DESIGNING A QUALITY-CONTROL PROCEDURE FOR COMMUTATOR MANUFACTURING

Valentin Koblar[1,3]*, Bogdan Filipič*[2,3]

[1]Kolektor Group d.o.o., Vojkova ulica 10, SI-5280 Idrija, Slovenia
[2]Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
[3]Jožef Stefan International Postgraduate School, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
E-mail: valentin.koblar@kolektor.com, bogdan.filipic@ijs.si

ABSTRACT

This paper presents a methodology for designing an automated quality-control procedure in manufacturing graphite commutators for automotive industry. It focuses on assessing the quality of the copper-graphite joints on the commutators, using an advanced approach involving computer vision, machine learning and optimization. Based on the features of the captured images, classification models are built as a basis of the quality-control procedure, and optimized with respect to their accuracy and complexity. The preliminary results indicate that the proposed approach successfully identifies the features important for quality assessment, but needs further improvements to be applicable in industrial production.

1 INTRODUCTION

Quality-control requirements in automotive industry are very strict. Typically, the allowed proportion of defective supplied products is 1 ppm (part per million). To ensure the required quality of the supplied products, their quality has to be verified during and at the end of the manufacturing process. In addition, to meet the quality standards, every production procedure in the manufacturing process has to be controlled. In parallel with the improvement of the production processes, the quality control-systems have to be upgraded as well.

This paper deals with designing an automated procedure for inspecting the quality of commutators during their manufacturing at the Kolektor production plant. A commutator is an integral part of the commutator motor, mounted on the shaft of the rotor. Its function is to periodically reverse the direction of electric current in the rotor's winding and thus maintain commutation of the motor. During the motor operation, the commutator can be exposed to various stresses: mechanical, electrical, and, in the case of corrosive media, also chemical. To ensure reliable operation of the commutator over its entire lifetime, the commutator has to pass several quality tests. Currently, the quality of the manufactured commutators at the plant is checked only at the end of the production process. To reveal potential irregularities in the production process and minimize the waste of the raw material, faults on the semiproducts need to be detected as early as possible. To accomplish this goal, quality control after each operation in the manufacturing process is needed.

The plant produces commutators of different types for various applications, including graphite commutators for fuel pumps in the vehicle fuel systems. To prevent excessive wear and corrosion of commutators in the fuel pump application, these commutators are made of graphite and cooper. One of the phases in graphite commutator manufacturing is soldering of metalized graphite, representing the brush track, to the copper base. Soldering of metalized graphite is one of the most critical phases of commutator production, since the reliability of the end user application directly depends on the strength of the copper-graphite joint. In commutator production at the plant, there is currently no automated quality-control procedure of the soldering phase. All the manufactured graphite commutators are inspected manually, which is time consuming. Moreover, human errors are present, and the results of manual inspection may be subjective.

To overcome the drawbacks of manual product inspection, we aim to design an automated quality-control procedure for graphite commutator manufacturing. Currently, we focus on the soldering phase with the goal of synthesizing an advanced, information technology based procedure which is to employ computer vision for capturing images of the copper-graphite joints, digital image processing for extracting features from the images, and machine learning based classification for determining the quality of the joints, utilizing the extracted features. The design of the quality-control procedure is at its initial stage of producing a suitable classification model to predict the quality of the copper-graphite joints. To that end we apply machine learning from examples of the joint images. A challenge here is to produce a model accurate enough to ensure the required product quality, and sufficiently compact to allow for efficient on-line operation. In this paper we present off-line experiments involving machine learning and multiobjective optimization aimed at generating and tuning the quality prediction models.

The paper further describes data preparation for preliminary experiments in designing the quality-control procedure, presents learning and optimization of the classification models that represent the basis of the quality-control procedure, reports on the preliminary results, and concludes with a summary of work and directions for further improvements.

2 DATA PREPARATION

The initial stage of designing automated quality control is data preparation for machine learning. It consists of capturing the images of the joints resulting from copper-graphite soldering, extracting features from the images and assigning quality classes to the joints. An example of the copper-graphite joint can be seen in Figure 1. To capture as informative images as possible, several layouts of the camera and lightning were tested. In total, 394 images were captured.

