Transactions on Machine Learning
and Data Mining
Vol. 11, No 1 (2018) 27-40
© ISSN: 1865-6781 (Journal),
ISBN: 978-3-942952-58-3

ibai Publishing
www.ibai-publishing.org

Parsimonious Modeling for Binary Classification of Quality in a High Conformance Manufacturing Environment

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

[1] General Motors, Research and Development, Warren MI 48092, USA,
carlos.1.escobar@gm.com
[2] Tecnológico de Monterrey, Monterrey, NL. 64849, México,
rmm@itesm.mx

Abstract The world of *big data* is changing dramatically; in the domain of data mining, machine learning and pattern recognition, the feature access has grown from tens to hundreds or even thousands. This trend presents enormous challenges, specially for classification problems. In manufacturing, classification of quality is one of the most important applications; however, feature explosion, combined with high conformance production rates are two of the most important challenges for *big data* initiatives. Empirical evidence shows that discarding irrelevant or redundant features improves prediction, helps in understanding the system, reduces running time requirements, and reduces the effect of dimensionality. In this paper, the *Hybrid Correlation- and Ranking-based (HCR)* and *ReliefF* filter feature elimination algorithms are presented as a wrapper method, which uses the *Naive Bayes* as the learning algorithm. To boost parsimony, the algorithms are combined with the *Penalized Maximum Probability of Correct Decision* – a model selection criterion – to develop a *Hybrid Feature Selection and Pattern Recognition* framework aimed at rare quality event detection. A flexible approach that can be widely applied to various machine learning algorithms.

Keywords Quality control · Manufacturing systems · Feature elimination algorithm · Model selection criterion · Unbalanced binary data · Defect detection

1 Introduction

We are living in a world that is highly influenced by the rise of *big data*. The information explosion that companies are facing with ever-increasing amounts of data highlights the importance of information extraction techniques. When analyzing large volumes of data, data mining, machine learning and pattern recognition techniques are used for data-driven knowledge discovery (e.g., model discovery), pattern recognition (e.g., classification) and/or to display hidden patterns in the data. In these *big data*-driven techniques, a feature (e.g., variable) is an individual measurable property of a phenomenon being observed [1]; the

prediction ability of a learning algorithm is mainly determined by the inherent class information available in the features included in the analysis [2]. And generalization refers to the prediction ability of a learning algorithm-based model on unseen data.

Theoretical analysis and empirical evidence show that irrelevant and redundant features 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 computational time; and (3) it is more difficult to extract high-level knowledge from the analysis [3–5].

Dependence can be described as any statistical relationship between two random variables. Correlation refers to a broad class of statistical relationships involving dependence. The most common measure of linear dependence is the Pearson product-moment correlation coefficient[6].

In this context, a feature may be considered 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 – redundant [5, 7]. On the other hand, a feature may be considered irrelevant if the information that it contains is independent from the class label. In the *Feature Selection (FS)* domain, the selection of relevant features and elimination of irrelevant and redundant ones is one of the main challenges [8].

1.1 *Big Data* in Manufacturing

Manufacturing companies are intense users of *big data*, this industry generates and stores more data than any other [9]. Learning algorithms e.g, support vector machine, logistic regression, decision trees to name a few, are applied for quality monitoring and process control [10]. Classification of quality is one of the most important applications, where relevant quality characteristics of the process or product are observed and related to an ordinal or binary output aimed at detecting defects [11]. *Big data* initiatives have the potential to solve a whole range of hitherto intractable manufacturing problems [12].

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. This may cause serious problems to many learning algorithms with respect to the scalability and learning performance [5]. Because most mature manufacturing organizations generate only a few defects per million of opportunities, another common challenge when analyzing manufacturing-derived data sets is their highly unbalanced data structure. The feature explosion combined with high conformance production rates are two of the most important challenges of *big data* initiatives in manufacturing.

Table 1: Acronyms Table

Acronym	Definition
FN	False Negatives
FP	False Positives
FS	Feature Selection
HCR	Hybrid Correlation- and Ranking-based
HFSPR	Hybrid Feature Selection and Pattern Recognition
MPCD	Maximum Probability of Correct Decision
MS	Model Selection
NB	Naive Bayes
PMPCD	Penalized Maximum Probability of Correct Decision
SUFL	Sorted and Uncorrelated Feature List
TN	True Negatives
TP	True Positives

In contrast with other industries, where prediction is the main goal, in manufacturing, model interpretation – from a physics perspective – is very important. Since the extracted information of the cases yielding high quality can be used by engineers to plan and to design randomized experiments to find optimal levels of process/product parameters. This problem representation highlights the importance of finding a few good empirical-data-derived features to approximate the patterns of manufacturing systems (parsimony [13]).

