# A priori evaluation of simulation models preparation processes using artificial intelligence techniques

Florence DANGLADE, Jean-Philippe PERNOT, Philippe VERON, Lionel FINE

**Abstract**

Controlling the well-known triptych costs, quality and time during the different phases of the Product Development Process (PDP) is an everlasting challenge for the industry. Among the numerous issues that are to be addressed, the development of new methods and tools to adapt to the various needs the models used all along the PDP is certainly one of the most challenging and promising improvement area. This is particularly true for the adaptation of Computer-Aided Design (CAD) models to Computer-Aided Engineering (CAE) applications, and notably during the CAD models simplification steps. Today, even if methods and tools exist, such a preparation phase still requires a deep knowledge and a huge amount of time when considering Digital Mock-Up (DMU) composed of several hundreds of thousands of parts. Thus, being able to estimate a priori the impact of DMU adaptation scenarios on the simulation results would help identifying the best scenario right from the beginning. This paper addresses such a difficult problem and uses Artificial Intelligence (AI) techniques to learn and accurately predict behaviours from carefully selected examples. The main idea is to identify rules from these examples used as inputs of learning algorithms. Once those rules obtained, they can be used on a new case to a priori estimate the impact of a preparation process without having to perform it. To reach this objective, a method to build a representative database of examples has been developed, the right input (explanatory) and output (preparation process quality criteria) variables have been identified, then the learning model and its associated control parameters have been tuned. One challenge was to identify explanatory variables from geometrical key characteristics and data character-

izing the preparation processes. A second challenge was to build a effective learning model despite a limited number of examples. The rules linking the output variables to the input ones are obtained using AI techniques such as well-known neural networks and decision trees. The proposed approach is illustrated and validated on industrial examples in the context of Computational Fluid Dynamics simulations.

*Keywords:* Process evaluation, Digital Mock-Up preparation, artificial intelligence, machine learning, knowledge formalization.

## 1. Introduction

The Product Development Process (PDP) relies on a multitude of activities such as design, sizing, analysis, product optimization, process simulation or prototyping. Each activity is often based on an adapted Digital Mock-Up (DMU) used to model the product with more or less details. The preparation process of an original DMU to a representation adapted for a given activity is still a very challenging issue. It often requires a succession of operations which are based on different tools driven by many control parameters. Today, even if the methods and tools used to perform these operations exist, following such a preparation process strongly relies on the knowledge of the experts that is not fully formalized. This lack of formalization and the associated lack of knowledge on the performance of a given preparation process induces numerous iterations between the original model and the model prepared for an activity. Thus, being able to estimate a priori the cost and quality of a given preparation process will help optimizing the transfer between Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) models. As a consequence, the PDP will be shortened and the over-quality avoided.

Today, even if commercial software does incorporate some functionalities dedicated to the adaptation of CAD models to CAE applications, the preparation process still requires a deep knowledge and a huge amount of time when considering Digital Mock-Up (DMU) composed of several hundreds of thousands

of parts. The preparation process consists of three main steps: simplification, adaptation and meshing (Figure 1).

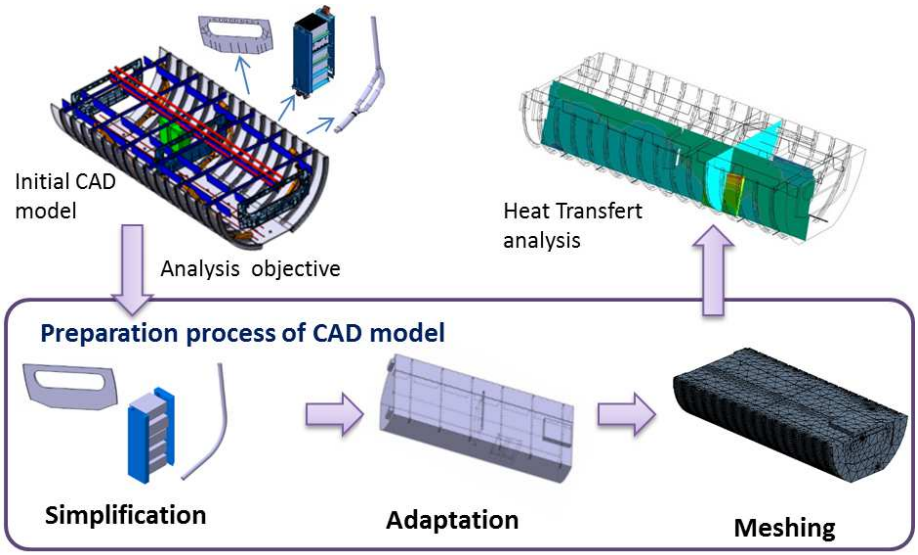


Figure 1: Main stages of cad model preparation (application to CFD analysis).

The CAD model simplification eases the meshing and simulation steps by removing items and modifying the geometry. Simplification techniques are detailed in the section 2.1. The adaptation steps consist in extracting faces for meshing and in identifying the surfaces supporting the boundary conditions. The CAD model meshing allows the numerical analysis of the problem by approximating a geometry with more or less small and complex elements (e.g. triangles, tetrahedra, hexahedra) depending on the available computing time and the expected accuracy. The preparation process can be described and modelled by a set of operations, a sequencing and a set of control parameters. For each operation, the user adjusts one or more parameters (e.g. the size of mesh elements, the level of simplification, the list of sub-assemblies to remove). Therefore, for a given simulation objective, there exists many preparation processes. Today, the sequence of operations and the associated control parameters are selected by the experts who try to minimize the impact of the adaptation on

the results while minimizing the preparation costs. Those costs are strongly correlated to the time spent by the expert on the different tasks.

They exist many tools and operations to simply a CAD model, section 2.1 presents the main simplification techniques applied to our case study. However, the criteria used to select which operations and which parameters are to be used are not fully formalized and the effects not always mastered. Section 2.2 introduces methods to evaluate the impact of a simplification on the results of an analysis. However, there is a lack of methods to a priori estimate the impact of a simplification on the quality and accuracy of a simulation.

Therefore, the aim of this work is to define a new approach to estimate a priori the quality of a preparation process. In this way, the analysts can test different adaptation strategies and thus identify the best one with respect to a given simulation objective. Of course, this does not exempt the analysts to make the numerical simulation at the end, but only one time following the preparation process considered as the best. The proposed approach is based on the use of Artificial Intelligence (AI) techniques [1] for the evaluation of preparation process quality. The quality of a preparation process could be evaluated by orders of magnitude of analysis errors, preparation duration and analysis duration. Amongst AI techniques, supervised learning techniques are able to estimate output variables from carefully selected examples without knowing rules that link input and out variables. Variables to predict can be discrete values that are divided into several classes. So, the retained AI techniques must be able to predict a discrete output variable from a set of input variables. Classifiers like Bayesian classifier, Decision Trees, Neural Networks, Support Vector Machine or RBF Networks can take on this task.

Section 2.3 gives examples on the use of AI techniques in the mechanical engineering domain . Actually, existing AI techniques are sufficient and well appropriated to our purpose. So, this paper does not aim at developing a new one but rather it aims at finding a way to model our preparation process so that it can be used by existing AI techniques. Regarding the use of these techniques, the first challenge is to identify the most determinant explanatory variables that

are extracted from CAD models and preparation processes. A second challenge is to find a good quality learning model despite a limited number of examples.

To reach these objectives, a dedicated framework has been devised (Figure 2) . First, the knowledge embedded in a set of preparation examples is stored in a set of so-called instances. Each instance contains the data able to describe the preparation process, the initial CAD models, the simplified and prepared CAD models as well as the results of the analysis. Then, those instances are implemented in a learning tool which is used to configure a classifier that can then estimate the quality of a process for a new unknown case. Each steps of this overall approach will be developed in section 3.

