

A soft computing system using intelligent imputation strategies for roughness prediction in deep drilling

Maciej Grzenda · Andres Bustillo · Pawel Zawistowski

Received: 30 July 2010 / Accepted: 21 October 2010 / Published online: 19 November 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract A soft computing system used to optimize deep drilling operations under high-speed conditions in the manufacture of steel components is presented. The input data includes cutting parameters and axial cutting force obtained from the power consumption of the feed motor of the milling centres. Two different coolant strategies are tested: traditional working fluid and Minimum Quantity Lubrication (MQL). The model is constructed in three phases. First, a new strategy is proposed to evaluate and complete the set of available measurements. The primary objective of this phase is to decide whether further drilling experiments are required to develop an accurate roughness prediction model. An important aspect of the proposed strategy is the imputation of missing data, which is used to fully exploit both complete and incomplete measurements. The proposed imputation algorithm is based on a genetic algorithm and aims to improve prediction accuracy. In the second phase, a bag of multilayer perceptrons is used to model the impact of deep drilling settings on borehole roughness. Finally, this model is supplied with the borehole dimensions, coolant option and expected axial force to develop a 3D surface showing the expected

borehole roughness as a function of drilling process settings. This plot is the necessary output of the model for its use under real workshop conditions. The proposed system is capable of approximating the optimal model used to control deep drilling tasks on steel components for industrial use.

Keywords Deep drilling · Incomplete data · Imputation · MQL · Surface roughness · Multilayer perceptron

Introduction

Deep drilling and lubrication systems

In recent years, interest has grown in performing machining operations under dry or near-dry conditions, which may be partly explained by health and also economic reasons (Nandi and Davim 2009). One of the most complex manufacturing processes to change from high pressure conventional flood cooling to near-dry cut is the drilling of deep holes, also known as deep drilling. This change is because the coolant is a key factor in chip evacuation from the cutting area. Dry cutting does not allow chip evacuation and tool life is greatly reduced under these cutting conditions. New lubrication systems have therefore been developed such as Minimum Quantity Lubrication (MQL) (Braga et al. 2002). In MQL, a very small amount of oil (less than 30 ml/h) is pulverized into the flow of compressed air to cool the surface, and, even more importantly in deep drilling, to assist chip evacuation from the tool tip to the external surface of the hole.

The ecological impact of MQL systems is clearly lower than the traditional high-pressure coolant fluid systems. Doubts persist, however, over the degree of productivity and quality that these systems will be able to achieve. Among the main reasons for these doubts are the problems that relate to the investigation and visualization of the impact of cut-

M. Grzenda (✉)

Faculty of Mathematics and Information Science,
Warsaw University of Technology, Pl. Politechniki 1,
00-661 Warsaw, Poland
e-mail: M.Grzenda@mini.pw.edu.pl

A. Bustillo

Department of Civil Engineering, University of Burgos,
Edificio C. C/Francisco de Vitoria, s/n, 09001 Burgos, Spain
e-mail: abustillo@ubu.es

P. Zawistowski

Faculty of Electronics and Information Technologies,
Warsaw University of Technology, ul. Nowowiejska 15/19,
00-665 Warsaw, Poland
e-mail: p.zawistowski.2@elka.pw.edu.pl

ting conditions on the key attribute of a drill hole i.e. its roughness. In reality holes that are bored under exactly the same experimental conditions, using exactly the same tool settings will often have differing roughness measurement values. Some of the reasons for this phenomenon are related to the cutting process itself. There are also other reasons related to the limited precision of roughness measurement. Cutting processes depend also on many variables that can not be easily measured, the most important being: irregularities in the metal blank to be drilled, tool wear, inhomogeneities in cooling performance and chip formation (Benardos and Vosniakos 2003). Therefore, some process variables, such as axial force, are measured, as accurately as possible, during the drilling process to take these factors into account. The roughness measurement of the drill also has an instrumental error. In addition, the whole drill length can not be measured, which therefore adds a further error. Moreover, to decide on optimal tool settings, a process engineer should be provided with surface plots showing the impact of tool settings on roughness. These plots should approximate the data coming from real experiments, taking into account the aforementioned diversity of roughness measurements under the same cutting conditions. However, only a limited number of experiments may be set up for such surfaces. Hence, a model is needed that approximates existing data and predicts roughness values for tool settings that are not covered by experimental values. Such a model would contribute to maximising the industrial use and development of MQL High Speed deep drilling on steel components. High-speed drilling is an especially interesting industrial process, due to the broad use of steel as a base material for different kinds of high value industrial components. In fact, one of its main industrial applications is the drilling of boreholes for knockout pins or coolant circuits in moulds and dies.

There is no standard definition for *deep drilling*, although it is usually defined by considering the length-to-diameter ratio of the drill-hole. Whenever drill-hole length is 2–3 times larger than its diameter, almost no cutting fluid reaches the drill tip, mainly because the drill and the counter-flow of chips restrict further penetration (Kubota and Tabei 1999). When the drill-hole length is 3 times larger than the drill-hole diameter, drilling with conventional flood cooling may be considered a near-dry cutting process, which is the main reason why MQL is often preferable. MQL industrial applications are therefore appropriate for drill-hole lengths that are 4 times larger than the drill-hole diameter ($4 \times D$). In line with other authors (Hayajneh 2001; Weinert et al. 2004) and for the purposes of the experimental tests described in this study, *deep drilling* will be therefore defined as any drill-hole length that is 4 times larger than the drill-hole diameter.

Research on MQL deep drilling is still mainly focused on aluminium, as dry deep drilling is not possible on this material, because of its high adhesion to the drill flutes

(Braga et al. 2002; Davim et al. 2006). On steel, research has mainly focused on experimental tests to define tool wear (Heinemann et al. 2007), tool life (Heinemann et al. 2006; Filipovic and Stephenson 2006) and surface quality (Davim et al. 2006; Zhang and Chen 2009) in relation with MQL and cutting conditions. Certain studies have attempted to predict chatter phenomena in deep drilling using analytic solutions (Mehrabadi et al. 2009). With regard to soft computing techniques, many publications are either devoted to drilling modelling (Chandrasekaran et al. 2010) or refer to the problem (Choudhary et al. 2009), but not so many examine deep drilling and even fewer study MQL deep drilling, where the physical phenomena differ from standard drilling. Fuzzy logic has been used to predict forces and surface quality on MQL deep drilling of aluminium (Nandi and Davim 2009), drill life (Biglari and Fang 1995; Jantunen and Vaajoensuu 2010) and better cutting conditions (Hashmi et al. 2000) in deep drilling of steel with conventional flood cooling. Artificial neural networks (ANN) modelling approaches have been used for predicting burr size (Davim et al. 2006) and drill wear (Sanjay et al. 2005) in deep drilling of steel with conventional flood cooling. Finally, genetic algorithms have been applied to maximise metal removal rate in a standard drilling process (Zang et al. 2006).