Parsimonious modeling aimed at detecting rare quality events is the main driver of this research. Parsimony is induced through *FS* and *Model Selection (MS)*.

The *Hybrid Correlation- and Ranking-based* [14], is a filter *FS* algorithm aimed at eliminating redundant features, where the *Pearson*'s correlation coefficient is used as a measure of redundancy. The basic idea of the algorithm is to keep the *best* feature – highest ranked – from a set of two or more highly correlated variables and eliminate the rest. It uses the *ReliefF* algorithm to rank the features according to their discriminative capacity.

In this paper, *HCR* and *ReliefF* algorithms are presented as a wrapper method. Due to the strong assumption of independence of variables, the *Naive Bayes (NB)* is used as the learning algorithm. To boost parsimony, the algorithms are combined with the *Penalized Maximum Probability of Correct Decision (PMPCD)* – a model selection criterion [15] – to develop a *Hybrid Feature Selection and Pattern Recognition (HFSPR)* framework; aimed at analyzing highly unbalanced data structures.

This paper is organized as follows: it starts with a review of the theoretical background in section 2. Section 3 describes the *HFSPR* framework, followed by a binary classification empirical study in section 4. Finally, conclusions and opportunities for future research are included in section 5.

Table 2: Variables Table

Variable	Description
α	type I error
β	type II error
δ	high-correlation threshold
F	list of features in descending order
FC	feature correlation matrix
k	number of nearest neighbors
K	number of features in the candidate model
m	number of sampled instances
n	number of features
r_{xy}	Pearson correlation coefficient
τ	feature relevance threshold
$\bar{x}$	mean of variable x
x_i	data point i of variable x
x, y	correlated variables
$\bar{y}$	mean of variable y
y_i	data point i of variable y

2 Theoretical Background

2.1 Feature Selection Methods

Feature selection can be defined as the process of choosing a subset of good features, and eliminating irrelevant and redundant ones from the original feature set. From a given data set, evaluating all possible combinations (2^n) becomes an NP-hard problem as the number of features grows [16]. The *FS* methods broadly fall into two classes: filters and wrappers [17].

Filter 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 and thus selected if its associated weight is greater than the user-specified threshold [5]. 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 [7].

Wrappers, use the learning algorithm as a black-box to evaluate the relative performance of a feature subset [18, 19]. In this procedure, a set of candidate features are input to the learning algorithm, and the prediction performance is used as the objective function to evaluate the feature subset. Although the wrapper methods can become computationally intensive, they perform better than filters due to the bias induction by the algorithm [17]. However, the classifier may learn the training data too well (i.e., become over-fitted), but exhibit poor generalization ability. To avoid this situation, a holdout set can be used to track the classifier's accuracy on unseen data.

Recently, hybrid approaches have been proposed by [3] to take advantage of the particular characteristics of each method. These approaches mainly focus on

combining filter algorithms with either wrappers or regularization to solve the scalability problem and to achieve the best possible learning performance with a particular algorithm. The basic idea is to break down the *FS* problem into several stages, namely feature ranking, correlation-based feature elimination, and prediction optimization.

2.2 *Relief* and *ReliefF*

The basic idea of *Relief* is to estimate the quality of features according to how well their values distinguish between instances that are near to each other [20]. Its advantages are that it is not dependent on heuristics, it runs in low-order polynomial time, and it can be applied to nominal or numerical features. However, *Relief* does not eliminate redundant features, cannot deal with incomplete data and is limited to two-class problems.

ReliefF is an extension of the *Relief* algorithm, it was improved by Kononenko to generalize to multiclass problems. In addition, the improved algorithm (*ReliefF*) is more robust to incomplete and noisy data sets [21]. *ReliefF* searches for a k of its nearest neighbors from the same class, called nearest *hits*, and also a 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* [22] to select the most important features [20], developing a significance threshold τ. Features with an estimated weight below τ are considered irrelevant and therefore eliminated. The proposed limits for τ are $0 < \tau \leq 1/\sqrt{\alpha m}$ [22]; where α is the probability of accepting an irrelevant feature as relevant.