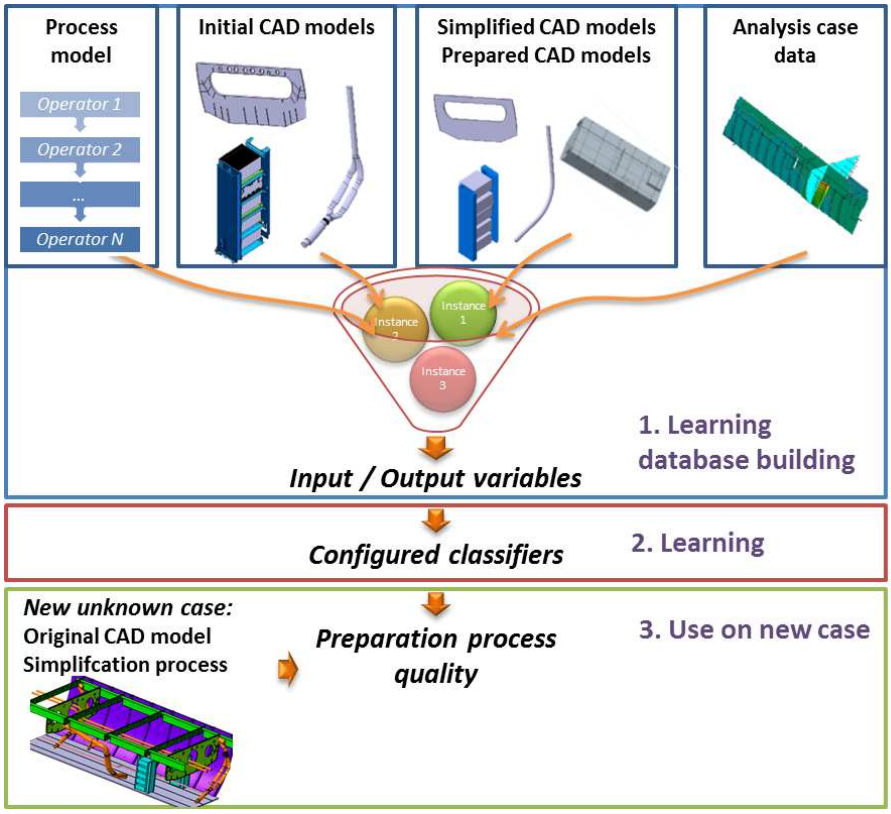


Figure 2: General approach for preparation process evaluation by using machine learning techniques.

To validate it, the proposed method has been applied to the adaptation of large CAD models for Finite Elements Analysis (FEA). However, it is not restricted to such an application and the proposed approach can be extended to the other steps of the PDP. Some experimental results are discussed in section 4.

## 2. Related works

### *2.1. CAD models simplification techniques*

There exist a huge number of techniques to simplify a geometric model according to different criteria. The purpose of this section is to identify which simplification methods are appropriate to the adaptation of complex CAD models (i.e. defined with a large number of parts and numerous features) to FEA like CFD simulation. Thakur and al. [2] have proposed a classification of simplification techniques based on surface entity operators, volume entity operators, explicit features operators or dimension reduction operators. We can add to this list, operations based on the simplification of assembly trees. Among all the simplification methods, selected techniques are described below. Figure 3 shows the results of some simplification operators on a sub-assembly.

*Part filtering.* Part filtering consists in deleting parts in an assembly. Usually, small parts far from boundary conditions are removed.

*Defeaturing.* The defeaturing step consists in removing details like holes, pockets, pads, fillets or chamfers. This method is well adapted when the native CAD models are available. Nevertheless, the cost of the operation can be very high when the building tree of the model is not available, i.e. if a neutral format like STEP is to be used. Some tools like NX SIEMENS [3] or GPURE [4] offer ready-to-use defeaturing functions based on surface entity simplification [5] (e.g. hole filling, cutting, removal of the bosses, or surface reconstruction for fillets and chamfers). These tools can remove a family of features based on their size, but other criteria such as the distance from a boundary condition is not available or request many non-automated operations.

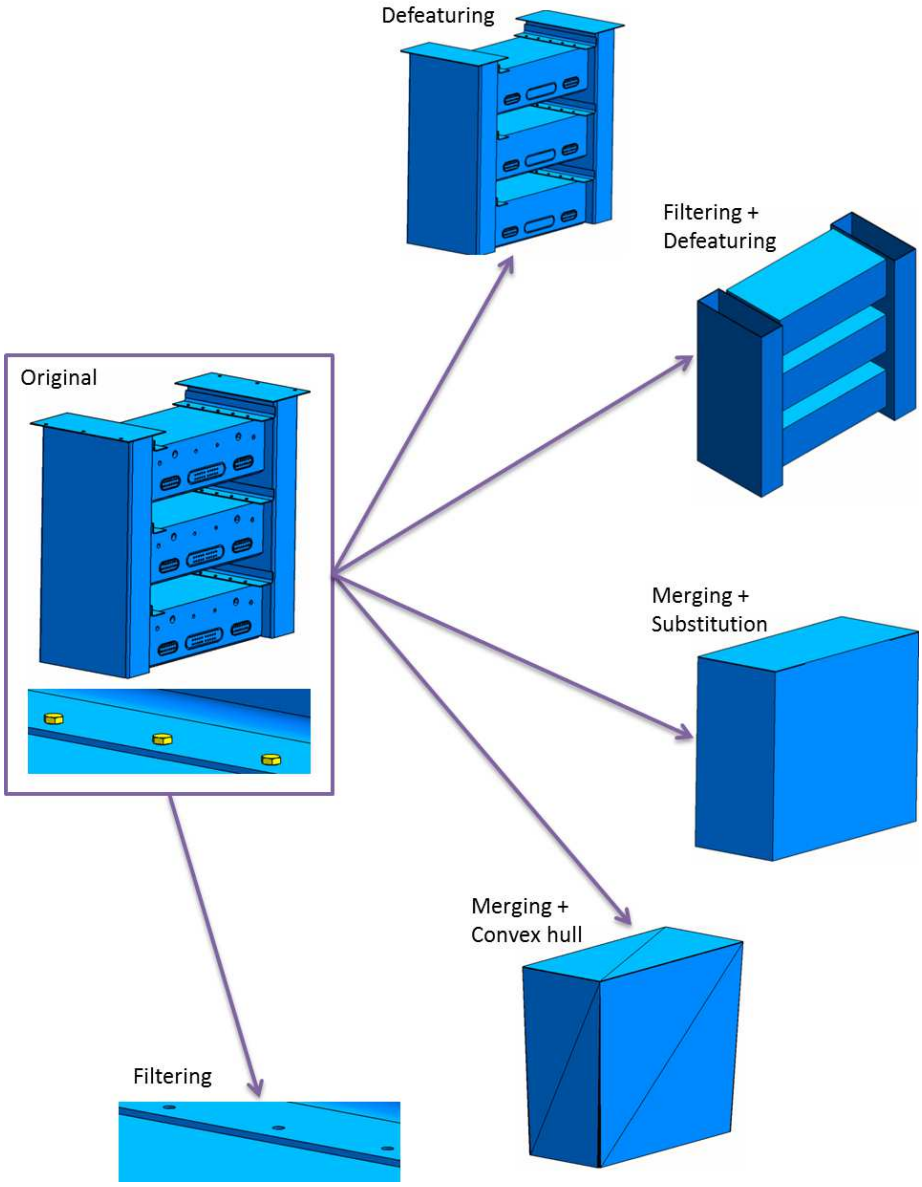


Figure 3: Examples of simplifcation operators and admissible combinations.

*Convex hull.* The creation of a convex hull keeps only outer wrapping of the model based on the smallest convex set containing the geometry [6]. The input model can be either a native CAD model, or a standard CAD file like STEP,

or a mesh. The output model is usually a polyhedral mesh. Thus, the original CAD model is discretized, the triangles outside of the envelope are then filtered. Simplification tools included in NX or GPURE provide fast modelling of convex hull. The control parameter is often the level of accuracy defined for instance by the distance between the original model and the simplified one.

*Decimation and faces clustering.* The details are simplified by decimation of edges, faces or vertices, or by faces clustering [7], [8]. Here again, the control parameter is often the accuracy between the original and simplified models. The input model can be a native CAD model, a manifold B-Rep model, a non-manifold model or a mesh. The output model is usually a polyhedral mesh. GPURE offers ready-to-use decimation operators.

*Substitution.* This operation consists in removing surfaces and/or volumes and in rebuilding them with a less complex geometry, for instance using cylinders and parallelepipeds. Unfortunately, the removal and rebuilding functions are not automated within the commercial CAD modellers. The parameters to be considered are the dimensions of the new model. Experts suggest rules to identify these dimensions (e.g. they preserve the length of the bounding box and the volume of the model).

*Merging.* This operation merges several parts into a single sub-assembly in order to ease the model handling and the wrapping of a volume. This operation reduces the risk of crashes during the meshing and simulation phases. The level of simplification is not affected a lot when this operation is used alone. This is usually implemented in addition to other simplification operations. Ready-to-use tools for merging exist in CATIAV5 [9] for instance. The experts have to select sub-assemblies to be merged and they do have to decide whether the merging is to be executed before or after other simplification operations. During the preparation of a CAD model for FEA, experts classically make use of several of these techniques in different orders. If we combine the possibilities offered by the tools, by the simplification techniques and by their control parameters, we

obtain a large number of admissible preparation processes. Actually, there exist 62 admissible sequences of simplification operations. For a sub-assembly, there is up to 300 different simplification sub-processes and it is therefore impossible to test all of them. In addition, the settings are varied (e.g. number of removed parts or details, maximum distance between the original model and the simplified one, difference in faces number). Moreover, they are not significant with respect to the impact of simplification on the analysis. Thus, it is necessary to define criteria to compare the original and simplified models regardless of the simplification process that can characterize the impact of simplification on the analysis result.