However, the way soft computing techniques can be applied to predict borehole roughness under MQL deep drilling conditions remains an open issue. The term roughness is used in the remainder of this paper to refer to the roughness of a borehole as a result of deep drilling.

Although all these approaches could lead to good results under laboratory conditions, a further fundamental requirement has to be fulfilled under industrial conditions. In the world of manufacturing, experimental testing is particularly expensive as it often involves the use of high-precision machinery and costly processes. But these experimental tests are essential to adjust the mathematical models that simulate the manufacturing processes. Among these models, artificial intelligence (AI) techniques have been shown to provide good accuracy in industrial problems governed by high complexity phenomena and a large number of variables that influence the outcome. But AI techniques generally require the production of reliable models and the completion of an extensive battery of experimental tests with a significant variation of the process parameters, in many cases outside optimal working limits. This implies a high experimental cost and, ultimately, it means that these techniques might be adjusted with overly limited data sets, which would therefore present high error rates under industrial conditions. This lack of broad datasets is accentuated by the fact that some of the tests that could be done under industrial conditions are not completed. Usually the datasets contain variables that are not always measured because data acquisition is frequently not considered a core task in many enterprises. In the case

of drilling data used for this study, a significant part of the data set was incomplete. On the other hand, as in many other cases, it remained unclear whether sufficient data was forthcoming from the experiments that had already been carried out to obtain a high-quality model of the process. Frequently, incomplete data is removed from the data set, which may negatively impact the quality of a model. Thus, among its objectives, our work investigates the way incomplete data can be fully exploited to deliver a prediction model of borehole roughness and assesses the need to perform further experiments to acquire additional input data for modelling purposes.

Dealing with incomplete data

Numerous techniques have been proposed to deal with incomplete data. In general, there are two approaches when faced with incomplete data that are used to model industrial processes. One possibility is to revise the data set and delete all the impaired instances or even attributes (this approach is sometimes called *complete case analysis*). However, when the data are valuable and scarce, this approach may significantly reduce the quality of a model. Thus, another solution is to impute the missing values using a proper method. The imputation approach makes it possible to avoid deleting possibly useful information, but can equally introduce errors into the data set. Nevertheless, in many cases imputation is the best solution to incompleteness.

Filling in missing data requires choosing, from among all the available methods, the one that gives the best possible results. One of the most popular methods is probably the *EM algorithm* presented, among others, by Dempster et al. (1977) and Schafer (1997). The procedure assumes a distribution of the missing values and then fills them in using a two step iterative procedure: the *E-step* estimates the expected values of the missing data and the *M-step* changes the parameters of the distribution to maximize the likelihood of the data. Another interesting iterative method called non-invasive imputation has been proposed by Gediga and Düntsch (2002). Abdella and Marwala (2005) proposed the use of neural networks as an imputation method. A framework for dealing with incomplete data for use in data mining was also proposed by Wei and Tang (2003). One-class classifiers were proposed as a method by Juszczak and Duin (2004). One of the most popular techniques, the kNN algorithm was used as an imputation method by Acuña and Rodriguez (2004), Batista and Monard (2001), Hu et al. (1998) and Jönsson and Wohlin (2006). Finally, Hu et al. (1998) gave an overview of popular imputation methods. Last but not least, simple, yet popular techniques of replacing missing values with mean or median values are used. A further issue to consider is that some imputation techniques such as these based on k-means clustering require method parameters to be set.

When there is a need to fill in missing values for more than one attribute of a data set, the choice of a single method to perform all the imputations becomes even more difficult, or impossible. In particular, no single method suitable for all the attribute types may exist at all. In the latter case, performing imputation means finding the combination of different imputation techniques and their settings.

Solution overview

As the problem of finding the optimal set of imputation methods and their attributes is a difficult optimisation task, an evolutionary algorithm was applied in the case under analysis to search for the best imputation strategy. Moreover, the strategies are evaluated in view of the problem for which the data set is used. In other words, the suitability of an imputation strategy is evaluated in terms of its impact on the prediction process that is performed using the data set. This approach has been shown to outperform different standard imputation techniques (Zawistowski and Grzenda 2009).

The framework used in this work combines genetic algorithms and multilayer perceptrons. Genetic algorithms have been used to evolve method vectors representing imputation strategies. Multilayer perceptrons (MLP) have been applied to represent roughness prediction models. The error rate of prediction performed for the data sets, which are imputed using a strategy, provides the basis for a fitness function. In other words, the higher the accuracy of the roughness prediction performed on the imputed data set, the better the imputation strategy used to fill in the missing values. Once the imputation strategy has been determined, the impact of data set size on the roughness prediction accuracy may be investigated. This was done using both incomplete and complete records.

Finally, the MLP-based model presented in this work evaluates the impact of cutting conditions and lubrication systems on the surface quality of a borehole. This provides the basis for an industrial solution i.e. the set of 3D models showing the impact of drilling process settings on the expected roughness. The entire data flow is summarised in Fig. 1.