2.3 Correlation-Based Redundancy Measure

The *Pearson* product-moment correlation coefficient (or *Pearson* correlation coefficient) is used as a measure of redundancy between two random variables [6]. The *Pearson* correlation coefficient (r_{xy}), is a measure of strength of linear relationship between two variables (x, y), and it can take a range of values from +1 to -1, eq. (1). A value of 0 indicates that there is no linear relationship between the two variables, while an absolute value of 1 (or close to 1) indicates strong linear relationship, and therefore considered highly redundant.

$$r_{xy} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \tag{1}$$

2.4 Naive Bayes

Naive Bayes is a probabilistic 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 [23]. *NB* is fast calculating the needed probabilities as it only performs one scan to the data [24]. *NB* has a strong independence of variables assumption [25]. Another assumption of *NB* is that numerical values have always a normal distribution. *NB* is easy to develop [23] and its classification process is easy to understand as well. It also offers computational time savings for training as it only needs a small amount of data; it is also fast classifying and requires minor storage space in both previous tasks . Besides, it is not affected by missing values as it omits them. In this sense, *NB* is suitable for working with high amount of data [23]. *NB* cannot remove irrelevant features and its performance is highly dependent on the feature selection procedure used. Finally, this algorithm is very affected by irrelevant features [24].

2.5 Maximum Probability of Correct Decision

In predictive analytics, a confusion matrix [26] is a table with two rows and two columns that reports the number of *False 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* [6]. They are defined as:

$$\alpha = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \quad \beta = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}. \tag{2}$$

The *MPCD* is a probabilistic-based measure of classification performance. It is more sensitive to the recognition rate by class than just the proportion of correct guesses. The α, and beta β errors are combined to estimate *MPCD*:

$$\mathrm{MPCD} = (1 - \alpha)(1 - \beta) \tag{3}$$

where higher score ($0 \leq MPCD \leq 1$) indicates better classification performance.

2.6 Penalized Maximum Probability of Correct Decision

It is a *MS* criterion for binary classifiers in highly unbalanced data structures (i.e., 0.1-3% of defects) [15]. This criterion solves the posed tradeoff between model complexity (e.g., number of features) and prediction ability.

$$PMPCD = (1 - \alpha)(1 - \beta) - ln(K)/34.55 \tag{4}$$

where K is the number of features, and the model with the highest estimated value on the validation set [27–29] is the preferred one.

The term $(1 - \alpha)(1 - \beta)$ rewards the prediction capacity, while the penalty function $ln(K)/34.55$ 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, with no penalty imposed for a single-feature model.

2.7 Hybrid Correlation and Ranking-based Algorithm

The *HCR* algorithm [14] eliminates redundant features based on *Pearson*'s correlation coefficients and the *ReliefF* algorithm ranking. The basic idea is to keep the *best* feature – highest rank – from a set of two or more highly correlated variables and eliminate the rest in that group.

3 Hybrid Feature Selection and Pattern Recognition

Parsimonious modeling is induced through feature selection and model selection, Fig. 1. Since most manufacturing systems are time-dependent, cross-validation methods are not encouraged. Instead, time-ordered hold-out method seems to be more appropriate. The data set should be splitted into training, validation and testing sets (e.g., 50%, 25%, 25% respectively) [28]. And the search space defined by many candidate pairwise combinations – based on different values of k for *ReliefF* and δ for *HCR*. The values of k can be determined by generating a logarithmically spaced vector [30] – p logarithmically spaced points between decades 10^a and 10^b, where $X = sum(bad)$ in the training set, $a = 0$ and $b = log10(X)$.

1. Feature selection
   The primary purpose of this stage, is to find a small subset of features with high prediction capacity. Since the optimal combination – with respect to prediction – of k and δ is not known in advance, a hyperparameter [31] optimization is performed through a grid search [32, 33]. Using the training set, irrelevant and redundant features are eliminated by applying *ReliefF* and *HCR* algorithms. First, features are ranked based on *ReliefF* and irrelevant features are eliminated based on τ – feature relevance (significance) threshold. From the selected features, high correlations are eliminated based on δ. These two steps are performed in a filter-type approach, where the learning algorithm is not considered. The outcome of this step, is a subset of relevant features with no high correlations.
   A *candidate* model is developed with the subset of features at each pairwise combination, and the predictive fitness of each model is evaluated to find the *incumbent* (best so far) model – highest validation *MPCD*. The features in the *incumbent* model are selected and their associated *ReliefF* ranking recorded.
2. Model selection
   Although a good feature subset has been obtained in the previous step, their individual relevance in the model is not known in advance. To evaluate their prediction-contribution, a set of n *candidate* models are developed – where n is the number of selected features – using the top 1 feature in the first *candidate* model, and the top 2 features in the second one, and so on. Finally, the *PMPCD* of each *candidate* model is estimated and used as a *MS* criterion to induce parsimony – solve the tradeoff between model complexity and prediction ability. The *final* model is the one with the highest *PMPCD* score.