### *2.2. Techniques to estimate the simplification impact on analysis results*

When considering FEA on a large assembly model, the number of meshed elements can be so large that the adaptation and meshing steps are often impossible without simplifications. Additionally, without a high-level of simplification, meshing and simulation operations can be very time-consuming. However, simplifying a CAD model may result in a variation of the simulation results to be analysed. Thus, it is important to control how the simplification may impact the results of a simulation. Related works focus on three methods: physical behaviour approaches, subjective approaches and geometric approaches.

In recent years, physics-based approaches for the evaluation of simplification impact on analysis results have gained interest. Tang and al. [10] proposed a new index to evaluate defeaturing impact on FEA results by using the change of a model's strain energy. This method is restricted to the linear elasticity. Ferrandes and al. [11] have developed a posteriori criteria by using an approximation of the energy norm of the difference between the FEA results on the original and simplified models. The impact of simplification on global simulation results is evaluated from influence indicator of each detail. The equations carried out for calculation by convection and radiation are very different and these methods cannot be applied to heat transfer analysis. In the field of heat

transfer analysis, Gopalakrishnan [12] has proposed a theory for estimating analysis errors in case of heat transfer with a high accuracy of the estimated error. This is very efficient, but it requires accurate information about the simplified geometry. However, in the context of this work, we want to a priori estimate the impact of the simplification on the simulation results, i.e. without preparing the model and without performing the simulation. So, in our case, the simplified geometry is not available. Moreover, the above described methods focus on the defeaturing. Little attention has been paid to the impact of global simplification methods (e.g. convex hull modelling or substitution) on the simulation results.

Subjective approaches are based on knowledge and skills of analysts [13]. These methods need to know and to formalize exhaustively the criteria that influence the errors on the results of an analysis. Here, examples of simplification for which the errors are known have to be envisaged. Actually, the first set of criteria expressed by the experts are geometric criteria [14]. The differences between the reality and the analysis results are estimated from changes in volume, area or barycentric coordinates between the original and simplified models. Other numerous geometric criteria can be used (curvature, number of faces, number of features and so on). But these criteria do not give accurate indicator on the analysis errors and also require the computation of the simplified models, which is not necessary in our approach.

Finally, the use of estimation techniques does not require the computation of the simplified models. The analysis is performed a priori, i.e. on the initial CAD models and before any adaptation. Danglade and al. have introduce a technique to identify and delete the features which have a low impact on the accuracy of the results [15]. However, this method was limited to the defeaturing of a single part. In this paper, the idea is to extend this principle to all the previously introduced simplification operations and to global preparation processes of large assembly models.

### *2.3. Techniques of AI in mechanical engineering*

*Learning objective.* In mechanical engineering, AI techniques are used in various applications such as physical behaviour estimation, design , recognition , reverse engineering or material sciences. In those applications, classifiers are often used to estimate one or several output parameters of different natures (e.g. geometrical, statistical, physical), or even to classify shapes or 3D points sets. For sizing and shape design, classifiers are often estimating global geometric parameters of the model ([16],[17], [18], [19]). The estimation of a physical quantity ([20], [21], [22], [23]) like a load, a stress, a pressure or a temperature, remove the need to solve complex equations. Statistical parameters ([24], [25], [26]) don't give directly the physical quantity. However, they offer the opportunity to estimate for example a standard deviation, a mean, a trend or a physical effect probability. The classification of shapes ([27], [28]), or digitized 3D points sets, is used for models recognition and reuse. The aim of our work is to a priori estimate the impact of the CAD model simplification on the results of the analysis as well as the cost of the preparation. Thus, the idea is to be able to perform the estimation without doing the simplification itself. Here, we are not trying to estimate the analysis results but only the errors due to the simplification. Physical quantity estimation is not useful in our case. A statistical parameter (e.g. percentage of deviation) seems more appropriate in our study.

*Input variables and examples.* When using artificial intelligence techniques on CAD models, the most important challenge is to identify the input variables to be processed. Physical problem is generally described by physical quantities vectors ([21], [23], [26], [19]). Geometrical data are described by coordinates ([24], [18]), by histogram [27] or by a vector of parameters ([20],[16], [17]). In our case, the complexity of the manipulated CAD models makes it difficult to use graphs or histograms. The high-level of simplification between two configurations does not allow the use of the points' coordinates. A vector of carefully selected parameters seems to be the reasonable solution. The choice of the most representative input parameters, the selection and configuration of the classifiers

are so many issues that are addressed and developed in section 3. Moreover, in our approach we also need to identify variables which best characterize the preparation process to be evaluated. This is also a challenging issue that has been addressed in this paper.

Finally, it has to be recalled that the Machine Learning Techniques (MLT), which can be used to identify the estimation rules, often require a large number of examples, also called instances. However, being the preparation of models for numerical simulation a very long process, the number of examples will be limited. Thus, it is necessary to propose a method which guarantees the reliability of the estimations despite a limited number of examples. This has also been addressed in this paper.

## 3. Proposed framework to evaluate CAD model preparation processes

### *3.1. Overall approach*

This section aims at introducing our new approach (Figures 2 and 4) which makes use of AI techniques to a priori estimate the impact of a preparation process on the quality of a FEA. The performance of the preparation process is evaluated by means of a performance indicator that is computed from the impact of the simplification on the simulation results and from the preparation and simulation costs. Generally speaking, to build a classifier able to estimate an output variable from a set of input variables, it is necessary to determine four elements:

1. examples database which should be as representative as possible of existing cases;
2. the explanatory or input variables that are used by the classifier to estimate output variables;
3. the type of classifier and its overall architecture;
4. the classifiers parameters.

A method to set these elements is briefly introduced hereunder and is detailed in the following subsections.

First, a database of CAD model preparation process examples is built (Part 1 of Figure 2). The section 3.2 proposes a method to model a representative database by covering all range of preparation processes. The extracted data from CAD models are the ones that seem most logical according to the analysts' experience. These choices will then be validated by AI techniques. Data relative to the preparation process are also extracted to set up the database and from the inputs.

Relevant explanatory variables for the estimation of the output variables are selected and treated in a learning database. Those variables are extracted from the prepared CAD models and available simulation results. Finally, the learning data are compiled in a matrix where rows describe input or output variables and lines match to examples of CAD model preparation processes (Figure 4). The selection of variables are detailed in section 3.2.2. For a new case, available variables are limited to the original CAD model data, to the boundary condition information and to the preparation process description. Some unknown data (simplified model and prepared model) are needed for the output variables estimation. These intermediate variables must be estimated before the output variables.

During the learning phase (Part2 of Figure 2), AI techniques are used to select, configure and optimize classifiers able to estimate the intermediate and output variables from a set of input variables.

We will name "learning model" a combination of configured classifier and variables. The objective of this part is to identify, for each variable to predict, the best learning model as illustrated in the figure 5. For this purpose, it is likely to use four stages [29]: machine learning initialization (1), resampling and distribution (2), optimization (3) and quality of learning models evaluation