The remainder of this work is organised as follows:

- Data collection process and industrial experiments are discussed in section “Experimental procedure and data collection”,
- The imputation algorithm used to increase the number of measurement records available for modelling purposes is outlined in section “Imputation algorithm”. In particular, the formal method of evaluating the quality of imputation in terms of prediction modelling is proposed. This provides the basis for analysing the results of the imputation of roughness data set and to decide whether further exper-

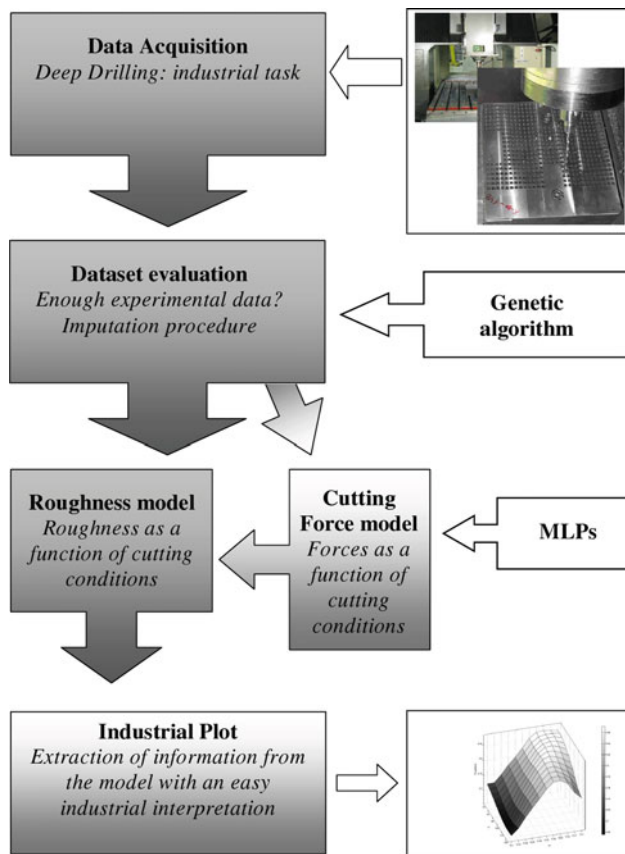


Fig. 1 Data acquisition, axial force and roughness modelling

iments could result in new models that might improve roughness prediction accuracy.

- Once the size of the data set is determined to be sufficient, the results of roughness modelling are summarised in section “Roughness modelling results”.
- The way the model can be used by process engineers is outlined in section “Industrial applications”. A sample roughness surface, which can be used to optimise the drilling process is also provided.
- Finally, the conclusions are summarised under section “Conclusions and future works”.

Experimental procedure and data collection

Two different milling centres were used to obtain the experimental data: one for traditional coolant tests and the other for MQL tests. In both cases the blank material used for the tests was F114 steel. Considering the dimensions of bore holes for knockout pins, two holes diameters were chosen: 5 and 10 mm. Two hole lengths were tested for each diameter: 5 times the diameter ($5 \times D$) and $8 \times D$. The tools were selected from two different providers to test different geometries: HAM and Mitsubishi. More extensive tests were

Table 1 Cutting conditions proposed by the tool supplier

Tool model	Tool code	V_c (m/min)	av (mm/rev)
HAM-286 ($5 \times D$), 5 mm diameter	1	80–128	0.10–0.2
HAM-286 ($8 \times D$), 5 mm diameter	1	70–112	0.10–0.2
HAM-292 ($5 \times D$), 10 mm diameter	2	80–128	0.10–0.2
HAM-292 ($8 \times D$), 10 mm diameter	2	70–112	0.10–0.2
Mitsubishi MPS0500-DIN-C VP15TF	3	50–100	0.15–0.25
Mitsubishi MPS0500-L8C VP15TF	4	50–100	0.15–0.25

done with HAM drills because they showed better performance. The tool references are as follows: Tool 1— $5 \times D$ HAM-286; Tool 3—Mitsubishi MPS0500-DIN-C VP15TF; Tool 2— $8 \times D$ HAM-292; Tool 4—Mitsubishi MPS0500-L8C VP15TF.

Cutting conditions were chosen to reproduce real industrial conditions. Each tool was tested under those conditions that provide higher productivity depending on the lubrication system. In this way a comparison of productivity and quality between the two lubrication systems could be obtained for the same set of tools. In any case, all the cutting conditions for the tests were defined within the cutting range proposed by the tool manufacturer. These cutting ranges are summarised in Table 1.

The experimental tests should provide datasets with 7 input variables: tool type, tool diameter, hole length, federate per revolution av , cutting speed V_c , type of lubrication system and axial cutting force. The experimental design includes combinations of three such variables: tool diameter, hole length and coolant system. For the other variables, which relate to the cutting conditions, the experimental design aims to achieve higher productivity, lower cutting forces and good surface quality (roughness lower than $1 \mu\text{m}$). The whole experimental test includes 90 different conditions. All the tests were repeated to increase the amount of data. Thus, a data set of 220 records was obtained. Table 2 shows the selected values for each variable in each test.

The tests were performed along the Z-Axis of the milling centres. Hence, the monitoring of the Z-Axis feed motor consumption provided the axial cutting force during the drilling operation because both variables are proportional. Roughness was measured using a rugometer Mitutoyo SV-2000N2. The roughness of the boreholes is obtained as a mean value of the roughness measured along the first 10 or 15 mm of the hole, depending on its total length.

After the experiments, the measurement process and data that had been collected were carefully investigated. Unfortunately, it was revealed that some axial forces had not been correctly measured, so they had to be removed from the data set. As a consequence, the experimental data was incomplete. This is very often an industrial reality, where not all the test

Table 2 Cutting conditions selected for the experimental tests

Tool	Diam.	Hole length (mm)	av (mm/rev)	Vc (m/min)	Coolant
1	5	25	0.12, 0.15, 0.2, 0.25	70,80,90,100,125,130	1
1	10	50	0.2, 0.25, 0.3	70,80,90,100,110,125	1
1	5	25	0.12, 0.15, 0.2, 0.25	70,80,90,100	2
1	10	50	0.15, 0.2, 0.25	70,90,100	2
2	5	40	0.1, 0.15, 0.2, 0.25	70,80,100,125	1
2	10	80	0.15, 0.2, 0.25, 0.3	80,90,100,125	1
2	5	40	0.1, 0.12, 0.14	70,80,90,100	2
2	10	80	0.1, 0.12, 0.14	80,90,100	2
3	5	25	0.15, 0.2, 0.25	70,90,100,110,125	1
4	5	40	0.15, 0.2, 0.25	70,90,100,110,125	1

results are acceptable, because some of the variables are not properly acquired. In the presented case, axial force is one of the main inputs to model the drilling process. Unfortunately, it can not be calculated directly from the other variables. In 44 experiments, this variable was not reliable and had to be removed from the data set. Therefore 20% of the data set was incomplete because of the lack of this variable.