3. Generalization evaluation
   To obtain an unbiased estimation (or closest to) of the generalization ability of the *final* model, the prediction on testing set (unseen data) should be reported in a confusion matrix [26] .

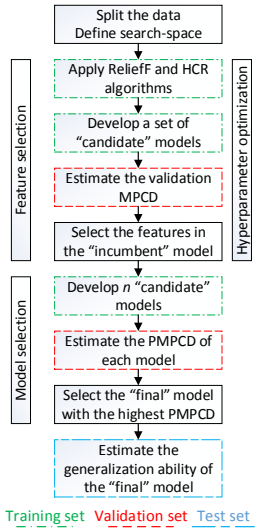


Fig. 1: *HFSPR* framework.

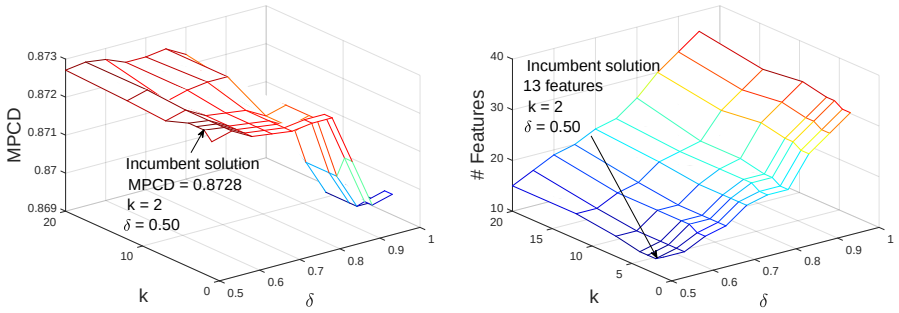
4 Case Study – Ultrasonic Metal Welding

To validate the practical and theoretical advantages of the *HFSPR* approach a manufacturing-derived data set is analyzed. Due to the strong independence of variables assumption, the *NB* learning algorithm is used in this analysis, however, the proposal can be virtually applied to any binary classifier. The data used for this analysis is derived from the *Ultrasonic Metal Welding* of battery tabs for the *Chevrolet Volt* [11], an extended range electric vehicle. A very stable process, that only generates a few defective welds per million of opportunities.

4.1 Hybrid Feature Selection and Pattern Recognition

The collected data set contains a binary outcome (*good/bad*) with 54 features. The data set is highly unbalanced since it contains only 35 *bad* batteries out of 30,731 examples. To run the analysis, the data set is partitioned following a time-ordered hold-out validation scheme: training set (18,495, including 20 *bad*), validation set (12,236 - 8 *bad*), testing set (9,500 - 7 *bad*).

1. Feature selection
   The search space contains 35 pairwise combinations; for *ReliefF*, 7 logarithmically spaced points are defined – $k = \{1, 2, 3, 4, 7, 12, 20\}$ – and for δ, 10 even spaced points – $\delta = \{0.50, 0.55, ..., 0.95\}$. At each iteration, feature relevance is determined by comparing their weights with $\tau = 0.0329$ – calculated with an α of 0.05, and m of 18,495. Prediction results and number of features of each *candidate* model are shown in Fig. 2.



(a) Validation *MPCD*. (b) Number of features.

Fig. 2: Candidate model information (denoted by line intersections).

   According to the grid search results, the *incumbent* model has an estimated validation $MPCD = 0.8728$, Fig. 2(a), and 13 features, Fig. 2(b). This model was developed with the following relevant hyperparameters – $k = 2, \tau = 0.0329, \delta = 0.50$. All *candidate* models failed to detect one of the defective items, therefore, the $\beta = 0.125$ in all models. And they are basically competing over the α error. As displayed by the plots, as the number of low quality features included in the model increases, the α error increases too. The proposed hyperparameter optimization allowed to find a good subset of features.
2. Model selection
   To induce parsimony, 13 *candidate* models are create, and *PMPCD* is used as a model selection criterion to find the *final* model. The basic idea is to evaluate the individual prediction-contribution of each of the 13 selected features, Fig. 3 shows the selected features and their associated ranking. *Candidate* model 1 contains top 1 feature (25), *candidate* model 2 contains the top 2 features (25,5) and so on.
   According to the model selection criterion, *Candidate* model 2 should be selected, with an estimated $PMPCD = 0.8501$, Fig. 4. This analysis, discloses that only two features are needed to approximate the pattern in the manufacturing system, since the prediction improvement is not significant if more features are added to the *final* model.

