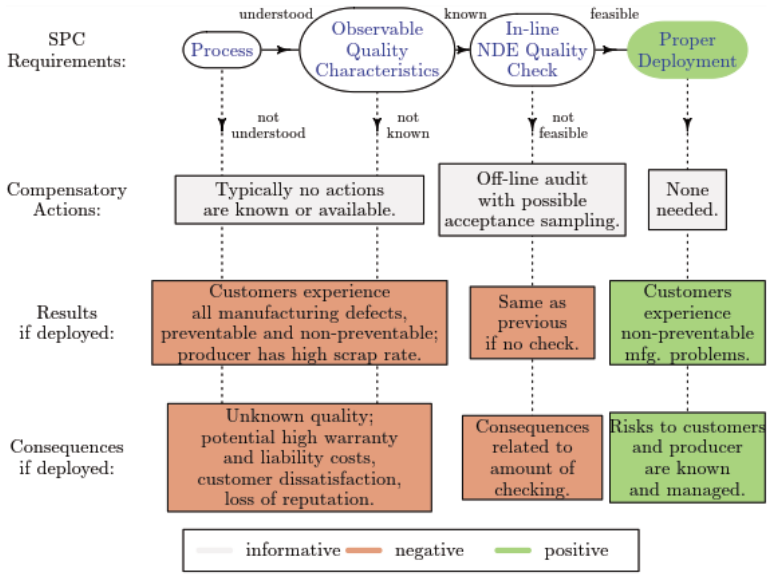


Fig. 1. Challenges to *SPC*.

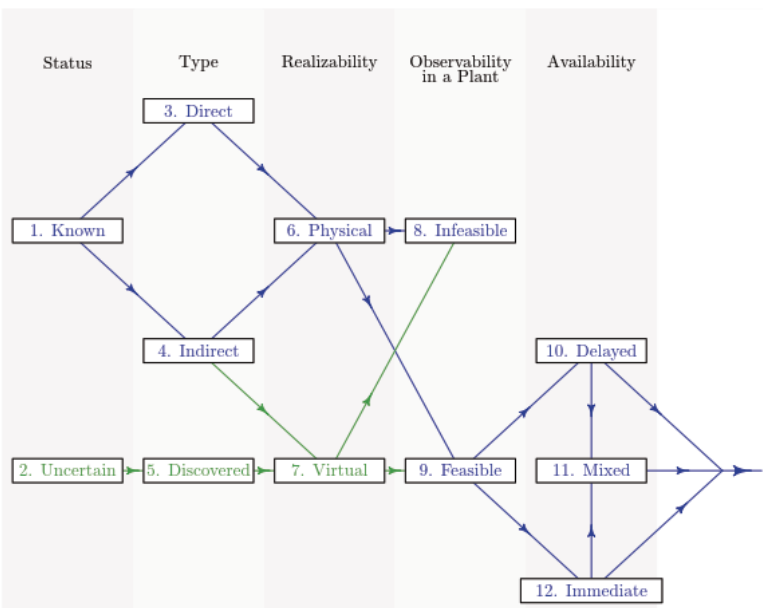


Fig. 2. Taxonomy of quality features in manufacturing.

direct or indirect means such that it is feasible in a manufacturing line and can immediately provide a good/bad quality status. Sometimes the characteristic cannot be measured directly. This situation can occur when the value of the characteristic involves destroying the item. When the characteristic cannot be measured directly, theory or engineering practice often provide a proxy through which the desired value can be obtained indirectly.

Even when a physical characteristic exists and is measurable, it may not be feasible to measure it within the time constraints of the manufacturing process. We call this the *infeasible measurement problem*: $(1 \rightarrow \{3,4\} \rightarrow 6 \rightarrow 8)$. The preferred scenario is for the characteristic to be measured immediately after the item is produced, $(1 \rightarrow \{3,4\} \rightarrow 6 \rightarrow 9 \rightarrow 12 \rightarrow)$. When there is a delay in the time of measurement and the buffer has a large capacity, all

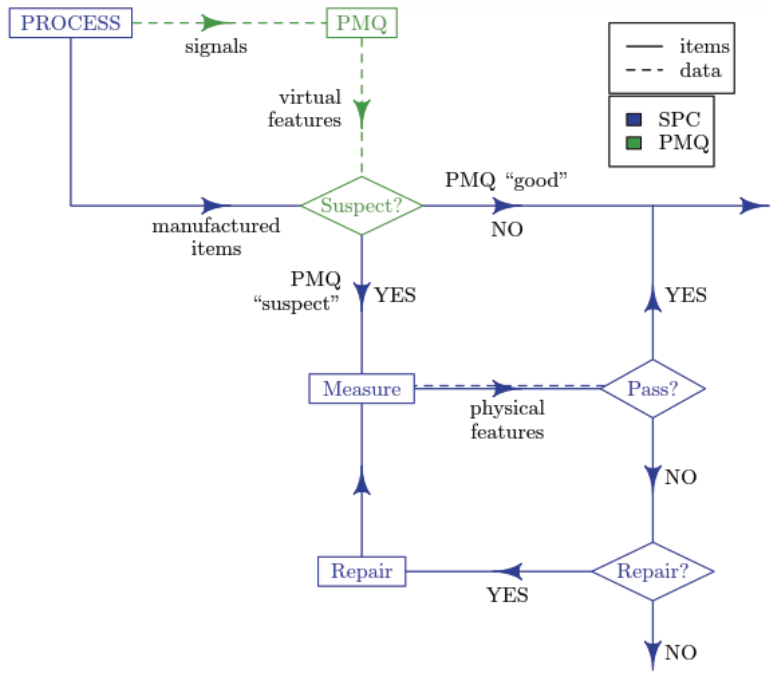


Fig. 3. *SPC* and *PMQ* quality control framework.

items in the buffer are suspect when a problem is detected at the measurement station. Extra time and expense are required to test and possibly repair or scrap the items. We call this the *delayed measurement problem*: $(1 \rightarrow \{3, 4\} \rightarrow 6 \rightarrow 9 \rightarrow 10)$.