Next, the images were then processed using a computer vision algorithm, designed in the LabView graphical programming environment [4]. The computer vision algorithm allows for the extraction of preselected image features to be later used as the attributes of the learning examples. In fact, these were values describing physical properties of the soldered joints, such as the size and the number of unsoldered areas, the size and the number of defected graphite areas, orientation of the cooper-graphite joint, the size of the solder spots on cooper, etc. There were 16 attributes describing each joint and they were all numerical.

Finally, the quality control experts were involved to assign a quality class to each joint. There were five quality classes possible (see Table 1 for their meaning and distribution).

Figure 1: The copper-graphite joint on a commutator

Table 1: Characteristics of the commutator soldering learning domain

Class	Number of examples	Frequency [%]
Well soldered	241	61.2
Excess of solder	49	12.4
Deficit of solder	36	9.1
Metalization defect	36	9.1
Disoriented	32	8.1
Total	394	100.0

The data preparation stage resulted in a set of learning examples described by attribute vectors, each containing 16 numerical attribute values and a discrete class value.

3 LEARNING AND OPTIMIZATION OF CLASSIFICATION MODELS

The next stage in designing automated quality control is induction of a classification model from the provided learning examples. This model is then aimed at classifying previously unseen instances from the copper-graphite soldering domain into appropriate classes. Machine learning of such a model can be viewed as search for a function that maps the attributes of the domain to the target classes. Generally, the machine learning algorithms do not provide optimal results without being properly tuned. For building models of high classification accuracy and, at the same time, reasonable size, it is not only necessary to choose an appropriate machine learning algorithm but also requires time-consuming setting of the learning algorithm parameters.

For building the classification model for the copper-graphite soldering domain, we used the Weka data mining environment [9], more specifically, its implementation of the C4.5 algorithm [6] for building decision trees, called J48. Decision trees are the most widespread classification models since they are easy to use and understand. However, the J48 learning algorithm enables us to set more than ten parameters that influence the resulting decision tree. In our study we selected five key parameters (see Table 2) and optimized them to balance between the accuracy and the complexity of the resulting classification models.

The classification accuracy of the models was assessed using 10-fold cross-validation, and their size was measured by the number of nodes in the decision tree. As Table 2 suggests, there are many possible combinations of the learning algorithm parameter values. Manual tuning of parameters and finding high-quality settings can be very time-consuming. It also requires expert knowledge of the used algorithms and properties of the learning domain.

An alternative approach to the manual parameter tuning is employing suitable optimization methods. The problem of tuning the machine learning algorithm parameters can be defined as an optimization problem. The objectives functions are the classification accuracy of the induced decision trees and their size. To solve this optimization problem, we integrated the Weka data mining environment with suitable optimization algorithms. A schematic representation of the machine learning algorithm parameter optimization is shown in Figure 2. Three optimization algorithms were tested in this scheme: grid search, random search and DEMO (Differential Evolution for Multiobjective Optimization) [7].

The grid search algorithm uses predefined discrete values of the parameter values. The algorithm systematically searches and evaluates decision models, built by using the grid values of machine learning algorithm parameters. Grid search is a computationally demanding process: by increasing the number of parameters and reducing the discretization step, the number of possible combinations increases exponentially. This method has proven to be more effective and efficient than manual search, as it provides more accurate decision models in shorter time [2].

The random search algorithm randomly selects discrete parameter values of the machine learning algorithm. It then evaluates decision models built by randomly generated parameter values and identifies the set of nondominated solutions to the multiobjective optimization problem. In general,

Table 2: Parameters of the J48 machine learning algorithm used classification model tuning

Parameter	Possible values	Default value
M – Minimum number of instances in a leaf	1, 2, ...	2
U – Use of unpruned trees	yes/no	no
C – Confidence factor used in postpruning	[0.01; 0.5]	0.25
S – Subtree raising operation in postpruning	yes/no	yes
B – Use of binary splits	yes/no	no

random search rather quickly converges to good solutions, but the found solutions are usually not optimal [8].