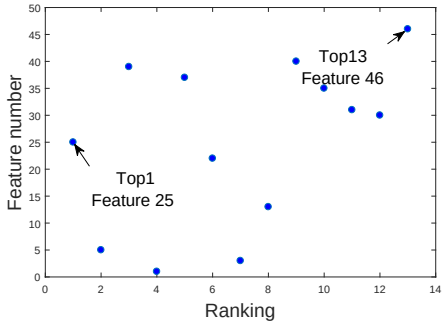


Fig. 3: Features in the *incumbent* model.

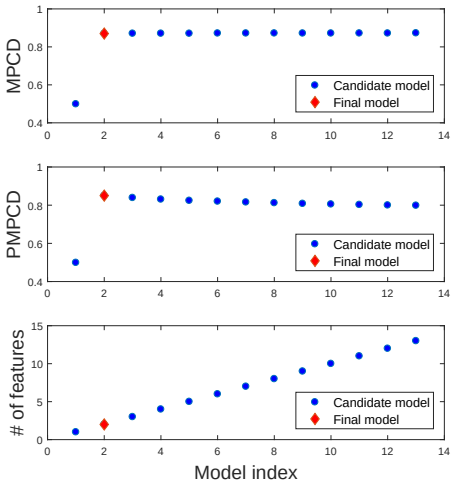


Fig. 4: *Candidate* models using the top 13 features.

3. Generalization evaluation
   The testing set is used to estimate the generalization ability of the *final* model, recognition rates are summarized in the confusion matrix, Table 3. This model includes only two features (25,5), and it correctly detected the seven defective items with only five FPs – $MPCD = 0.9995$. It is clear that the system can be explained by only these two features.

Table 3: Confusion Matrix

	Declare good	Declare bad
good	9488	5
bad	0	7

4.2 Solution Evaluation and Discussion

Although the feature combination is subject to combinatorial explosion, 1.80144E+16 number of combinations in this case study, the *HFSPR* approach only required 48 models to find a solution. To evaluate its relative quality, an exhaustive search (due to computational feasibility) is performed with all the possible combinations – up to two features – and compared with the *final* model. Since no model selection is performed, the training set is used to develop the models and the testing set to evaluate their generalization ability: (1) 54 ($_{54}C_1$) one-feature models, Fig. 5(a); and (2) 1431 ($_{54}C_2$) two-feature models, Fig. 5(b).

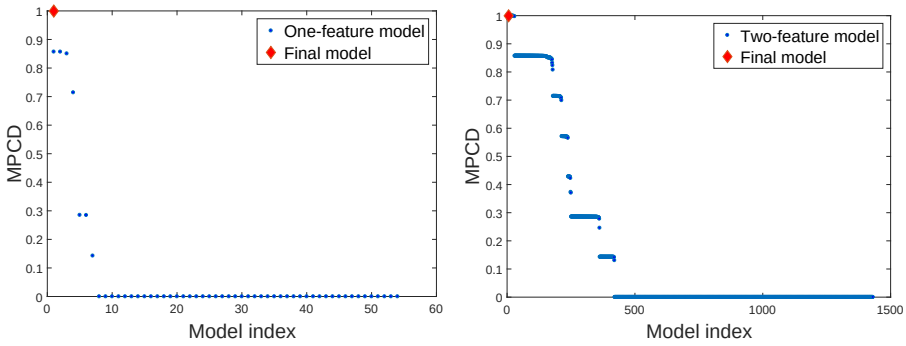
Based on exhaustive search, no single-feature model has better generalization ability. Whereas six two-feature models outperformed the *final* model, Table 4 summarize their relevant information. However, evaluating all possible combinations to find an optimal solution rapidly becomes unfeasible as the feature space grows.

The optimal solution could be defined as the model with the least number of features and the highest prediction ability. For example, in this case study, if there is no other model with an estimated $MPCD > 0.9998$, the optimal solutions would be model indexes 1032 and 1035, Table 4. However, since the number of combinations is huge, a model with more features may have greater *MPCD*. In this context, oftentimes due to the tradeoff between model complexity and prediction ability, there is no straight forward optimal solution. Instead, this tradeoff should be solved.