Development of a quality control procedure cannot even begin when the quality characteristic is "uncertain". When manufacturing is not allowed to commence until a quality procedure is in place, new product launches may be delayed and early-to-market advantage may be lost, even when the product has high quality. We call this the *uncertain characteristic problem*: $(2 \rightarrow)$. When manufacturers proceed anyway, they accept the consequences, Fig. 1.

3. *PMQ*-enabled opportunities and exercisable options

Conventional quality control is built on a mature knowledge of the product and process and a plant-capable measurement technology. *PMQ* is an extension of traditional *SPC*, Fig. 3, and it partially alleviates the three problems mentioned in the previous section. *PMQ* uses the *BDBM* style of thinking to extract features from signals and then to create binary classification models that declare items as "good" or "suspect". The feature creation and model building is usually done off-line to produce a rule[1] that is deployed on-line. Since the features and the rule are products of the *BDBM* learning algorithms, the resulting features are called "virtual" because essentially they are mathematical quantities designed for their predictive power, not their explanatory power. Hence, they may not have a direct physical interpretation. Virtual features can be thought of as "weaker" than the usual physical features, but the two together address the three stated problems and offer other possibilities.

[1] Whereas a classifier is a procedure for predicting the class of an item, we use the term "rule" to designate a procedure with all its parameters and unknowns completely specified so that it can be deployed.

The *delayed measurement problem* can obviously be avoided or eliminated by testing the item immediately after it is processed. When that is not possible, *PMQ* offers the possibility of creating a rule than can give an early warning of a problem so that the potential of a buffer full of defective items is diminished or eliminated.

PMQ provides two possible remedies for the *infeasible measurement problem*: (1) rely totally on the rule to declare an item as either "good" or "bad", when the classifier has the required false alarm (type I error) and miss (type II error) rates; and (2) use *PMQ* to reduce the number of items that must be measured, as it did in [3]. If the product has high conformance, a *PMQ* rule may be able to declare enough items as "good" so that the line rate is not affected while allowing the "suspect" items to be tested and then returned to the line when they are found to be in conformance. This strategy involves paths through the "Mixed" node (11) in Fig. 2.

PMQ enables *SPC* to commence even when theory and engineering practice do not provide a quality characteristic. This can occur when the technology is new and incompletely understood so that standards and specifications have not yet been developed. Typically some method of quality verification is available, otherwise management would not manufacture the item, but such a method is typically infeasible or impractical in some way: time consuming, inefficient, or costly. *PMQ* can address the *uncertain characteristic problem* in a manner similar to that in the *infeasible measurement problem*. The virtual features in the classifier can be used to reduce the number of items tested by the impractical method. The number of items that travel down the *PMQ* "good" path in Fig. 3 have to be sufficiently numerous to keep the line running and simultaneously to allow the "suspect" to be measured and possibly reinserted into the assembly line.

The virtual features and resulting rule play an important but auxiliary role in addressing the above three problems. Especially for the latter two problems, they may be seen as providing a

temporary solution until sufficient knowledge has been acquired. The *PMQ* approach allows the manufacturing process to continue and still ensure quality for the customer without the usually required mature understanding of the product and process.

PMQ can enhance the quality movement in three other ways:

1. **Multiple testing** The *PMQ* rule can be used with the traditional quality test to form a new test comprised of two sub-tests: the first test using the virtual features and the second test using the physical features. A number of strategies are possible such as only retest the "suspects". The choice of an appropriate strategy and rule requires finding the error rates of the feasible retest–rule.
2. **Learning catalyst** *PMQ* can be divorced from the immediate process and be used for continuous learning about the process. This is helpful when the traditional theory based approach is not leading to further boosts in quality [9,10]. PMQ is founded on the principle of parsimonious modeling [13] which facilitates high-level information extraction through model interpretation. If applied to an appropriately acsensorized process, PMQ can serve as a catalyst to innovative ideas that can move the learning curve off the plateau of no progress.
3. **Usage monitoring for quality** *(UMQ) SPC* is often thought of as only an intra-plant activity, but the *BDBM* environment makes one realize that it is also an extra-plant activity because data on the performance of the manufactured item can be and are being collected on the item as it is used. From this point of view, a warranty event can be loosely viewed as a "failure" in the binary "success-failure" context of a plant. The hypothesis is that warranty data may provide some information about the role of the manufacturing process in creating the problem. The challenge is to link the warranty issue to the process via the signal data collected during the manufacturing of the item. Usage data are being used for maintenance scheduling, failure prediction, and product health management. The challenge is to extend the analysis to link usage data to the plant.