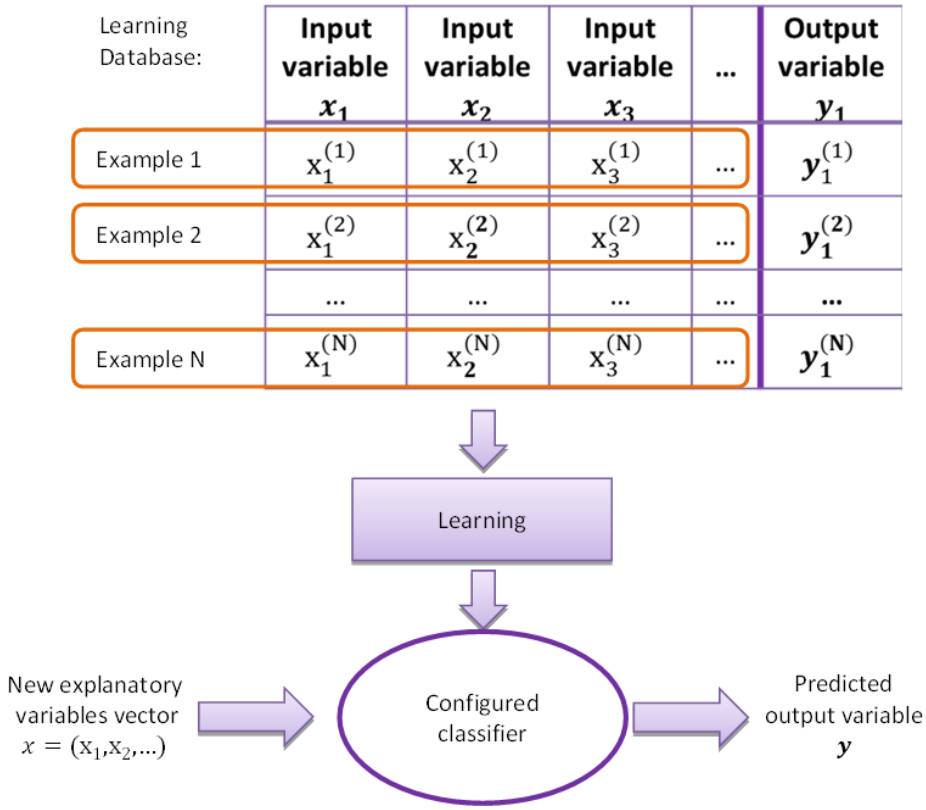


Figure 4: Learning model (classifier and learning variables) for the estimation of an output variable y from an input variables vector x.

(4). In stage 1 AI techniques are pre-selected, an architecture is proposed for candidate classifiers. In stage 2, examples are partitioned, output variables are distributed in several classes and the selection of explanatory variables is refine. In stage 3, best classifiers are selected for each output variables and are then optimized. In stage 4, the evaluation of the learning models allows to identify the best configuration of variables and classifier.

The method developed for learning is given in section 3.3.

For a new case (Part 3 of Figure 2), available variables are extracted. Then, classifiers are used to estimate intermediate and output variables. In order to

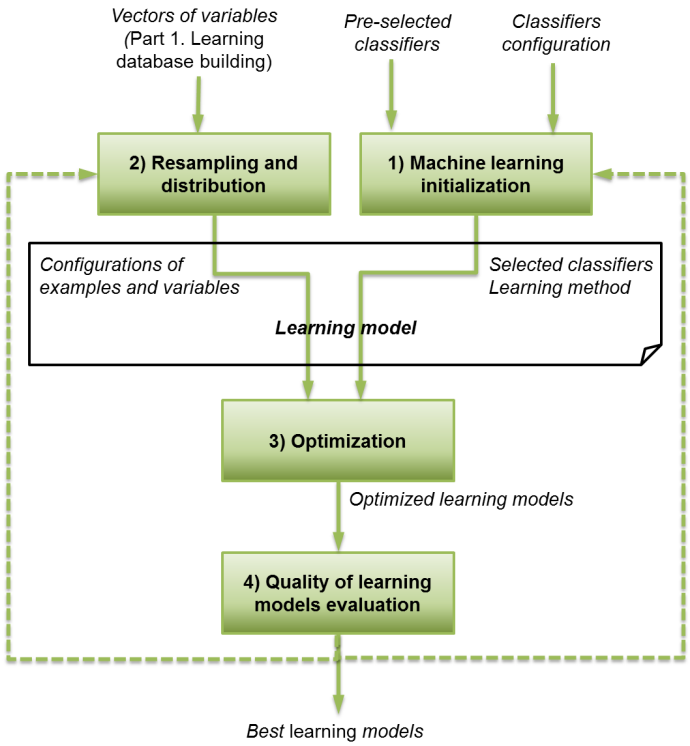


Figure 5: Part 2 : General method for Learning.

evaluate the impact of the tested preparation process, experts finally analyze the estimated costs and the analysis result error. The best preparation process is identified by analyzing performance indicators among a large set of tested processes. The workflow for evaluating processes on a new case is described in section 3.4.

Eventually, new examples can be added to the learning database once their impact on the simulation has been estimated. This refers to the capitalization of the knowledge embedded in the examples.

In this paper, the proposed method will be validated while considering the preparation of CAD models for CFD simulations. Input (explanatory) variables

X are parameters that are extracted from the original and prepared CAD models, from the simulation model and from the preparation process description. These input variables are described in the next section. Output variables $y$ are parameters that characterize the quality of the preparation process to be evaluated. Five output variables will be estimated by learning: the impact of the simplification on a sub-assembly (y1), the simplification cost of a sub-assembly (y2), the cost of preparation (y3), the cost of analysis (y4) and the analysis result error (y5).

### *3.2. Part 1. Learning Data Base Building*

#### *3.2.1. Learning database*

*Examples modeling.* The learning database must contain a significant number of examples that must be evenly distributed in each output variable class. Actually, all ranges of Level Of Simplification (LOS) and Analysis Result Error (ARE) must be covered. The LOS is defined by means of the Hausdorff distance between the less simplified model and the tested model. The $ARE$ (Eq. 1) is the error between the analysis result of the tested model $R(M^m)$ and the analysis result of the reference model $R(M^m_{ref})$.

$$ARE(M^m) = \frac{R(M^m) - R(M^m_{ref})}{R(M^m_{ref})} \tag{1}$$

A CAD model $M^m_i$ is made of sub-assemblies $C^n_j$. $m$ index is the reference of the global CAD model, $n$ is the reference of the sub-assembly, $i$ is the reference of the global model preparation, $j$ is the reference of the simplification sub-process. The learning database contains specific models ($M_0$, $M_1$, $M_{ref}$) having high, intermediate or low LOS as well as models which cannot be simulated. $M_0$ is the model without simplification. $M_1$ is the model with the highest LOS (all sub-assemblies are substituted by parallelepipeds or cylinders). $M_{ref}$ is the less simplified model that can be simulated. Models with high LOS are simplified to the maximum except one. Models with low LOS are built from the $M_{ref}$

by removing or simplifying one sub-assembly. Sub-processes of sub-assemblies simplification are ranked according to their impact on the analysis results. Examples with intermediate LOS are built by simplifying and/or removing one by one sub-assemblies from $M_{ref}$ according to their rank.

*Raw Data extraction .* The choice of the explanatory variables is strongly linked to the purpose of the preparation process to be evaluated. Said differently, the variables which affect the result of a CFD simulation can be different from the variables which impact a heat diffusion simulation. As a consequence, to be sure that the learning phase will capture the best explanatory variables for a given preparation objective, the idea is to try to be exhaustive when considering the input variables. Then, selection methods will be implemented to identify the most determinant variables objective by objective. These methods are described in section 3.2.2. To build the learning database, raw data are extracted from CAD models, preparation process description and simulation information (Figure 6).

The so-called extracted data are the output and input variables which characterize the examples to be used during the learning phase. When considering a new unknown example, the extracted data only concern the input variables that are then processed by a configured classifier which estimates the values of the output variables $\{y1, , y5\}$. Input data are explanatory variables that describe the simplification process and that characterizes criteria for evaluating this process (model geometry, original and simplified models comparison, simulation information). The explanatory variables database should be as complete as possible in order to best characterize a given example. It is important to underline that the learning technique only sees the examples by means of those numerical values. The learning technique will never work on a CAD model nor a simplification process directly but rather on a set of values characterizing them.

*Simplification process description.* The simplification process of a sub-assembly is described using parameters that specify which operators are used, their pa-

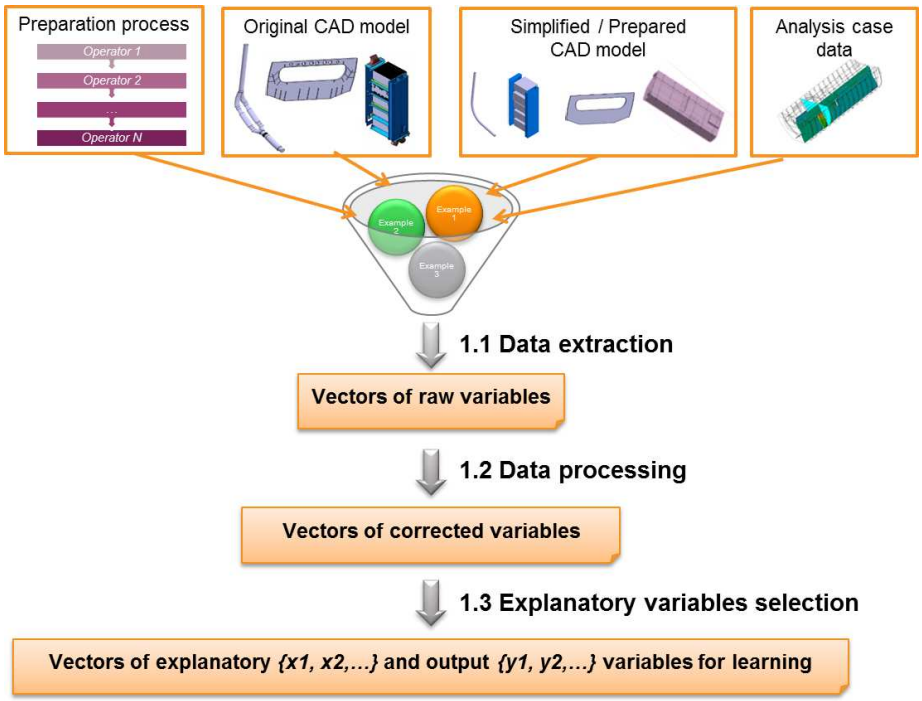


Figure 6: Part 1 Learning database building.