Although the *HFSPR* did not find the optimal solution, it did promptly find a good quality solution – a model that efficiently addresses the posed tradeoff. Fig. 5 show the relative location of the solution – *final* model.

5 Conclusions and Future Work

In manufacturing domain, traditional quality initiatives have merged to create a more coherent approach, therefore most mature organizations generate only



(a) One-feature models. (b) Two-feature models.

Fig. 5: *MPCD* exhaustive search in the one-feature and two-feature spaces.

Table 4: Top models (**HFSPR* solution)

Model index	Features	MPCD	FN
1032	26,33	0.9998	2
1035	26,36	0.9998	2
413	9,26	0.9997	3
1042	26,43	0.9997	3
1044	26,45	0.9997	3
1045	26,46	0.9996	4
Final	5,25	0.9995	5*

a few defects per million of opportunities. As shown in this paper, machine learning, pattern recognition and data mining techniques have the potential to detect these very few defects, and therefore move quality standards forward. However, several intellectual challenges have to be addressed to explode the full potential of *big data* initiatives.

A *Hybrid Feature Selection and Pattern Recognition* approach aimed at detecting rare quality events was developed. Although it does not guarantee to find the optimal solution (if exists), it does promptly find a good quality solution.

Although the proposed approach was inspired by the challenges that manufacturing companies are facing in detecting rare quality events – (1) feature explosion; and (2) high conformance production rates – it can be generalized to other domains, where the main challenge is to detect rare events through a parsimonious model.

In this paper, hyperparameter optimization was performed through a grid search. Future research along this path, can focus on developing an algorithm to improve the hyperparameter optimization process.

Bibliography

[1] C. Bishop, *Pattern Recognition and Machine Learning*. springer, 2006.

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

[3] 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 5^{th} *Int Conf on Software Eng and Service Science*, 2014, pp. 305–309.

[4] 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.

[5] 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.

[6] J. Devore, *Probability and Statistics for Engineering and the Sciences*. Cengage Learning, 2015.

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

[8] S. Wu, Y. Hu, W. Wang, X. Feng, and W. Shu, "Application of Global Optimization Methods for Feature Selection and Machine learning," *Mathematical Problems in Eng*, 2013.

[9] M. Baily and J. Manyka, "Is Manufacturing'Cool'Again," *McKinsey Global Institute*, 2013.

[10] 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.

[11] 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.

[12] C. A. Escobar, M. Wincek, D. Chakraborty, and R. Morales-Menendez, "Process-Monitoring-for-Quality — Applications," *to appear in SME Manufacturing Letters*, 2018.

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

[14] C. A. Escobar and R. Morales-Menendez, "Machine Learning Techniques for Quality Control in High Conformance Manufacturing Environment," *DOI:10.1177/1687814018755519, Advances in Mechanical Eng*, 2018.

[15] Carlos A. Escobar and Ruben Morales-Menendez, "Process-Monitoring-for-Quality — A Model Selection Criterion," *DOI:10.1016/j.mfglet.2018.01.001, SME Manufacturing Letters*, 2018.

[16] G. Chandrashekar and F. Sahin, "A Survey on Feature Selection Methods," *Computers & Electrical Eng*, vol. 40, no. 1, pp. 16–28, 2014.
[17] A. Ng, "On Feature Selection: Learning with Exponentially Many Irrevelant Features as Training Examples," in *Proc of the* 15^{th} *Int Conf on Machine Learning*. MIT, Dept. of Electrical Eng and Computer Science, 1998, pp. 404–412.
[18] H. Deng and G. Runger, "Feature Selection via Regularized Trees," in *Int Joint Conf on Neural Networks*, 2012, pp. 1–8.
[19] 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.
[20] K. Kira and L. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," in *AAAI*, vol. 2, 1992, pp. 129–134.
[21] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF," in *European Conf on Machine Learning*. Springer, 1994, pp. 171–182.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
[27] S. Arlot and A. Celisse, "A Survey of Cross-Validation Procedures for Model Selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.
[28] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Statistics Springer, Berlin, 2001, vol. 1.
[29] 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, lncs 10357.
[30] T. M. Inc. (2017) Logspace. [Online]. Available: https://www.mathworks.com/help/matlab/ref/logspace.html
[31] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer, 2013, vol. 810.
[32] J. Bergstra and Y. Bengio, "Random Search for Hyper-parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.
[33] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," *arXiv preprint arXiv:1502.02127*, 2015.