4. Summary and discussion

PMQ is an addition to *SPC*. It was made possible by the *BDBM* environment and is critical for value creation and success of future large-scale industrial applications. Modern plants generate large volumes of data from processes that are complex and not necessarily understood from the first principle perspective. This letter pointed out three situations, viz. the *delayed measurement problem*, the *uncertain characteristics problem*, and the *infeasible measurement problem*, where *PMQ* addresses what traditional *SPC* lacked. In fact, traditional *SPC* is "*Product* Monitoring for Quality" whereas *PMQ* stands for "Process Monitoring for Quality". *PMQ* uses measurements on the process and a learning strategy to create virtual features and a rule that identifies a large percentage of the good items. Unlike *SPC* [11,12], *PMQ* does not rely on the first principle understanding of the process though it is equipped to exploit any such knowledge. Instead of labeling "good" or "bad" like *SPC* would, *PMQ* calls "suspect" those items whose features have characteristics that differ from those in the training set. It could be viewed as a weaker classifier because its declaration of "suspect" is really equivalent to a declaration of "uncertain".

The *PMQ* classifier can be used as the first classifier in a multiple testing situation to either improve performance or to reduce the burden on subsequent classifiers. Model interpretation may provide new ideas and directions for process improvement.

Future Work

PMQ may be extended to UMQ , where extra-process data are used to improve the manufacturing process. PMQ and its potential extensions give new meaning to and opportunities for continuous improvement in manufacturing.

References

[1] Yin S, Kaynak O. Big data for modern industry: challenges and trends [point of view]. Proc IEEE 2015;103(2):143–6.
[2] Sadiku MN, Wang Y, Cui S, Musa SM. Industrial internet of things. Int J Adv Sci Res Eng 2017;3.
[3] Abell JA, Chakraborty D, Escobar CA, Im KH, Wegner DM, Wincek MA. Big data driven manufacturing — process-monitoring-for-quality philosophy. Sci Eng Data Sci Enhanced Manuf 2017;139(10).
[4] Schwab K. The fourth industrial revolution: what it means, how to respond. World Economic Forum; 2016.
[5] Murphy K. Machine learning: a probabilistic perspective. MIT press; 2012.
[6] Kagermann H, Helbig J, Hellinger A, Wahlster W. Recommendations for Implementing the Strategic Initiative Industrie 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group. Forschungsunion; 2013.
[7] Yin S, Li X, Gao H, Kaynak O. Data-based techniques focused on modern industry: an overview. IEEE Trans Indus Electr 2015;62(1):657–67.
[8] Pyzdek T, Keller P. The handbook for quality management: a complete guide to operational excellence: a complete guide to operational excellence. McGraw Hill Professional; 2012.
[9] Escobar CA, Morales-Menendez R. Machine learning and pattern recognition techniques for information extraction to improve production control and design decisions. In: Perner P, editor. Advances in Data Mining, ICDM. Springer Verlag; 2017. p. 285–95.
[10] Escobar CA, Morales-Menendez R. Machine learning techniques for quality control in high conformance manufacturing environment. Adv Mech Eng 2018;10(2). https://doi.org/10.1177/1687814018755519.
[11] Baily M, Manyka J. Is manufacturing 'cool' again? McKinsey Global Institute; 2013.
[12] Jagadish H, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, Shahabi C. Big data and its technical challenges. Commun ACM 2014;57(7):86–94.
[13] Escobar CA, Morales-Menendez R. Process-Monitoring-for-Quality—A Model Selection Criterion. Manuf. Lett. 2018.

Chapter 6

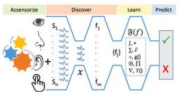
Conclusions

A summary of the contributions of this work to the field is presented, followed by the detection results of the three manufacturing case studies. Global conclusions and future research work conclude this chapter.

6.1 Contributions

PMQ is an *IBD*- and *AI*-driven quality philosophy founded on *BM*, as shown in Table 4 (e). *PMQ* is a novel approach that adds prediction to the quality control space. This new philosophy poses several theoretical challenges that must be addressed before it can be generalized across the manufacturing industry and materialize its contribution in the quality movement. This research work addressed a few of these challenges, specifically in its application for rare quality event detection in highly/ultra-unbalanced data structures.

Table 4. PMQ, a new quality philosophy for defect detection [28].

Quality philosophy	(a) SQC	(b) TQM	(c) Six sigma	(d) DFSS	(e) PMQ
Quality objective					
	Controlling	Managing	Reactive	Proactive	Predicting
Problem-solving strategy	Specification Production Inspection	Plan Do Check/study Act	Define Measure Analyze Improve Control	Identify Design product Design process Optimize Validate	Acsensorize Discover Learn Predict

The modeling paradigm of *BM* was developed and is based on several well-known *MLA* (*SVM, LR, KNN, ANN, RF, NB*) ad hoc model selection/validation tools and prediction optimization techniques. It was developed by considering the time-dependency of manufacturing systems, and it also eliminates the overoptimistic-biased created by the iterative learning process of *MLA*. *BM* facilitates information extraction, as it is founded on the principle of parsimony.

To boost parsimony and detection ability, the proposed modeling paradigm includes several novel developments that are aimed at analyzing highly/ultra-unbalanced data structures and hyper-dimensional feature spaces (*IBD*):

- A learning scheme aimed at rare quality event detection based on the L1-regularized *LR* algorithm was developed, following the *BM* learning paradigm**.** It includes the *OCTM* algorithm aimed at finding the optimal *CT* and the *HCR* filter-type redundant-feature-elimination algorithm.

- To boost parsimony, the *PMPCD MS* criterion was developed, which is aimed at inducing parsimony while maintaining detection as the main driver. It is a generic model selection

criterion that can be virtually applied to any model in which its complexity is mainly defined by the number of features it contains (SVM, NB, LR, KNN, etc.).