Imputation algorithm

Motivation

As in the case of many other data sets, some attribute values were missing from the original data set. Thus, imputation techniques were considered to increase the number of complete records. The primary objective of using imputation was to improve the accuracy of roughness prediction by building prediction models that take account of all available records, including those which were originally incomplete. Moreover, imputed data were used to decide whether further experiments, as well as the additional records that would be collected were the experiments conducted, would result in significant improvements in terms of roughness prediction accuracy.

Model-based evaluation

When evaluating an imputation method, the distances between the original and the filled in values are often used as a performance measure. However when the data set is noisy, and this is often the case, such an approach may lead to efforts to recreate noise. Bearing in mind the goal, which is roughness prediction, the methods that impute the missing data in a way that does not distort the prediction model can be considered suitable for roughness prediction. There is no need to try to find a method which could recreate the missing part precisely, with all the noise that might have been present in the

original measurements. Model-based evaluation of imputation methods is therefore applied (Zawistowski and Grzenda 2009). The algorithm presented below, which is based on the evolutionary development of method vectors, was applied to select both the imputation method and its parameters.

Method selection

For a given set of imputation methods Γ and a given data set D , in which attributes a_1, \dots, a_n suffer from incompleteness, let $V = [m_1, \dots, m_n]$ be a *method vector*, and $m_i \in \Gamma$ —the imputation method used to fill in missing values of attribute a_i . Thus, method vectors are vectors in an n -dimensional imputation method space defined by a specific data set D and a specific set of imputation methods Γ . Such vectors are used to fill in the incomplete data sets. The goal is of course to achieve the best possible results.

Let M denote a model to be used. The role of this model is to address the problem for which the data set is used. In the presented case, the role of the model is to perform roughness prediction. Thus, in general $M : R^N \rightarrow R^C$, while $D \subset R^N$ and $n \leq N$. In particular, all attributes can miss some values. In the latter case, $n = N$. In case of classification or prediction of a single signal, the number of output signals produced by model M is $C = 1$. Model M can be implemented using different techniques. In our case, the model is implemented using a bag of multilayer perceptrons.

Continuing the notation, let $e_M(D)$ denote the mean absolute error of model M on data table D . Furthermore D_V is the data table created from D by filling in all the missing values using methods from vector V . Some of the methods in Γ have parameters and in such situations not only the method vectors have to be found, but also their parameters. Let $P(V) = \{p : p \in P_{m_1} \times \dots \times P_{m_K}\}$ denote the parameter space for vector V , where $K \leq n$ and P_{m_i} is the parameter space for the i -th parametric method. D_V^p denotes the data

table created from D by filling in all the missing values using methods from vector V with parameters $p \in P(V)$.

The objective of finding the best imputation method vector with parameters, means actually finding the best pair $[V^*, p^*]$, where $p \in P(V^*)$, and it follows that:

$$e_M(D_{V^*}^{p^*}) = \min_{V, p \in P(V)} e_M(D_V^p). \quad (1)$$

An imputation procedure that uses method vectors may now be proposed. For a given data model and a given incomplete data table, the imputation procedure proposed in Algorithm 1 is used. This formulates quite a difficult optimization

Input: Γ - a set of imputation methods to be used, D - an incomplete data table, M - a model to be used for data table D
Result: $D_{V^*}^{p^*}$, which is a complete data set to be used by model M
begin
 $[V^*; p^*] = \text{find a pair for which Eq. 1 holds;}$
 $D_{V^*}^{p^*} = \text{fill in the incomplete data table } D \text{ using methods defined in vector } V^* \text{ with parameters } p^*;$
return $D_{V^*}^{p^*};$
end

Algorithm 1: Method vector based imputation procedure

task, which can be solved using a genetic algorithm. Obviously, in most cases only suboptimal solutions may be found.

The genetic algorithm works on a population of method vectors. Each individual consists of n genes representing imputation methods used for individual incomplete attributes. To simplify the notation, it can be assumed that there are two types of genes: non-parametric genes—used for non-parametric methods and parametric genes applied for parametric methods. This means that parametric genes are actually pairs $[m, p_m]$ where m is an imputation method, and p_m is the vector of parameter values for this method.

The proposed algorithm uses two types of genetic operators to diversify the population, namely crossover and mutation. Both operators have two versions. One version is used for non-parametric and one for parametric genes. In general, in the first stage of the algorithm, changes can occur to the set of methods and the parameters of methods comprising the imputation strategies. In the final stage of the algorithm, changes are applied only to the parameters. These changes are applied by a mutation procedure. Further details on the method, mutation and crossover operators and the results of benchmark tests can be found in (Zawistowski and Grzenda 2009).

Evaluation of imputation results

When filling in missing data, some criterion is needed to evaluate whether or not the obtained results are acceptable. The

following two scenarios are considered, assuming a data set with one incomplete attribute for the sake of simplicity.

In a *classic* scenario, a complete data set is obtained, split into training and testing subsets from which some attribute values are removed. The training subset is then used to tune the parameters of the imputation method and the testing subset—to select the best available method. Performance evaluation in such a scenario is done by measuring the distance between the original and the filled-in attribute values.

The classic scenario has two important requirements: the complete data set has to be sufficiently large (the precise meaning of sufficiently large of course depends on the data set itself) and the distribution of the actual missing values has to be considered. Because these demands may be difficult to meet when solving real life problems, using a model-based evaluation scenario might be a feasible alternative.

In a simple *model-based evaluation* scenario, an incomplete data set is split into training and testing subsets. The training subset is used to prepare imputation methods and the testing subset—to select the best one. These steps are similar to the classic scenario, however the difference lies in the imputation method evaluation procedure. When a method is evaluated, it is used to prepare a complete data set on which a model is trained and tested using cross-validation. The model's performance is the evaluation value of the method. This scenario does not involve preparing an incomplete data set from a previously complete one. It does however facilitate the best choice of imputation method with respect to a given model.