rameters and the adopted tools. This description formalizes the six simplifying operations that have been identified in section 2.1. It is reported in table 1 which is filled as follows:

- Defeaturing parameters are, for each type of details (e.g. bumps, pockets, holes, or rounds), numerical variables that give the relative size of the removed details and their distance to the nearest boundary condition. Details smaller than $x * CS$ are removed, being $CS$ the sub-assembly size obtained from the average length of the sub-assembly bounding-box and $0 < x < 1$. Details for which the distance to the nearest boundary conditions exceeds $x * MS$ are deleted, being $MS$ the global assembly size.

- Filtering parameters give, for parts that are candidate to the removal, conditions on size and distance from the nearest boundary condition.

- Merging, building by substitution, decimation and convex hull operators are described by nominal variables that specify if and how the sub-assembly is simplified.

Some examples of these parameters are given in Table 1 for the simplification operators illustrated in Figure 3. Variables that describe simplification processes are known for a new case.

| **Simplification examples** | **Defeaturing** | | | | **Filtering** | **Merging** | **Substitution** | **Convexe hull** | **Decimation** |
|---|---|---|---|---|---|---|---|---|---|
| | Bump | Pocket | Hole | Round | | | | | |
| **(a) Original** | 0(1)/0(2) | 0/0 | 0/0 | 0/0 | 0/0 | No | No | No | No |
| **(b) Small holes defeaturing** | 0/0 | 0/0 | 0.2/1 | 0/0 | 0/1 | No | No | No | No |
| **(c) All small details filtering** | 0/0 | 0/0 | 0/0 | 0/0 | 0.05/0.1 | No | No | No | No |
| **(d) Small parts filtering + all small details defeaturing** | 0.2/1 | 0.2/1 | 0.2/1 | 0.2/1 | 0.2/1 | No | No | No | No |
| **(e) Merging + substitution** | 0/0 | 0/0 | 0/0 | 0/0 | 0/0 | Before | Yes | No | No |
| **(f) Merging + convex hull** | 0/0 | 0/0 | 0/0 | 0/0 | 0/1 | Before | No | Yes | No |

Table 1: Examples of simplification process description on sub-assemblies. (1) relative size x of removed details. (2) relative distance x between removed details and the nearest boundary condition.

To be generic, the modelling of the preparation process should not depend on the number of parts and sub-assemblies. Moreover, the process will be described by a vector of six variables $\{x_1, , x_6\}$ indicating the overall simplification level for each type of operation (Table 2) and computed from the area $Area(P^p)$ of parts, the area $Area(F^f)$ of features, the area $Area(C_j^n)$ of sub-assemblies and the area $Area(M_0)$ of the overall model.

| Part filtering | Defeaturing | Substitution | Merging | Convexe Hull | Decimation |
|---|---|---|---|---|---|
| $\frac{\sum_{p=1}^{P} Area(P^p)}{Area(M_0)}$ | $\frac{\sum_{f=1}^{F} Area(F^f)}{Area(M_0)}$ | $\frac{\sum_{n=1}^{N} Area(C_j^n)}{Area(M_0)}$ | | | |

Table 2: Overall parameters for simplification process description.

*CAD and meshed models description.* The variables describing the CAD models (original and simplified), the adapted models and meshes are are based on

geometric quantities characterizing the size (e.g. area, volume, volume of the bounding box, number of parts) and the shape (e.g. compactness, curvatures, number of faces, number of details, number of mesh elements). There exist different ways to describe these characteristics. The values can be raw (without treatment) but it can also be a mean value (calculated from values of each parts or details), a maximal value, a dimensionless value or a value treated by normalization. So, CAD and meshed models are described by a great number of variables described according to different ways. For a new case, the only known variables are those that characterize the original models.

*Original and simplified models comparison.* The comparison of an original and a simplified model is a mean to evaluate the impact of simplification [30]. To do so, the similarity between models can be measured by computing the Minkowski distance, the Hausdorff distance or a correlation index. Another method is to compute differences between the original and simplified models while considering geometric criteria like volume, area, compactness, curvature, number of faces, number of features and so on. These differences are expressed by benefits (Eq. 2) between $M_0$ the original CAD model and $M_i$ the simplified one. Of course, for the a priori estimation, those distances are not known for a new case.

$$Benefit_i = \frac{Characteristic(M_i) - Characteristic(M_0)}{Characteristic(M_0)} \tag{2}$$

*Influence factors on analysis.* Data extracted from the simulation refer to the factors of the preparation process influencing the analysis. These factors quantify the geometrical changes due to simplification. They take into account the distances and positions of the simplified components relatively to the boundary conditions or analysis target zones. In order to take into account the size of the different parts of a component, moments have been proposed. This moment (Eq. 3) is determined from the distance $BCD(C_j^n)$ between each sub-assembly $C_j^n$ and its nearest boundary condition and the area $Area(C_j^n)$ of the sub-assembly.

$$Moment(M_i) = \sum_{n=1}^{N} \left( BCD(C_j^n)^2 . Area(C_j^n) \right) \quad (3)$$

At the end, the database contains more than 250 explanatory variables $x_v$. Other factors could be added for other preparation goals (position of gravitys center, moments ...). Thus, the proposed methodology has to ensure the completeness of the variables. For a new case, the number of known variables is limited. The unknown explanatory variables will be called intermediate variables. They must be computed first, to be able to estimate the main output variables $\{y1, ..., y5\}$.

*Data processing.* Input ($x_{raw}$) and output ($y_{raw}$) raw data are represented in matrix (Figure 4) in order to implement them in the classifiers and to provide a single representation regardless of the number of sub-assemblies and parts of the model. Each row matches to an example of a simplified sub-assembly or global model. Each column is a variable that describes the preparation process of the model or that characterizes an evaluation criterion. Before aggregating raw data in the learning database, they must be consolidated. Aberrant or missing values, which are due to lack of entry or computation errors, are deleted or replaced with exact values if they are known. This treatment increases the confidence indicators of the classifiers from about 3 to 7%.

At the end, the database contains output variables $y_{base} = \{y1, ..., y5\}$ and a set of vector of input variables $x_{base}$. Selected input variables are listed in Table 4.

*3.2.2. Explanatory variables selection*

*Selection method .* Since the most important factors are not known at the beginning of the analysis, a quite exhaustive set of explanatory variables has been proposed. Actually, more than 250 explanatory variables are used to characterize each example of the database. The selection of variables ensures the quality of the classification and helps to formalize knowledge. The proposed method

first removes correlated variables and selects the most relevant variables using well-known selection algorithms. This algorithm is depicted on Figure 7.

*Correlated variables removing.* Once the data processed (aberrant values removing, normalization, and discretization) and the groups of correlated variables $Groups(x_{cor})$ identified, a correlation coefficient is computed between each correlated variable and the variables to estimate. The less correlated variables $x_{\overline{(cor/y)}}$ with the variables to estimate are removed from the vectors of the base $x_{base}$.