- To optimize the detection ability of a top-performer classifier, a decision combination approach was proposed. The proposed rules can efficiently identify the combination rule and which models should be included in the ensemble/non-ensemble decision system to optimize prediction.

- To boost parsimony, a series of three-objective optimization *MS* criteria were developed: (1) *3D−GP,* for *GP*-based models, *3D−LR,* for LR-based models and *3D−SVM*, for *SVM*-based models.

- To evaluate the robustness of prediction of a classifier, two novel separability indexes founded on the margin theory were developed.

Finally, three intractable manufacturing problems when *SPC* solution is not feasible were identified: (1) *infeasible measurement problem,* (2) *delayed measurement problem and* (3) *uncertain characteristic problem.* These problems helped to disseminate the application of *PMQ* the manufacturing industry.

6.2 Main Results

To demonstrate the ability of the *BM* modeling paradigm to detect defects and to identify the driving features, three data sets derived from real manufacturing processes were analyzed: (1) *UWBT*, (2) LSW, and (3) *SDD*. The initial number of features, features in the final model, *MLA* used in the solution, confusion matrix, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ errors**,** percentage of detection and *MPCD* scores are reported in Table 5. To report an unbiased detection ability of the solutions, all values are based on the test set (i.e., training, validation, test).

Table 5. Detection results by data set.

Data set	Features	Driving Feat.	MLA	FN	FP	TN	TP	α	β	% detected	MPCD
UWBT	54	4	LR	0	20	9973	7	0.0020	0	100%	0.9980
UWBT	54	13	*ANN, SVM	0	3	9490	7	0.0003	0	100%	0.9997
LSW	2199	1	LR	0	0	50	27	0	0	100%	1
SDD	48	3	KNN	3	27	10973	1097	0.0025	0.0027	99.72%	0.9948

*A multi-classifier solution with a predefined decision combination scheme.

According to the three case studies, classifiers developed using the *BM* learning paradigm have the potential to virtually detect all defects with a very low false positive rate ($\boldsymbol{\alpha}$ error). In addition, since *BM* is founded on the principle of parsimony, the driving features of each system were identified which are only a small fraction of the initial feature space.

6.3 Global Conclusions

Although terms like *Industry 4.0* and *Quality 4.0* [33], may seem too broad for quality leaders and organizations to be able to take any specific action, with the introduction of *PMQ*, a route for value-creation out of *AI* was paved.

Today's business environment sustains mainly those companies that are committed to a "zero-defect policy". Although *TQC* methods are applied to make a process extremely reliable, they have plateaued off, and their limitation in solving complex problems, managing *IBD* and virtually creating a *defect-free process* is evident.

The quality movement includes controlling, managing, reacting and proactive quality philosophies, in which the problem-solving strategy has been adapted based on the goals of each philosophy. *PMQ* is an addition in this movement. It is a predictive philosophy that has also adapted the problem-solving strategy to guide engineers in efficiently solving the pattern classification problem.

The learning and predicting steps of this strategy are solved by *BM*, which can mine massive amounts of data to automatically learn models that represent manufacturing systems, and, therefore, move the process learning curve off the plateau of no progress.

Pattern classification is an extremely broad topic with a plethora of options. Poor domain knowledge can lead to a low-quality solution. The *BM* modeling paradigm was developed based on theory and empirical evidence to guide manufacturing engineers to efficiently solve the problem or to determine, under uncertainty, if the learning targets for the project can or cannot be met. Based on empirical results, there is compelling evidence to conclude that this modeling paradigm is a step forward in the development of a "zero-defect policy".

Today's innovative manufacturing environment and competitive business environment sometimes force the launching of a product even though the three requirements for an *SPC* program cannot be satisfied. In these cases, *PMQ* can supplement an empirical predictive framework for process control without any other *QC* tool – as demonstrated in the Chevrolet Volt case study. Thus, *PMQ* applications and its potential extensions provide new meanings to and opportunities in the path of continuous improvement.

PMQ was designed to model uncertainty, therefore it should be applied to support *SPC* or to address situations in which *SPC* is not feasible, but *PMQ* was not designed to replace a first principle-based solution (when it is available). Moreover, *PMQ* was designed to be applied to discrete manufacturing systems, where its application to continuous processes is out of the scope. Finally, the extension of *PMQ* to detect more than one defect is straight forward (each class against all [34]), if each defect is well characterized in the training data.

According to the quality legend Dr. Joseph Juran (1904-2008), the 20th century was a century defined by productivity. He promised that the quality movement had just begun, and the 21st century would be the century of quality [35,36]. Therefore, it is clear that – while necessary – *TQC* and improvement techniques will not be enough; *PMQ* is a step forward in satisfying the expectations of customers in this century.

6.4 Future Work

This research work can be extended in the three following topics:

1. *Usage Monitoring for Quality (UMQ)*, where extra-process data are used to improve the manufacturing process.

2. Adaptive learning, frequently the training set do not include all the possible defects that can be generated by the process, consequently novel defects are not detected. To create the *defect-free process* described in Figure 5, an automatic relearning system is required, Figure 8.