When selecting an imputation method for incomplete data sets obtained from industrial experiments, the following factors should be considered:

- The real values of incomplete attributes are actually unknown, thus they can not be used for evaluating the method. The solution would be easy if the missing values were available for comparison purposes, so one approach could be based on using artificially removed data for testing different imputation techniques. Such a validation procedure could however lead to problems with the proper estimation of the missing data distribution. In other words, when an incomplete data set is artificially created, the existing values are replaced with null values in randomly selected instances. Hence, the imputation technique found to deal with such an incomplete set might be optimal, but for the set containing randomly placed null values, not for the set of actually incomplete instances arising from partly successful measurements. Expressed more simply, the distribution of actual incompleteness may not follow an *a priori* chosen distribution over the entire set of experiments e.g. uniform distribution. This phenomenon is very likely to occur due to technical reasons i.e. problems with completeness/correctness of the

experimental data may affect experiments sharing some common features.

- Apart from that, in the case of attributes originating from measurements, the errors which are present in the complete values also have to be considered. The question of whether the imputation method provides correct values may be answered more easily, if we put the question in the proper context. As in our case, the goal is to use the imputed data to create a model predicting borehole roughness, the question is how to determine the impact of training using imputed data on the obtained roughness prediction model. A procedure to evaluate whether the training process using imputed data produces prediction models that are as good as the training process that only uses complete data is proposed to determine this impact. To perform such an evaluation, a characteristic relating training set size to the obtained model’s mean square error can be used. This characteristic describes changes in the performance of the models when the training set size increases. The imputed data can be considered correct if the characteristic obtained while using imputed data is similar to the characteristic that is obtained when using only complete data. Therefore, the correlation calculated between the two aforementioned characteristics can be used as the imputed data validation measure.
- The other factor to consider is that the mean squared error of the testing set, in the case of a limited data set size, strongly depends on the division of the available data into training and a testing part. Thus, cross-validation should be applied.
- To make the evaluation of imputation techniques fully objective, all the performance measures should be calculated on the testing sets, that are not used for the training and that only contain originally complete instances.

Thus, an algorithm is proposed to train the prediction models on the data sets of the same size. The idea is to train the models using the data sets of the same size, but using complete data in one case and imputed data together with originally complete instances in the other. What is important, in both cases is that the models trained on the data sets containing an identical number of instances should be used for comparison. For every training set size, a cross-validation algorithm is run, as stated in Algorithm 4. The evaluation of the imputed data is performed using the quality of the roughness prediction models verified on the testing data sets, as summarized in Algorithm 3. In this way, performance rates are calculated for the two cases: a training set that only contains originally complete instances and a training set that contains imputed instances with a number of originally complete instances that are added to make the size of the latter set identical to the set of originally complete instances. Thus, the method described in Algorithms 2, 3, and 4 can be

summarized as follows: an investigation is performed if the models trained using imputed data provide predictions for testing instances that are as good as the corresponding models trained with the originally complete data, assuming both training data sets contain an identical number of instances.

In particular, Algorithm 2 calculates Pearson’s correlation coefficient between the two described characteristics. If this coefficient is close to 1 then we might assume that the imputed data are safe to use (i.e. do not distort the results). As the proposed algorithm depends on the order in which data instances occur in data table D^C , its results are averaged for different permutations of those instances.

```

Input:  $D^C$  - data table containing only instances that were originally complete;  $D^I$  - data table with imputed instances only;  $m$  - the model to be used;  $k$  - the number of cross-validation folds;  $l$  - number of training set sizes to check,  $\alpha$  - minimal size of individual  $D_i$  set
Constraints:  $\exists_{\alpha, \beta \in \mathbb{Z}^+} |D^C| = l * k * \alpha \wedge |D^I| = \alpha * (k - 1) * \beta$ 
Result: Imputation validity coefficient
begin
   $\{D_1^C, \dots, D_l^C\} = Divide(D^C, l);$ 
   $\alpha = \frac{D_1^C}{k};$ 
   $D = \emptyset;$ 
   $e_C = [0, \dots, 0]^T, e_I = [0, \dots, 0]^T;$ 
  for  $i = 1, \dots, l$  do
     $D = D \cup D_i^C;$ 
     $e_C[i] = Evaluate(m, D, \emptyset, k);$ 
     $e_I[i] = Evaluate(m, D, D^I, k);$ 
  end
   $\hat{e}_C = \{e_C[i] : \exists_j i * (k - 1) * \alpha = j * (k - 1) * \alpha + |D^I|\};$ 
   $\hat{e}_I = \{e_I[i] : \exists_j i * (k - 1) * \alpha + |D^I| = j * (k - 1) * \alpha\};$ 
  return Pearson’s correlation between  $\hat{e}_C$  and  $\hat{e}_I$ ;
end

```

Algorithm 2: $Validate(D_C, D_I, m)$: Algorithm used to validate imputation results. Algorithms 4,3 are used as sub procedures.

Imputation of the roughness data set

The purpose of using the roughness data set was to create a model that is capable of predicting roughness, given the other six attribute values. Because of the limited size of the data set and its incompleteness, an attempt to fill in the missing values using the presented approach was made to preserve as many data rows as possible.

In order to use this method, a roughness modelling technique had to be established, which could be used to perform model-based evaluation. A robust *black box* method that could be applied to this problem is the multilayer perceptron. As the goal is to fill in the missing data in a way that will maximize the quality of the models created with that data set, the individuals in the evolutionary algorithm were evaluated according to errors of bags of MLP neural networks

trained on the imputed data tables. The bagging technique (Breiman 1996) was used in order to lower the influence of noise and outliers and obtain more robust models.

Input: m - model to be used, D - a complete data table for cross-validation, \hat{D} - data table with additional training data, k - number of cross-validation folds
Result: Model m 's error
begin
 $\{D_1, \dots, D_k\} = \text{Divide}(D, k);$
 $e = [0, \dots, 0]^T;$
for $i = 1, \dots, k$ **do**
 $m_i = \text{train model } m \text{ on data set } (D - D_i) \cup \hat{D};$
 $e[i] = \text{calculate the mean square error of model } m_i \text{ on data set } D_i;$
end
return $\text{mean}(e[1], \dots, e[k]);$
end

Algorithm 3: $\text{Evaluate}(m, D, \hat{D}, k)$: Model performance evaluation. Algorithm 4 is used as a sub procedure.