*Relevant explanatory variables selection.* For each variable to estimate y, the explanatory variables $x_{base}$ are classified according to their influence on the variable $y$. Relevant explanatory variables $x_{exp}$ are selected by a stepwise backward, or forward, regression algorithm. This consists in eliminating (if backward) or adding (if forward) one by one a relevant variable according to its rank ($Rank[x_{base}(y)]$). Models are evaluated by the average quadratic error $AQE(x^q_{exp})$ (Eq. 4), where $q$ is the total number of initial variables in the base, $y^n$ is the actual variable for example $n$ and $p^n$ is the estimated variable that is given by the selected classifier. Variables are removed or added to the initial $q$ variables models giving a $q$ variables model. The operation is repeated until the $q$ variables model is not better than the $q$ variables model. When the evaluation criteria have reached an acceptable threshold and when this criteria no longer changes, then explanatory variables are correctly selected. So the key variables that were not initially known, are just identified. Otherwise, if the evaluation criteria have not reached an acceptable threshold and no longer changes, the completeness of the explanatory variables is not achieved. It will be necessary to identify new input variables.

$$AQE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y^n - p^n)^2} \tag{4}$$

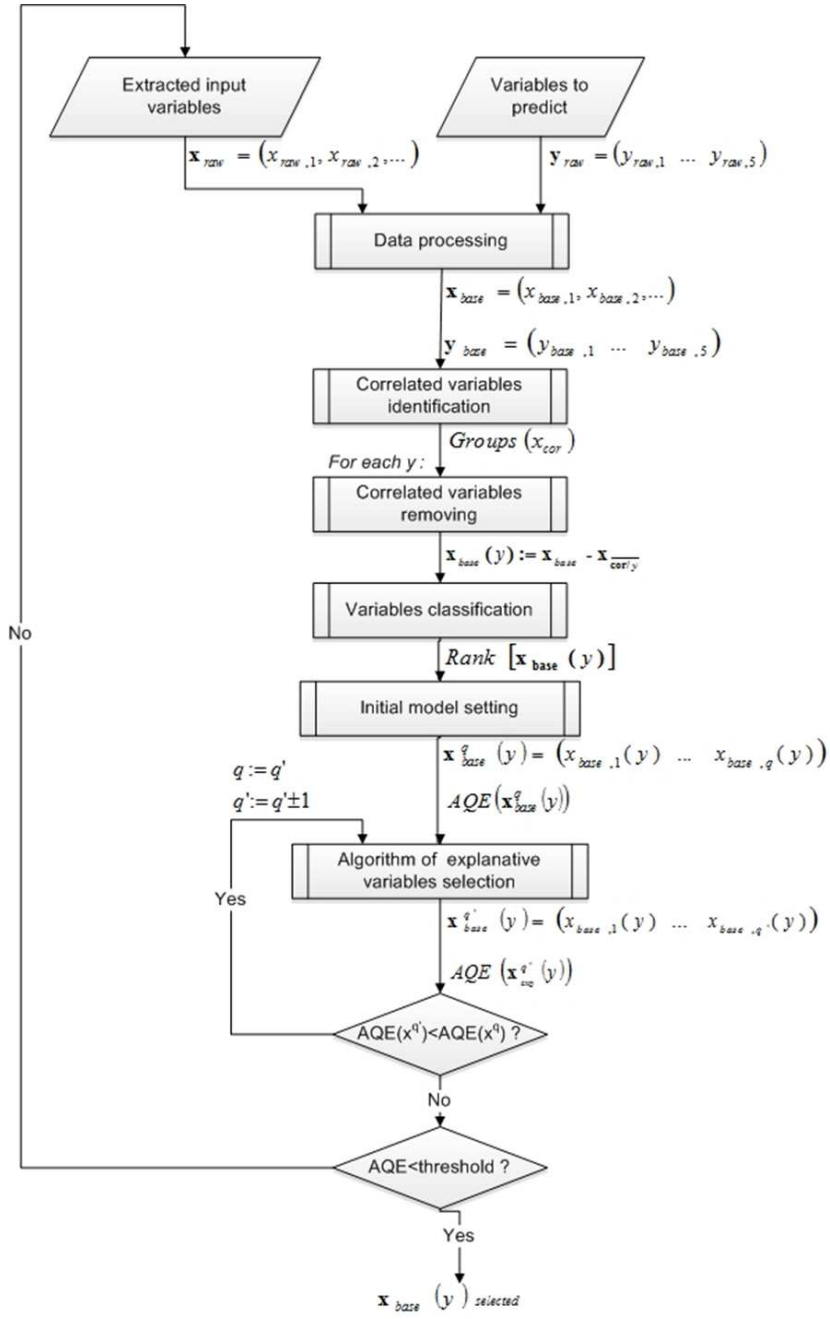


Figure 7: Method for relevant explanatory variables selection.

### *3.2.3. Selected variables*

As explained in the previous section, the most relevant explanatory variables are selected by removing the correlated variables and by ranking them according to the impact of each variable on the outputs (i.e. the result error, the cost of preparation and the cost of analysis in the present case). This algorithm has been applied to CFD analysis context and the selected variables are listed In Table 4. Naturally, in a different context (e.g. heat transfer simulation or linear elasticity), the most relevant explanatory variables obtained by the proposed algorithm can be different.

| **Selected explanatory variables** | **Estimation of :**<br>**y2= simplification cost**<br>**y3= preparation cost**<br>**y4 = analysis cost** | **Estimation of :**<br>**y1 = simplification impact**<br>**y5 = analysis result error** |
|---|---|---|
| **Process description** | All variables(1) | All variables(1) |
| **Models description** | Model area(1),(2),<br>Triangles numbers (1),(2),<br>Faces number(1),(2),<br>Ratio model / bounding box volume(1),(2),<br>Curvature(2). | Part number(1),<br>Compactness(1),<br>Ratio model / bounding box volume(1), (2),<br>Curvature(2),<br>Model area (2). |
| **Models comparison** | Area(2),<br>Volume(2)<br>Part number(2)<br>Ratio model / bounding box volume(2),<br>Curvature(2). | Area(2),<br>Volume(2),<br>Compactness(2). |
| **Influence factors with simulation data** | Moment area and boundary condition (BC) distance(2)<br>distances to BC and target(1). | None |

Table 3: Selected explanatory variables when considering cfd simulations. (1) known for a new case. (2)unknown for a new case (intermediate variables to estimate).

## *3.3. Part 2. Learning*

The learning step consists in selecting, configuring and evaluating classifiers to estimate each intermediate variable and each output variable for a given set of input invariables. The selection and configuration of classifiers are conducted simultaneously with variables configuration. The final result is a learning model (i.e. a pair $\{x(y); Cl(y)\}$) of optimized variables and classifiers. The method proposed to build the learning models is illustrated in Figures 5 and 8.

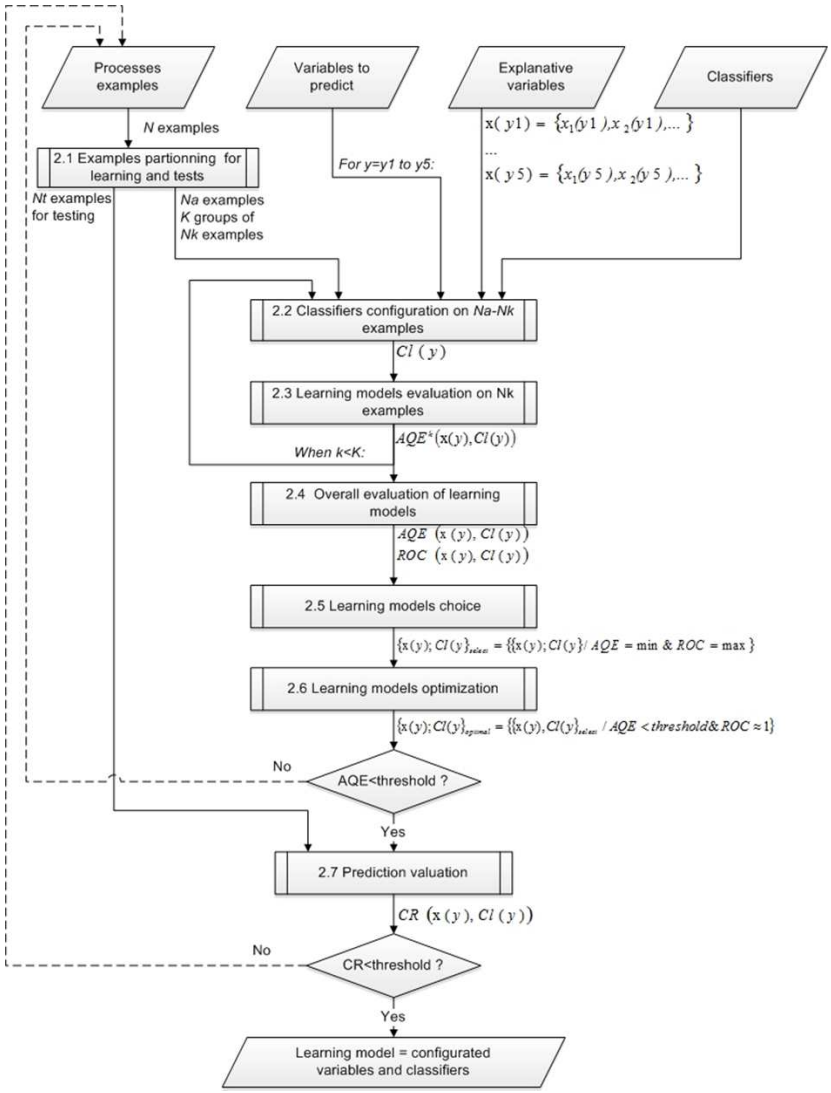


Figure 8: Part 2 : Learning. Method for the choice and the configuration of learning models.