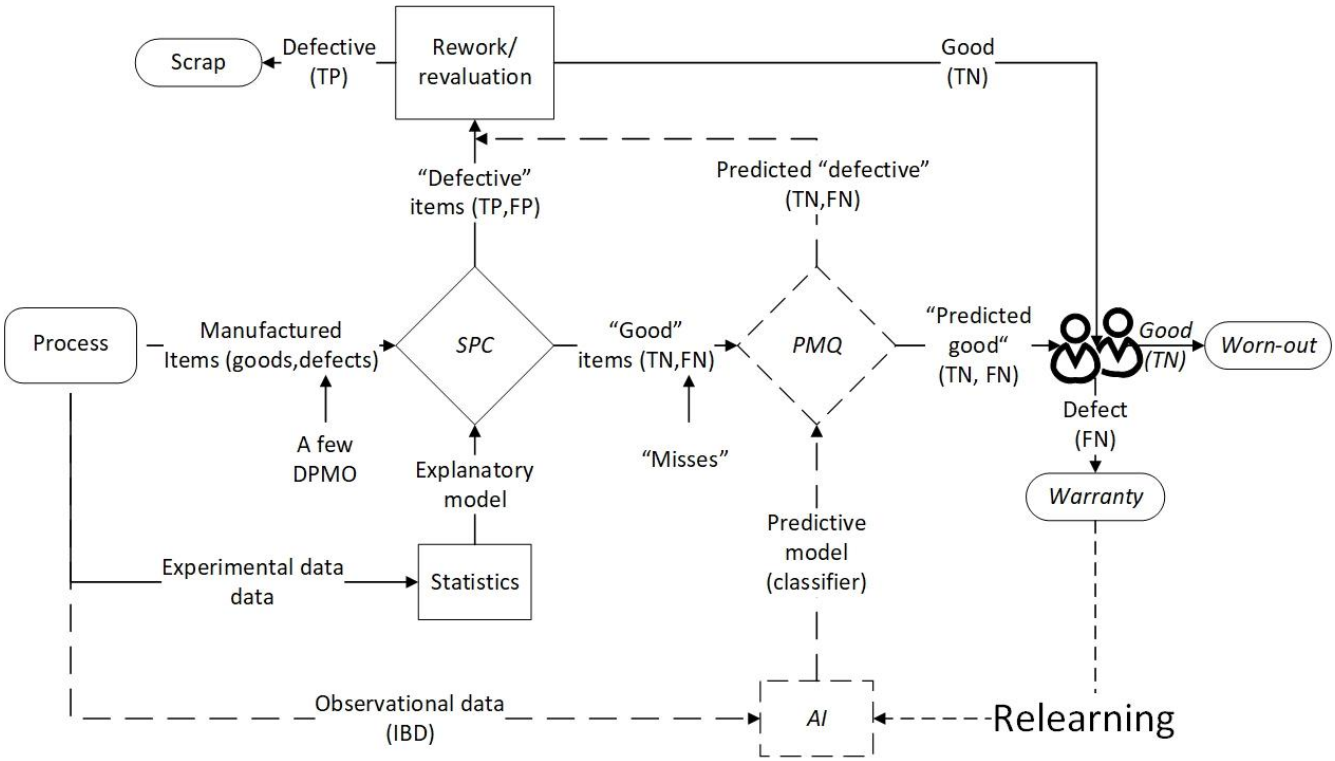


Figure 8. Relearning scheme based on undetected defective items.

3. Deep learning comprises a set of algorithms that automatically discover expressive features to develop a detection task. Defects can be detected through image recognition. In this context, this learning technique is also a future work to be considered to replace human-based inspections, especially considering that in 2015, deep learning surpassed human accuracy [37].

Appendix A

The main dataset used in this research work was derived from the ultrasonic welding of battery tabs for the Chevrolet Volt. It was generated in 2014 and stored in GM's repository.

Ultrasonic welding is an ideal process for bonding conductive materials such as copper, aluminum, brass, gold, and silver and for joining dissimilar materials. Recently, it has been adopted for battery tab joining in the manufacturing of vehicle battery packs. It is a joining

process where entities that are in physical contact are joined by rapid relative motion between adjacent surfaces.

An ultrasonic welder is a device that generates vibrations in a transducer assembly and applies them to the work pieces via a knurled sonotrode/horn. The work pieces are supported by a stationary knurled anvil. The sonotrode presses against the anvil, with the work pieces between, and vibrates (tangentially to the surface of the work pieces). During the welding process, the sonotrode presses against the anvil and compresses the work pieces. Too little pressure results in no weld; too much pressure destroys the weld. Hereafter, the sonotrode and the anvil will be collectively referred to as the tool. Figure 9 shows these components in a schematic of an ultrasonic welding application for battery tabs.

The battery cells used in the Volt battery come as pouches with tabs for electrical connection as shown in Figure 10(a). A cell group consists of three cells that are stacked together (electrically in parallel) and physically joined at the tabs to a busbar on an interconnect board as shown in Figure 10(b). All the cells that were joined to the same interconnect board constituted a module, and the busbar provided the series current carrier for the cell group in a module. The final battery pack (Figure 10(c)) was a series connection of such modules. Because the connections were in series, every single weld had to perform for the battery to function. Each weld had to meet two criteria: one was the mechanical strength of the weld and the other was its electrical conductance. The definitive test for strength is to pull the welds apart and note the force.

A primary source of data comes from the welder controller. The operational mode of the controller is determined by user specified parameters. The controller outputs a power signal to indicate the faithfulness of the performance to the specified mode.