Input: D - data table, k - number of sub-tables
Result: D_1, \dots, D_k
begin
 $\alpha = \lceil \frac{|D|}{k} \rceil;$
 $\{d_1, \dots, d_{|D|}\}$ - all data instances from D ;
for $i = 1, \dots, k$ **do**
 $\beta = (i - 1) * \alpha + 1;$
 $D_i = \{d_\beta, \dots, d_{\min(\beta + \alpha, |D|)}\};$
end
return $D_1, \dots, D_k;$
end

Algorithm 4: $\text{Divide}(D, k)$: Data division.

When an individual was evaluated, it was first used to fill in all the missing values. A 10-fold cross validation procedure was then applied to the imputed data set. In each fold, a bag of 10 MLP networks was trained and tested on the appropriate training/testing subsets. The Mean Squared Error (MSE) of all the cross validation network test errors was used as the evaluation value of the individual. The networks created during this procedure had a single hidden layer with 3 neurons and a hyperbolic tangent transfer function. Training was performed using 50 epochs of the resilient back propagation algorithm.

The incomplete data set consisted of 219 rows among which 44 rows had missing axial force values. As only one of the attributes suffered from incompleteness, it is important to note, that evolution only played a significant role in the parametric domain (to fine tune the methods' parameters), while a full search was actually done in the imputation method domain (to choose the best possible method).

The imputation methods available to the algorithm were as follows :

- random imputation with data distribution estimated using the complete instances
- mean/mode/median imputation
- non-invasive imputation (Gediga and Düntsch 2002)
- k-nearest neighbour clustering with mean/mode/median as aggregate functions
- Self-Organizing Map (SOM) clustering with mean /mode/median as aggregate functions.

The algorithm was run 50 times, with 50 generations of 10 individuals for each run. The best imputation method found by the algorithm was k-Nearest Neighbours with $k=2$ and mode as the aggregate function. This method was used to produce a complete data set which then was used to validate the effects of imputation.

Validation was performed using Algorithm 2. The algorithm was run 100 times for different permutations of the complete data table which resulted in an averaged correlation coefficient value of 0.992. This value suggests, that the imputed values are valid. The values obtained during validation are presented in Table 3 and in Fig. 2.

- The quality of the models trained on complete and imputed data is virtually the same as the quality of the models trained on only complete data set of the same size. Figure 2 clearly shows the benefits drawn from increasing the training set size. Although, in both cases, the results for smaller sets are distorted due to high variance in the testing sets, once the total number of records in the training set exceeds 110, the number of complete instances is sufficient for training and testing the model properly. Thus, the value of the imputed data in terms of its impact on modelling is almost identical to the value of complete instances and can be used to increase the training set size and thereby improve the model's performance.
- What should be emphasized is that in cases where the training set sizes are the same, the error rate shown by a model trained only with originally complete instances is slightly lower. This is illustrated in Fig. 2. Moreover exact error rates reported for training data sets of the same size clearly show this tendency, as may be seen in Table 3. Nevertheless, by using imputed instances, we can benefit from a higher overall number of instances present in the training data set. This results in better estimation of prediction model parameters. Thus, the performance of the models trained with all available complete instances and imputed instances turns out to be better than the performance of a model that is not trained with imputed instances i.e. trained with a lower number of instances.
- The error rate of the models trained on the set containing all available complete training instances and imputed instances (0.1115) is lower than the corresponding error

Table 3 Results used to calculate the correlation coefficient during validation

Training set size	Complete data		Complete+Imputed data	
	MSE	Std. dev.	MSE	Std. dev
8	0.4406	<i>0.4760</i>	—	—
16	0.3304	0.2709	—	—
24	0.2800	0.1803	—	—
32	0.2461	0.1353	—	—
40	0.2274	0.1222	—	—
48	0.2009	0.1000	0.2695	<i>0.4544</i>
56	0.1857	0.0912	0.2214	0.2478
64	0.1665	0.0751	0.2004	0.1660
72	0.1545	0.0634	0.1772	0.1217
80	0.1462	0.0576	0.1646	0.1146
88	0.1404	0.0499	0.1510	0.0801
96	0.1342	0.0435	0.1469	0.0756
104	0.1283	0.0455	0.1392	0.0666
112	0.1230	0.0410	0.1288	0.0564
120	0.1214	0.0360	0.1270	0.0517
128	0.1178	0.0325	0.1247	0.0444
136	0.1151	0.0325	0.1220	0.0448
144	—	—	0.1189	0.0440
152	—	—	0.1155	0.0360
160	—	—	0.1138	0.0346
168	—	—	0.1123	0.0333
176	—	—	0.1115	0.0328

As the imputed data were added only to the training sets, the test set sizes were the same. For example, the two italicized values depict the variation of tests performed on sets of equal size

of the models trained with complete training instances being 0.1151. In other words, after adding imputed instances an error reduction of 0.0036 is observed. Therefore, on the one hand, further error reduction caused by the use of imputed data has been achieved. On the other hand, taking into account a similar value of imputed and complete data, additional experiments that aim to increase the number of available instances are not justified due to limited error reduction. The same conclusion can be drawn from Fig. 2.

To sum up, imputed data can be used as a valid replacement of missing data. The roughness prediction models trained with extended data sets including the imputed instances show lower MSE error rates than their originally complete subsets. At the same time these models provide a valuable insight into further error reduction tendency. Taking into account the error reduction curve shown on Fig. 2, the existing set of measurements can be considered to be sufficient from the modelling perspective.