DEMO is a differential evolution algorithm [5] adapted for multiobjective optimization. The advantages of differential evolution are its simplicity and efficiency. Differential evolution starts with a population of random solutions from which progressively better solutions are obtained over generations. However, the major limitation of differential evolution originates from the requirement for numerical vector representation of solutions in the decision space that prevents the algorithm from being applicable to combinatorial optimization problems. A detailed description of the DEMO algorithm is available in [7]. In our experiment, the parameter values of the DEMO itself were set as follows: population size 20, number of generations 25, probability of crossover CR 0.6, differential evolution selection scheme DE/rand/1/bin and environmental selection procedure SPEA2 [1, 3].

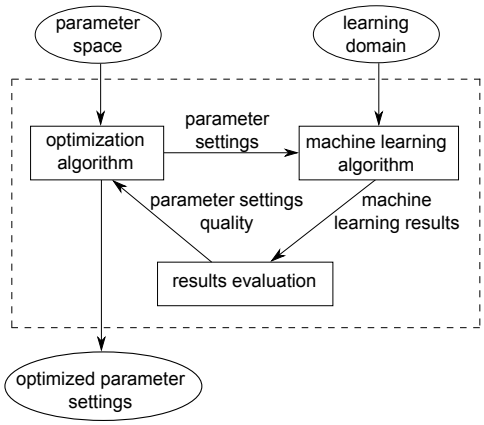

Figure 2: Schematic representation of machine learning algorithm parameter optimization

4 PRELIMINARY RESULTS

Preliminarily, we built decision trees for the copper-graphite soldering domain using the J48 algorithm default parameter settings. The default parameter values are shown in Table 2. We then compared the decision tree built using the default parameter values with trees built using the parameter values optimized by grid search, random search and the DEMO optimization algorithm. The result of building decision trees optimized with respect to two objectives, i.e., the classification accuracy and the size, is not a single decision tree, but a set of nondominated trees for each optimization algorithm. The decision tree built with the default parameter values and the nondominated solutions built using the optimized parameter values are shown in Figure 3. This figure presents the results of a single run of the optimization algorithms.

The comparison between the decision tree built using the default parameter values and the trees built with the optimized parameter values shows that the default decision tree tends to be larger and less accurate than the optimized trees. Because the grid search step size is determined by the number of evaluated solutions, in multiple runs the algorithm always finds the same nondominated solutions. In our experiments the stopping criterion was set to 500 evaluated solutions (built decision trees), defined by the grid discretization steps. Since random search and DEMO are stochastic algorithms, they find different parameter values in multiple runs. In the objective space this is reflected in different nondominated solutions.

The classification accuracy of the decision tree built using the default parameter values is 79.2 %. The highest classification accuracy of the decision tree found by the grid search algorithm was 80.5 %, but compared to the default decision tree, this tree is smaller. Decision tree with the highest classification accuracy was found by the random search algorithm. Its classification accuracy is 80.7 % and size 31 nodes. The decision tree with the highest classification accuracy found by DEMO was the same as the decision tree found by grid search. All three algorithms found several different parameter settings that results in the same nondominated solution in the objective space.

To understand the nondominated decision trees found by the optimization algorithms, we analyzed a number of them. The smallest nondominated decision tree that classifies the copper-graphite joints into all five quality classes, found by DEMO, is shown in Figure 4. Its classification accuracy is 77.2 % and the size 11 nodes. It is to be noted that this decision tree uses only 3 out of 16 attributes. The most informative attribute, *Max. size of graphite area*, describes the maximal size of the commutator area where the graphite metalization is not present. The second attribute, *Max. size of unsoldered area*, describes the maximal size of unsoldered area near the commutator cooper pad. If this area is too large, either there is a deficit of solder or the metalization of graphite is defected. Finally, the third attribute, *Size of unsoldered area in total*, describes the cumulative unsoldered area around the commutator pad. If this cumulative area is too large, there is

high possibility of the solder deficit. Knowing the background of the soldering process, we can claim that in practice the connection between the attribute *Max. size of graphite area* and the class *Excess of solder* is not so obvious. The decision tree shown in Figure 4 correctly classifies all instances of the class *Excess of solder*. Based on this fact, we can argue, that the described methodology in comparison to the usage of only machine vision application for classifications is advantageous.