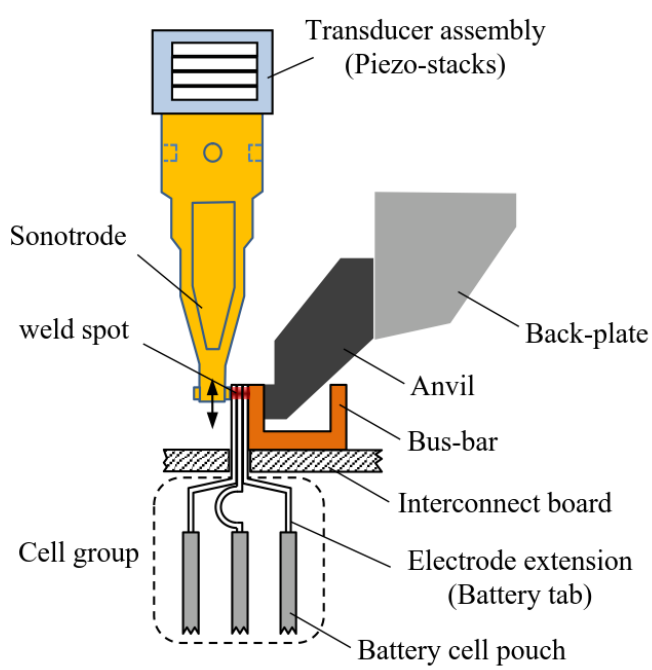
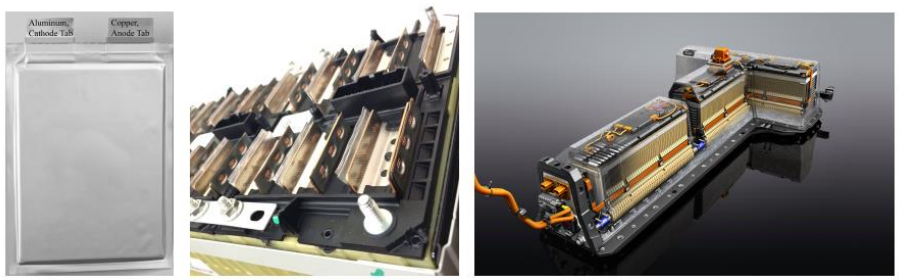


Figure 9. Ultrasonic welding schematic for battery tabs.

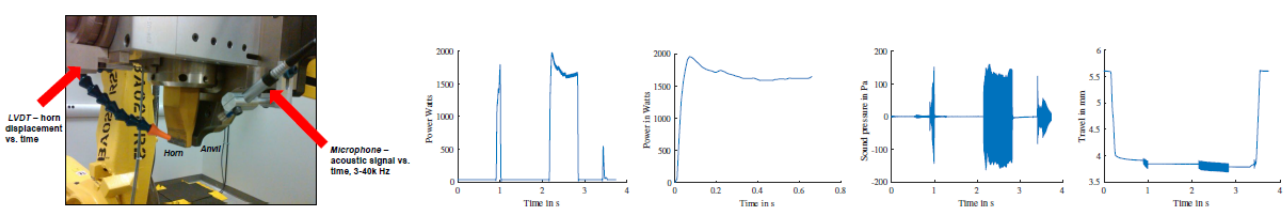


(a) Cell. (b) Module. (c) Graphical rendering of the battery.

Figure 10. The first generation Chevrolet Volt battery.

Following basic engineering knowledge and experience, the following sensors and example signals were deployed and collected respectively:

- The internal high-resolution power curve over the entire processing period was captured (Figure 11(b)) in addition to the readily available low-resolution power curve of the welding (Figure 11(c)).
- A temperature gauge was proposed (but not initially implemented).
- An acoustic sensor (a microphone) (Figure 11(a)) was added to hear beyond the human range (Figure 11(d))
- A linear variable differential transformer (LVDT) (Figure 11(a)) was added to measure in real time the compression (Figure 11(e)) of the battery tabs.



(a) Some additional sensors to ultrasonic welder. (b) Signal from power sensor (high resolution). (c) Signal from power sensor (low resolution). (d) Signal from microphone. (e) Signal from LVDT.

Figure 11. Acsensorization of ultrasonic welder and an example of observed signals.

The intent of acsensorization is to ensure that all the physical aspects of the process are captured so that information from the data could provide an insight into the process. In addition to the choice of sensors, the characteristics of the sensors, such as their range of sensitivity, and their placement and installation require engineering expertise.

References

[1] Colvin, Geoff. “How CEO Mary Barra is using the ignition-switch scandal to change GM's culture,” Fortune September 2015, [Online]. Available: **http://fortune.com/2015/09/18/mary-barra-gm-culture/** [Accessed 2018].

[2] Crosby, Philip B. “The Absolutes of Quality Management.” Industrial Management (1985).

[3] Baily, M. N., and J. Manyika. “Is Manufacturing’Cool’Again.” McKinsey Global Institute (2013).

[4] Wang, Ke-Sheng. "Towards Zero-Defect Manufacturing (ZDM)—A Data Mining Approach." Advances in Manufacturing 1.1 (2013): 62-74.

[5] Wuest, Thorsten, Christopher Irgens, and Klaus-Dieter Thoben. "An Approach to Monitoring Quality in Manufacturing using Supervised Machine Learning on Product State Data." Journal of Intelligent Manufacturing 25.5 (2014): 1167-1180.

[6] Granstedt Möller, Erik. "The Use of Machine Learning in Industrial Quality Control." (2017).

[7] Li, Yundong, Weigang Zhao, and Jiahao Pan. "Deformable Patterned Fabric Defect Detection with Fisher Criterion-Based Deep Learning." IEEE Transactions on Automation Science and Engineering 14.2 (2017): 1256-1264.

[8] Sun, Te-Hsiu, Fang-Cheng Tien, Fang-Chih Tien, and Ren-Jieh Kuo. "Automated Thermal Fuse Inspection using Machine Vision and Artificial Neural Networks." Journal of Intelligent Manufacturing 27.3 (2016): 639-651.