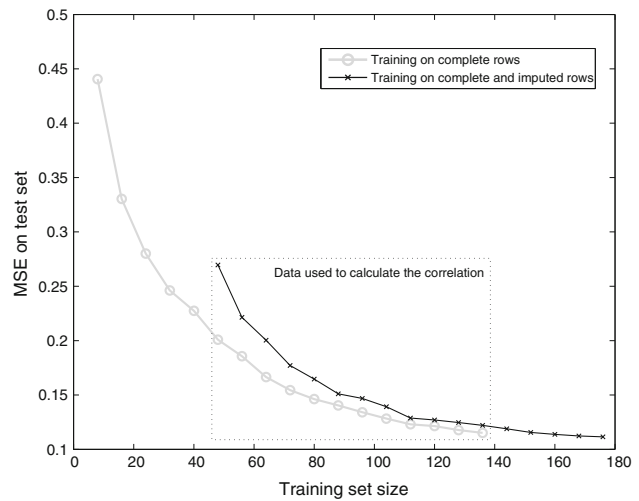


Fig. 2 Model performance characteristics used to validate imputation results

Table 4 Roughness modelling results

Model	MSE	σ_{MSE}
MLP	0.090161	0.004449
Naive	0.396846	0.001952

Roughness modelling results

The imputed data set was used to create an MLP neural network bag predicting roughness using the given attributes. This model was compared with a naive approach, which predicted roughness as the mean roughness value from its training set. This comparison made it possible to judge whether the MLP approach was actually generating reasonable results. The structure of the MLP-based model was the same as during the imputation phase (i.e. it consisted of a bag of 10 single hidden layer perceptrons, with 3 neurons and a hyperbolic tangent transfer function). A 10-fold cross validation procedure was applied 10 times for different permutations of the data table in order to evaluate the MLP based and the naive approach. The results are given in Table 4 and indicate that the use of neural networks results in improvements over a naive approach. These results may provide a baseline for comparison with future work connected with roughness prediction.

Industrial applications

From the industrial point of view, it is not sufficient to demonstrate that the developed soft computing model is accurate enough. Certain figures of merit or, better still, certain plots are required for the workshop to make use of the information collected in the model. The reality of a machining workshop usually fixes most of the parameters that have been

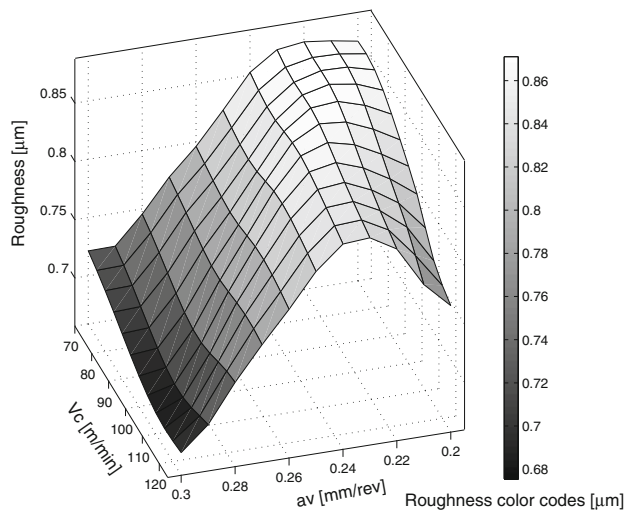


Fig. 3 Roughness prediction for different drilling parameters

considered in the soft computing model. For example, only one lubrication system (MQL or traditional coolant fluid) may be used in each milling centre. Furthermore, if the workpiece is defined by the customer, both tool and drill geometry are also fixed. In our case, the model should help the workshop technical office to prepare the Computer Aided Manufacturing (CAM) program. This program fixes tool trajectories and cutting conditions to machine the workpiece. Therefore, a plot that provides information on how the cutting conditions, specially av and Vc , affect workpiece roughness would be very useful. Thus, a figure showing predicted borehole roughness for a given tool type, diameter, hole length and lubrication system should be developed.

To do so, the third stage of the algorithm has to be performed:

- First of all, a simple model using MLP network to predict axial force is developed. The model is trained using the real data collected during experiments and provides expected axial force based on the input data: tool type, diameter, hole length, av , Vc and lubrication system.
- Then, not only different av and Vc settings together with known and fixed tool types, hole diameters and lengths and the lubrication system, but also the expected axial force are used as inputs in the bag of MLP networks. The MLP bag described above is used to predict borehole roughness.

Figure 3 shows an example obtained with this method. MQL cooling is considered for a 50 mm drill with a 10 mm diameter using tool 1. av and Vc are varied across all possible working ranges and are plotted on the X and Y axes. The Z axis shows the calculated roughness. In this case, it is easily concluded that av out of the range [0.22–0.24] and Vc in the range [90–120] will yield lower roughness. Productiv-

ity, but usually tool wear as well, increases with av and Vc values. Therefore, the final decision can be made on-site by an industry expert using this figure and taking into account other factors, tool wear being one of the most important of them. In particular, in this case a balance between low roughness, high productivity and a limited risk of quick tool wear may be expected for $av = 0.28 \frac{\text{mm}}{\text{rev}}$ and $Vc = 110 \frac{\text{m}}{\text{min}}$.

Conclusions and future works

A prediction model to optimize High-Speed deep drilling operations using both MQL and traditional working fluid has been proposed. The dataset was obtained from experiments performed on two different milling centres. The selected cutting conditions were chosen to reproduce real industrial conditions. Each tool was tested under those conditions that provide higher productivity in accordance with the lubrication system in use. In this way, a comparison of productivity and quality between the two lubrication systems could be obtained for the same set of tools. The input data include cutting parameters and axial cutting force measured by sensors on the milling centres. The output variable is the inner surface quality of the drill-hole measured by its roughness as defined by industrial standard ISO 4288:1996. Surface quality is considered an output variable due to its high industrial importance. Some of the tests were performed without properly recording all the input variables. This is a very often industrial situation, where not all the tests are useful because some of the variables are not properly acquired.

Therefore, a soft computing solution has been developed to address real life industrial needs. First of all, the algorithm using a combination of evolutionary approach and MLP networks was used to select the imputation strategy and to increase the number of measurements available for roughness modelling. Moreover, a formal method of evaluating imputation strategies in the context of roughness prediction has been proposed. Additionally, the question of whether the set of available measurements is sufficient for roughness modelling, has been assessed. Once the number of experiments was determined to be optimal, and incomplete records were imputed with the strategy selected by the algorithm, a bag of MLP networks was developed to obtain smooth roughness prediction and avoid susceptibility to noisy experimental data. Finally, the way in which roughness plots can be produced and used by domain experts to set optimal drilling parameters has been proposed. This is illustrated in a sample figure developed for process engineers working with the milling centres.

Future work will focus on the study and application of the model to other kinds of materials of industrial interest, especially aeronautical aluminium because of the huge numbers of drilling tasks involved in the manufacture of different

structural aeronautical components. Moreover, the application of this model to the optimization of slightly different industrial problems is envisaged. Among these industrial applications, the drilling of multicomponent plates for the aeronautical industry will be considered.