### *3.3.1. Stage 1. Machine learning initialization*

Machine learning initialization consists in configuring selected classifiers and general learning parameters.

*Choice of AI techniques.* In order to estimate the quality of a preparation process, output variables to predict can be statistical parameters or physical quan-

tities, input variables will be vectors of parameters. Values of variables can be discrete or continuous . Two learning methods are considered: regression if output variables are continuous or classification if output variables are discrete.

According to the section 2.3, the main techniques used in mechanical engineering for these objectives and variables are neural network ([24], [25], [26], [29]). Other techniques that can predict a discrete or continuous output variable will be explored, like Decision Trees, Support Vector Machines or Naive Bayes Functions.

*Configuration of classifiers.* The objective of this step is to define the architecture and the set of parameters that characterize the pre-selected classifiers and learning. During learning initialization, general parameters are chosen to define the architecture of classifiers and learning options. These general parameters are listed in the second row of table 4. During training, set of parameters are optimized in order to built classifiers. They are listed in the third row of table 4.

*Learning Initialization.* Learning initialization consists in defining methods to obtain all classifiers parameters. The parameters that define the classifier architecture (second row of the table 4) are chosen by trial-error series. Learning consists in optimizing classifiers parameters by minimizing a cost function. For that, it is necessary to choose a learning method like Back propagation, Levenberg-Marquardt algorithm, Gradient Descent,and so on.

The used cost function for an output variable $y$ and a predicted output variable $\hat{y}$ is a squared-error cost function $J(y, \hat{y})$ (equation 3.3.1.

$$J(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 \tag{5}$$

### *3.3.2. Stage 2. Resampling and distribution*

This stage consists in refining examples and variables, while taking into account the low number of examples and the high number of explanatory variables.

*Partitioning of examples.* Examples of preparation processes are partitioned into two main groups of $Na$ and $Nt$ examples. $Nt$ examples of the second group will be used for classification tests.

The preparation of a model for the simulation is a time-consuming operation. It is difficult to obtain a very large number of examples in an industrial context. The number of available examples is limited. There is therefore a significant risk of over-learning. In order to reduce errors during learning, it has already been proposed to reduce the number of explanatory variables (section 3.2.2). The k-fold cross validation method also reduces errors during learning.

For that, $Na$ examples of the first group will be sampled in K groups for learning and validation. In order to ensure the reliability of estimations quality despite the small number of examples, it will firstly be necessary to ensure that each class of variables contains a minimal number of examples (generally 10), then the k-fold cross-validation method is carried out. This consists of first building a classifier $Cl(y)$ to estimate an output variable y on all groups of $Nk$ examples.

The parameters to be defined in this step are the percentage of examples that are reserved for learning and for tests, the number of folds and the number of repetitions.

*Output variables distribution.* Continuous output variables are to be distributed in between 3 to 13 classes. The different classes can be defined using rules. For example, the values of the variable $y5$ "analysis result error" can be divided into 7 classes as defined in Table 5. The first class corresponds to cases for which the error on the analysis is negligible and the last class to cases for which the analysis is not possible.

The parameters to be defined in this step are, for each output variable, the number of classes, the interval data in each class and the minimum number of examples in each classes.

| **Classes** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|
| **y5** ARE | $0\% < y5 \leq 1.5\%$ Low error | $1.5\% < y5 \leq 4\%$ | $4\% < y56\%$ | $6\% < y5 \leq 9\%$ | $9\% < y5 \leq 15\%$ | $y5 > 15\%$ Important error | Failed analysis |

Table 5: Example of class distribution for the analysis result error $y5$ (Analysis Result Error).

*Use of intermediate and selected variables.* A selection method of explanatory variables has been proposed in section 3.2.2. For each output variable to predict, input variables can known (KV) or unknown for a new case (variables depending on the prepared CAD model). Unknown variables must be estimated by learning before to predict output variable. So, they are called intermediate variables (IV). Several scenarios can be considered. IV can be used (scenarios A on table 6) or not (scenarios B) for learning. All potential explanatory variables can be exploited, or only more sensitive variables. Table 6 summarizes the scenarios to study.

| | | Use of IV | | |
|---|---|---|---|---|
| | | All IV | More sensitive IV | None IV |
| Use of KV | All KV | A11 | A12 | B1 |
| | More sensitive KV | | A22 | B2 |

Table 6: Scenarios depending on the use of intermediate variables (IV) and selected variables for IV and known variables (KV).

The retained scenario is for each variable to predict the one that gives the best classification for a minimum number of intermediate variables according to the criteria given in section 3.3.4.

### *3.3.3. Stage 3. Optimal architecture*

The figure 8 shows the proposed approach for learning on each output variable to be predicted.

During the initialization (3.3.1) and resampling (3.3.2) stages, the selected classifiers have been configured (step 2.2 on figure 8) for a given architecture (second row in table 4), the examples have been partitioned (step 2.1 on figure 8) In k groups of Nk examples.

*Selection of classifiers.* The classifiers are built and tested on k groups (step 2.3 on figure 8) and then, on the whole learning model (step 2.4 on figure 8). Variable configurations are identical for all tested classifiers (same choice of explanatory variables, and same processing for the explanatory variables and the output variable).

The classifiers are evaluated based on the the Average Quadratic Error ($AQE$, Eq. 4) and on the mean area under the ROC curve. The $AQE$ should be as minimal as possible. The ROC curve gives the true-positive rate against the false positive rate for several thresholds. It is desirable to have a mean value of area under the ROC curve close to one.

The selected classifiers (step 2.5 on figure 8) are those that obtain the best scores according to these criteria.

*Optimization of learning models.* Learning process allows to determinate the parameters of the classifiers by minimizing a cost function.

Step 2.6 consists in improving the learning model by refining classifiers and variables. For that, steps 2.2 to 2.4 are repeated for different architectures of classifiers by adjusting parameters that are listed in the second row of the table 4. In a similar way, different configurations of variables are tested for different distributions of output variables and different scenarios by trying to find a compromise between a maximum number of classes and a minimum $AQE$.

Then classifiers parameters are optimized by implementing genetics algorithms or meta-classifiers [31] like Bagging, Boosting, Stacking or combination of 2-Classes classifiers.

When the $AQE$ threshold value cannot be reached for any model, input variables are not sufficiently relevant. It will be necessary to repeat the selection phase of the input variables and to suggest new one as discussed in section 3.2.2.

The criteria for the selection of a learning model are the $AQE$ value, the value under the Receiver Operating Characteristic (ROC) curve, the percentage of correct estimations, the number of classes of the values for each variable and

the duration of the learning. A high number of classes in which the values of variables are distributed give more accurate estimated values. However, this requires a large number of examples. Since the number of preparation process examples is limited for a given objective, a compromise has to be found between the accuracy of the estimated values and the quality of the estimations.

*3.3.4. Stage 4. Quality of models*

At the end (step 2.7 on figure 8), the set of learning models is evaluated using a confidence rate (percentage of acceptable and unacceptable misclassified cases) using a group of $Nt$ examples that has been reserved for testing. An misclassified case will be considered as unacceptable when it is optimistic about the actual value.

*3.4. Part 3. Use on a new case*

Once configured, the classifiers can be used to estimate all intermediate and output variables from a new set of input variables. The known input variables are extracted and processed from new case data (i.e. original CAD data, preparation process description and simulation case description). First (step 3.1 of Figure 9), the intermediate variables $IV_j^n$ related to sub-assemblies are estimated (estimation of original and CAD model comparison, estimation of influence factors on analysis). Then the impact of simplification on analysis ($y1 = SI$) and of simplification cost ($y2 = SC$) are evaluated for each sub-assembly and for a maximum of simplification processes (steps 3.2 and 3.3). Intermediate variables $IV_i^m$ related to the overall assembly are estimated. All of them are used to estimate the preparation ($y3 = PC$) and simulation costs ($y4 = AC$). Then, the error on the result of the analysis ($y5 = ARE$) can be estimated. Finally, experts take a decision on the relevance of the proposed process by making a trade-off between the costs and analysis result errors. Of course, we could imagine to extend the proposed approach while applying Machine Learning Techniques to understand how the experts take the final decision

using $y3$, $y4$ and $y5$. This is part of a future work.