[9] Pandey, Rashmi, Sapan Naik, and Roma Marfatia. “Image Processing and Machine Learning for Automated Fruit Grading System: A Technical Review.” International Journal of Computer Applications (2013).

[10] Sa, Inkyu, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. “Deep Fruits: A Fruit Detection System Using Deep Neural Networks.” Sensors 16, no. 8 (2016): 1222.

[11] Ghorai, Santanu, Anirban Mukherjee, M. Gangadaran, and Pranab K. Dutta. "Automatic Defect Detection on Hot-Rolled Flat Steel Products." IEEE Transactions on Instrumentation and Measurement 62.3 (2013): 612-621.

[12] Du, Shichang, Changping Liu, and Lifeng Xi. "A Selective Multiclass Support Vector Machine Ensemble Classifier for Engineering Surface Classification using High Definition Metrology." Journal of Manufacturing Science and Engineering 137.1 (2015): 011003.

[13] Elangovan, M., N. R. Sakthivel, S. Saravanamurugan, Binoy B. Nair, and V. Sugumaran. "Machine Learning Approach to the Prediction of Surface Roughness using Statistical Features of Vibration Signal Acquired in Turning." Procedia Computer Science 50 (2015): 282-288.

[14] Bosch, 2016. Kaggle Bosch Production Line Performance. [Online] Available at: **https://www.kaggle.com/c/bosch-production-line-performance** [Accessed 2017].

[15] Wuest, Thorsten, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. "Machine Learning in Manufacturing: Advantages, Challenges, and Applications." Production & Manufacturing Research 4.1 (2016): 23-45.

[16] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. “Why Should I Trust You? Explaining the Predictions of Any Classier," in in Proc of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2016): 1135-1144.

[17] Shmueli, Galit. "To Explain or to Predict?." Statistical Science 25, no. 3 (2010): 289-310.

[18] Chandrashekar, Girish, and Ferat Sahin. "A Survey on Feature Selection Methods." Computers & Electrical Engineering 40.1 (2014): 16-28.

[19] Tang, Jiliang, Salem Alelyani, and Huan Liu. "Feature Selection for Classification: A Review." Data classification: Algorithms and applications (2014): 37.

[20] Juran, Joseph M. "Early SQC: A Historical Supplement." Quality Progress 30, no. 9 (1997): 73-82.

[21] Montgomery, Douglas C. “Introduction to Statistical Quality Control,” John Wiley & Sons (2012).

[22] Montgomery, Douglas C. and John Bert Keats, eds. “Statistical Process Control in Manufacturing”. Marcel Dekker, (1991).

[23] Vazquez-Lopez, J. A., and Ismael Lopez-Juarez. "SPC Without Control Limits and Normality Assumption: A New Method." In Iberoamerican Congress on Pattern Recognition, Springer, Berlin, Heidelberg (2009): 611-618.

[24] Wasserman, Larry. “All of Statistics: A Concise Course in Statistical Inference,” Springer Science & Business Media, 2013.

[25] Devore, Jay L. “Probability and Statistics for Engineering and the Sciences.” Cengage learning (2011).

[26] P. Ghosh, “A Comparative Roundup: Articial Intelligence vs. Machine Learning vs. Deep Learning." June 2016. [Online]. Available: **www.dataversity.net/ai-vs-machine-learning-vs-deep-learning**

[27] Montgomery, Douglas C. “Exploring Observational Data,” Quality and Reliability Engineering International, 33(8), 2017: 1639-1640.

[28] Abell, Jeffrey A., Debejyo Chakraborty, Carlos A. Escobar, Kee H. Im, Diana M. Wegner, and Michael A. Wincek. “Big Data Driven Manufacturing — Process-Monitoring-for-Quality Philosophy.” ASME J of Manufacturing Science and Eng on Data Science-Enhanced Manufacturing, vol. 139, no. 10, (2017).

[29] Robnik- Sikonja M and Kononenko I. “Theoretical and Empirical Analysis of ReliefF and RReliefF.” Machine Learning (2003): 53: 23–69.

[30] Escobar, Carlos A., Diana M. Wegner, Abhinav Gaur, and Ruben Morales-Menendez. "Process-Monitoring-for-Quality—A Model Selection Criterion for Genetic Programming." In International Conference on Evolutionary Multi-Criterion Optimization. Springer, Cham (2019): 151-164.

[31] Escobar, Carlos A., and Ruben Morales-Menendez. "Process-Monitoring-for-Quality—A Model Selection Criterion for l1-Regularized Logistic Regression." To appear in Procedia Manufacturing.

[32] Escobar, Carlos A., and Ruben Morales-Menendez. "Process-Monitoring-for-Quality—A Model Selection Criterion for Support Vector Machine." To appear in Procedia Manufacturing.

[33] Jacob Dan. “Quality 4.0 Fresh Thinking for Quality in the Digital Era.” Quality Digest, July (2017).

[34] Allwein, Erin L., Robert E. Schapire, and Yoram Singer. "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers." Journal of Machine Learning Research 1.Dec (2000): 113-141.

[35] Paton Scott, M. "A Century of Quality: An Interview with Quality Legend Joseph M." Juran. Quality Digest, February (1999).

[36] AS of Quality. Emergence—2011 Future of Quality Study. Milwaukee, WI: ASQ: The Global Voice of Quality (2011).