Acknowledgments This work has been made possible thanks to the support received from Nicolás Correa S.A. and Fundación Fatronik-Tecnalia, which provided the drilling data and performed all the experimental tests. The authors would especially like to thank Mr. Eduardo Elizburu, Mrs. Iraitx Etxeberria and Mr. Germán Rodríguez for their kind-spirited and useful advice.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Abdella, M., & Marwala, T. (2005). The use of genetic algorithms and neural networks to approximate missing data in database. *IEEE 3rd international conference on computational cybernetics* (pp. 207–212).
- Acuña, E., & Rodríguez, C. (2004). The treatment of missing values and its effect in the classifier accuracy. In *Classification clustering and data mining applications*. Berlin: Springer.
- Batista, G. E. A. P. A., & Monard, M. C. (2001). A study of K-nearest neighbour as a model-based method to treat missing data. *Proceedings of the Argentine symposium on artificial intelligence* (pp. 1–9).
- Benardos, P. G., & Vosniakos, G. (2003). Predicting surface roughness in machining: A review. *International Journal of Machine Tools and Manufacture*, 43(8), 833–844.
- Biglari, F., & Fang, X. (1995). Real-time fuzzy-logic control for maximizing the tool life of small-diameter drills. *Fuzzy Sets and Systems*, 72(1), 91–101.
- Braga, D., Diniz, A., Miranda, G., & Coppinni, N. (2002). Using a minimum quantity of lubrication and a diamond coated tool in drilling of aluminum-silicon alloys. *Journal of Materials Processing Technology*, 122, 127–138.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.
- Chandrasekaran, M., Muralidhar, M., Krishna, C. M., & Dixit, U. S. (2010). Application of soft computing techniques in machining performance prediction and optimization: A literature review. *International Journal of Advanced Manufacturing Technology*, 46(5–8), 445–464.
- Choudhary, A. K., Harding, J. A., & Tiwari, M. K. (2009). Data mining in manufacturing: a review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, 20(5), 501–521.
- Davim, J. P., Sreejith, P. S., Gomes, R., & Peixoto, C. (2006). Experimental studies on drilling of aluminium (AA1050) under dry, minimum quantity of lubricant, and flood-lubricated conditions. *Proceedings of the Institution of Mechanical Engineers, Journal of Engineering Manufacture, Part B*, 220(10), 1605–1611.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39, 1–38.
- Filipovic, A., & Stephenson, D. A. (2006). Minimum quantity lubrication (MQL) applications in automotive power-train machining. *Machining Science and Technology*, 10, 3–22.
- Gediga, G., & Düntsch, I. (2002). Maximum consistency of incomplete data via non-invasive imputation. *Artificial Intelligence Review*, 19(1), 93–107.
- Hashmi, K., Graham, I., & Mills, B. (2000). Fuzzy logic based data selection for the drilling process. *Journal of Materials Processing Technology*, 108(1), 55–61.
- Hayajneh, N. T. (2001). Hole quality in deep hole drilling. *Materials and Manufacturing Processes*, 16(2), 147–164.
- Heinemann, R., Hinduja, S., Barrow, G., & Petuelli, G. (2006). Effect of MQL on the tool life of small twist drills in deep-hole drilling. *International Journal of Machine Tools and Manufacture*, 46(1), 1–6.
- Heinemann, R., Hinduja, S., & Barrow, G. (2007). Use of process signals for tool wear progression sensing in drilling small deep holes. *International Journal of Advanced Manufacturing Technology*, 33(3–4), 243–250.
- Hu, M., Salvucci, S.M., & Cohen, M.P. (1998). Evaluation of some popular imputation algorithms. *Proceedings of the Survey Research Methods Section, American Statistical Association* (pp. 308–313).
- Jantunen, E., & Vaajoensuu, E. (2010). Self adaptive diagnosis of tool wear with a microcontroller. *Journal of Intelligent Manufacturing*, 21(2), 223–230.
- Jönsson, P., & Wohlin, C. (2006). Benchmarking k-nearest neighbour imputation with homogeneous Likert data. *Empirical Software Engineering*, 11(3), 463–489.
- Juszczak, P., & Duin, R.P.W. (2004). Combining one-class classifiers to classify missing data. *Multiple Classifier Systems* (pp. 92–101).
- Kubota, H., & Tabei, H. (1999). Drilling of a small and deep hole using a twist drill. *Transactions of the Japan Society of Mechanical Engineers, Part C*, 62(601), 3691–3697.
- Mehrabadi, I. M., Nouri, M., & Madoliat, R. (2009). Investigating chatter vibration in deep drilling, including process damping and the gyroscopic effect. *International Journal Of Machine Tools and Manufacture*, 49(12–13), 939–946.
- Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*. Berlin: Springer.
- Nandi, A. K., & Davim, J. P. (2009). A study of drilling performances with minimum quantity of lubricant using fuzzy logic rules. *Mechatronics*, 19(2), 218–232.
- Sanjay, C., Neema, M. L., & Chin, C. W. (2005). Modeling of tool wear in drilling by statistical analysis and artificial neural network. *Journal of Materials Processing Technology*, 170(3), 494–500.
- Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Boca Raton: Chapman & Hall/CRC.
- Wei, W., & Tang, Y. (2003). A generic neural network approach for filling missing data in data mining. *IEEE international conference on systems, man and cybernetics, 2003* pp. 862–867.
- Weinert, K., Inasaki, I., Sutherland, J. W., & Wakabayashi, T. (2004). Dry machining and minimum quantity lubrication. *Ann. CIRP*, 53(2), 511–537.
- Zawistowski, P., & Grzenda, M. (2009). Handling incomplete data using evolution of imputation methods. *Proceedings of 9th international conference ICANNGA 2009. Lecture notes in computer science* (Vol. 5495, pp. 22–31). Berlin: Springer-Verlag.
- Zang, J. Y., Liang, S. Y., Yao, J., Chen, J. M., & Hang, J. L. (2006). Evolutionary optimization of machining processes. *Journal of Intelligent Manufacturing*, 17(2), 203–215.
- Zhang, J. Z., & Chen, J. C. (2009). Surface roughness optimization in a drilling operation using the taguchi design method. *Materials And Manufacturing Processes*, 24(4), 459–467.