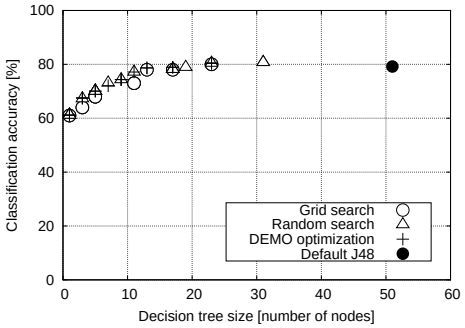

Figure 3: Results of decision tree building using default parameter values and optimized parameter values of the J48 machine learning algorithm

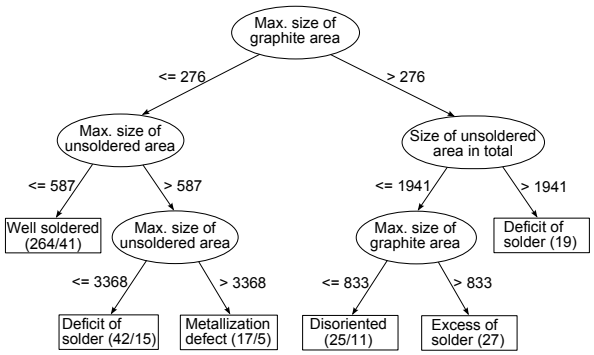

Figure 4: The smallest nondominated decision tree with all classification classes included, found by DEMO

5 CONCLUSION

We presented a preliminary study in the design of a quality-control procedure for graphite commutator manufacturing based on computer vision, machine learning and optimizations algorithms. The results of the initial experiments on real-world data show that, using the described methodology, we can enhance the performance of the classification models to serve as the basis of the quality-control procedure. Specifically, we used the J48 learning algorithm available in the Weka data mining environment, but the methodology could also be deployed using other machine learning algorithms.

From the perspective of practical implementation of the classification model, its high classification accuracy is essential. The highest classification accuracy achieved in the presented work is 80.7 %. Knowing the background of this domain, we can argue that this is still insufficient for practical application. To improve the classification performance, additional and more informative attributes from the captured images should be extracted. Moreover, tuning of the machine vision algorithm used for attribute extraction needs to be performed. Another direction for future work is to add additional objectives to the optimization problem. For example, from the practical point of view, minimizing the false-positive classification measure as an objective, would increase the reliability of the classification model.

ACKNOWLEDGMENT

This work has been partially funded by the ARTEMIS Joint Undertaking and the Slovenian Ministry of Economic Development and Technology as part of the COPCAMS project (http://copcams.eu) under Grant Agreement number 332913, and by the Slovenian Research Agency under research program P2-0209.

References

[1] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chhichester, 2001.

[2] Y. Ganjisaffar, T. Debeauvais, S. Javanmardi, R. Caruana, C. V. Lopes. Distributed tuning of machine learning algorithms using MapReduce clusters. *Proceedings of the KDD 2011 Workshop on Large-Scale Data Mining*, San Diego, 8 p., 2011.

[3] A. Ghosh, S. Dehuri, S. Ghosh (Eds.). *Multi-Objective Evolutionary Algorithms for Knowledge Discovery from Databases*, Springer, Berlin, 2008.

[4] T. Klinger. *Image Processing with LabView and IMAQ Vision*. Prentice Hall Professional, Upper Saddle River, NJ, 2003.

[5] K. V. Price, R. M. Storn, J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, 2005.

[6] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, 1993.

[7] T. Robič, B. Filipič. DEMO: Differential evolution for multiobjective optimization. In: C. A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Eds.), *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization EMO 2005, Guanajuato, Mexico*. Lecture Notes in Computer Science, vol. 3410, p. 520–533. Springer, Berlin, 2005.

[8] F. J. Solis, R. J. B. Wets. Minimization by Random Search Techniques. *Mathematics of Operations Research*, 6 (1): 19–30, 1981.

[9] I. H. Witten, E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd Edition. Morgan Kaufmann, San Francisco, 2005.