[37] Stanford Vision Lab, 2017. ImageNet Large Scale Visual Recognition Challenge. [Online] Available at: **http://image-net.org/challenges/LSVRC/** [Accessed 2017].

Curriculum Vitae

Carlos Alberto Escobar was born in Ciudad Juarez, Chih. Mexico. He earned an Industrial Engineering degree with concentration in automated manufacturing from the Instituto Tecnológico de Ciudad Juarez in 2003, a master's degree in engineering with Concentration in Quality and Productivity Systems from the Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Juarez in 2005, and a master of science in Industrial Engineering from New Mexico State University, Campus Las Cruces, New Mexico in 2015. He was certified six-sigma black belt from Arizona State University in 2008 and design for six-sigma black belt from University of Michigan in 2012. In 2013 he was inducted into the of Alpha Pi Mu Industrial Engineering honor society and into Tau Beta Pi engineering honor society in 2014. In 2017, Carlos was ranked into the top 3% in TEXATA, the Big Data Analytics World Championships. He was accepted into the PhD in Engineering Sciences program in Campus Monterrey in May 2015.

PUBLICATIONS

- Abell, J. A., Chakraborty, D., Escobar, C. A., Im, K. H., Wegner, D. M., & Wincek, M. A. (2017). Big Data-Driven Manufacturing—Process-Monitoring-for-Quality Philosophy. Journal of Manufacturing Science and Engineering, 139(10), 101009.
- Escobar, C. A., & Morales-Menendez, R. (2017, July). Machine Learning and Pattern Recognition Techniques for Information Extraction to Improve Production Control and Design Decisions. In Industrial Conference on Data Mining (pp. 286-300). Springer, Cham.
- Escobar, C. A., & Morales-Menendez, R. (2018). Machine Learning Techniques for Quality Control in High Conformance Manufacturing Environment. Advances in Mechanical Engineering, 10(2), 1687814018755519.
- Escobar, C. A., & Morales-Menendez, R. (2018). Process-Monitoring-for-Quality—A Model Selection Criterion. Manufacturing Letters, 15, 55-58.
- Escobar, C. A., Wincek, M. A., Chakraborty, D., & Morales-Menendez, R. (2018). Process-Monitoring-for-Quality—Applications. Manufacturing letters, 16, 14-17.
- Escobar, C. A., Abell, J. A., Hernández-de-Menéndez, M., & Morales-Menendez, R. (2018). Process-Monitoring-for-Quality—Big Models. Procedia Manufacturing, 26, 1167-1179.
- Escobar, C. A., Wegner, D. M., Gaur, A., & Morales-Menendez, R. (2019). Process-Monitoring-for-Quality—A Model Selection Criterion for Genetic Programming. Springer Nature Switzerland AG 2019, K. Deb et al. (Eds.): EMO 2019, LNCS 11411, pp. 1–14, 2019. https://doi.org/10.1007/978-3-030-12598-1_13

- Escobar, C. A., & Morales-Menendez, R. Process-Monitoring-for-Quality—A Model Selection Criterion for the L1-Regularized Logistic Regression. To appear in the Journal of Procedia Manufacturing.
- Escobar, C. A., & Morales-Menendez, R. Process-Monitoring-for-Quality—A Model Selection Criterion for the Support Vector Machine. To appear in the Journal of Procedia Manufacturing.

PRESENTATIONS

- Escobar, C. A., Wegner, D. M., Gaur, A., & Morales-Menendez, R. "Process-Monitoring-for-Quality — A Model Selection Criterion for Genetic Programming," (2019, March). 10th International Conference, EMO 2019 East Lansing, MI.
- Escobar, C. A., & Morales-Menendez, R. (2018, July). "Parsimonious Modeling for Binary Classification of Quality in a High Conformance Manufacturing Environment." 18th Industrial Conference on Data Mining (ICDM), New York, NY.
- Escobar, C. A., Abell, J. A., Hernández-de-Menéndez, M., & Morales-Menendez, R. (2018, June). Process-Monitoring-for-Quality—Big Models. 46th North American Manufacturing Research Conference (NAMRC 47), College Station TX.
- Escobar, C. A., & Morales-Menendez, R. (2017, July). Machine Learning and Pattern Recognition Techniques for Information Extraction to Improve Production Control and Design Decisions. 17th Industrial Conference on Data Mining (ICDM) , New York, NY.
- Escobar, C. A., (2015). "A Lean Six Sigma Analysis Supported by Discrete Event Simulation for Pecan Production Improvement." IIE Annual Conference and Expo - Applied Solutions Sessions, Nashville, TN, June 1-3.
- Escobar, C. A., & Sohn, H. (2015). "Simulation Optimization Model for Emergency Medical Service," Shared Knowledge Conference at the University of New Mexico, Albuquerque, NM, April 23-24.
- Escobar, C. A., & Sohn, H. (2015). "Simulation Optimization Model for Emergency Medical Service," Seminar, General Motors, Global Research and Development, Warren, MI, February 5.
- Escobar, C. A., & Sohn, H. (2014). "Simulation Optimization Model for Emergency Medical Service," SHPE National Conference, Detroit, MI, November 5-8.
- Escobar, C. A., & Sohn, H. (2014). "Decision Support System for Emergency Medical Service," HENAAC National Conference, New Orleans, LA, October 2-4.
- Escobar, C. A., & Sohn, H. (2013). "A Hybrid Analytical Approach for Optimal Ambulance Location in Emergency Medical Service," SACNAS National Conference, San Antonio, TX, October 3-6.

This document was typed in using Microsoft Word by Carlos Alberto Escobar Díaz