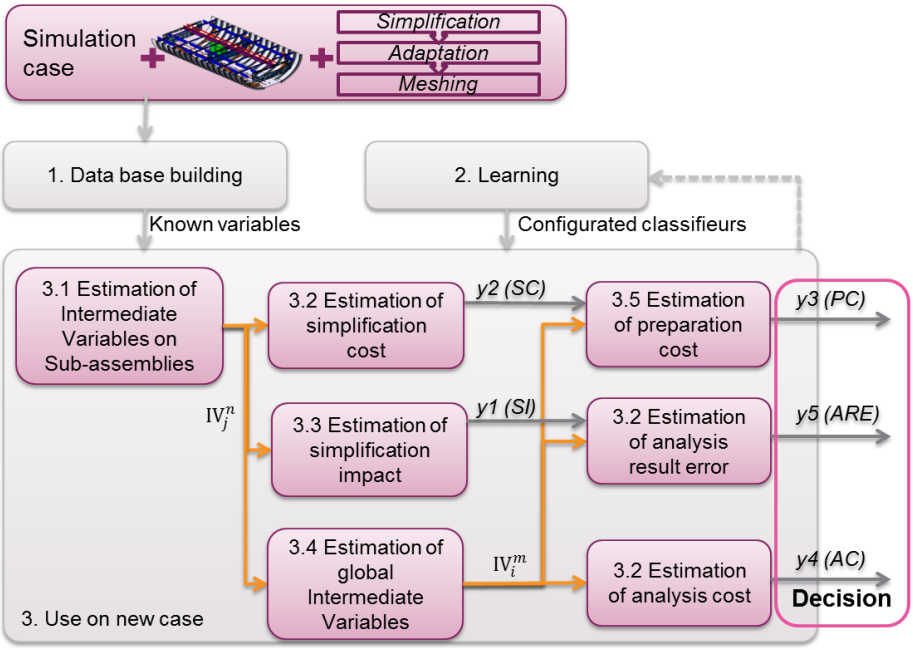


Figure 9: Part 3 : Use on a new case. A priori estimation of simplification cost (SC), preparation cost (PC), analysis cost (AC), simplification impact on sub-assembly (SI) and overall analysis result error (ARE) when considering new inputs for which the simplification and simulation have not been performed.

## 4. Application to the a priori evaluation of preparation processes of complex products

### *4.1. Application context*

The proposed method to a priori estimate the quality of preparation processes has been applied and validated on the preparation of CFD simulation model of products made of hundreds of parts. Four different products were used to build the database. They have been simplified using CATIA V5, NX Siemens and GPure. 325 examples of preparation processes were built and simulated from 4 original models and one preparation objective. The original model

contains 478 parts. In order to compare the results of the estimations with the actual values and validate the proposed approach, the cases dedicated to testing have also been prepared and simulated. Then results had been validated by an engineer. Figure 10 give examples of the overall simplified models.

For CFD analyses, the adaptation step consists in closing the geometry of the fluid volume and in modeling inlets and outlets. The CAE model is a meshed volume of the fluid, which is limited by the boundaries of the CAD model. It is important to stress that for one product, meshing characteristics were similar to all examples (e.g. tetrahedral or hexahedral mesh elements, map of sizes, boundary layer definition). All analysis data (e.g. materials characteristics, temperatures, heat flow, velocities) were also similar.

The Weka [32] software has been used to visualize the data, to process the data, to identify the relevant variables, to configure and to select the classifiers (neural networks, support vector machines, decision trees, and Bayesian Naives classifiers).

### *4.2. Results*

Different learning models were tested for the different factors that have been defined in sections 3.3.1 and 3.3.2 according to the proposed procedure in section 3.3.3.

#### *4.2.1. Learning initialization*

To limit the risk of over-learning, 75% of examples are used for the learning phase; other 25% are used for the testing phase.

For K-fold cross validation method, examples for learning had been divided in 10 sets. The used cost function is the squared-error test function (equation 3.3.1). For neural networks, the back-propagation method has been used.

The pre-selected classifiers were evaluated with the same learning and the variables configurations. According to the configuration of variables, only more

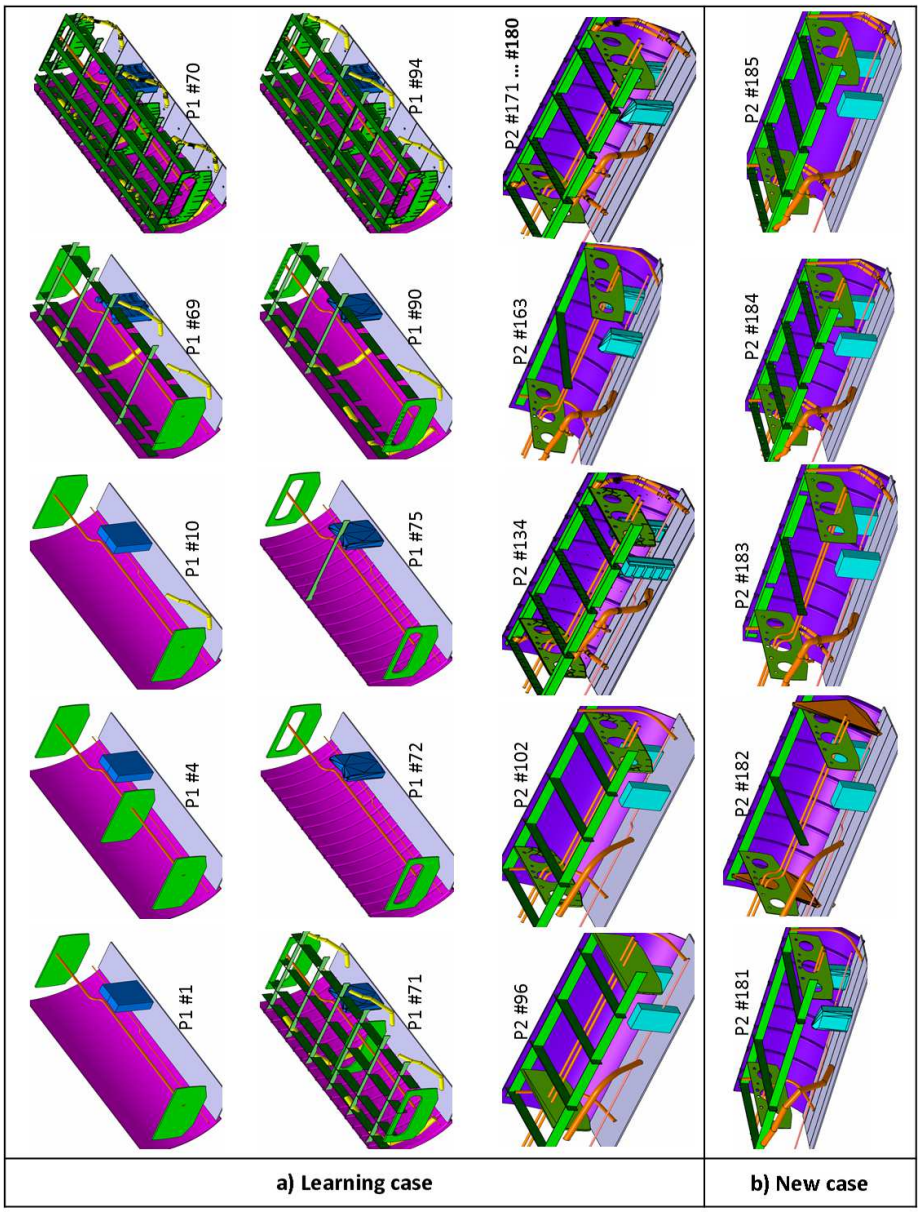


Figure 10: Examples of learning case (15 among the 244 learning cases) and new case (5 among the 80 new cases) of simplification on overall products for two products (P1 and P2).

sensitive variables had been used for learning. Output variables had been distributed in five classes.

*4.2.2. Selection of classifiers*

Table 7 gives the Average Quadratic Errors (AQE equation 4) of best classifiers for each type of pre-selected IA techniques before optimization.

| **Classifier** | **Y1 (IS)** | **Y2 (CS)** | **Y3 (CP)** | **Y4 (CA)** | **Y5 (ARE)** |
|---|---|---|---|---|---|
| **Neural networks** | 0,20 | 0,19 | 0,30 | 0,26 | 0,31 |
| **Support Vector Machine** | 0,32 | 0,33 | 0,33 | 0,32 | 0,34 |
| **Decision tree** | 0,20 | 0,18 | 0,29 | 0,22 | 0,27 |
| **Naives Bayes function** | 0,42 | 0,27 | 0,37 | 0,34 | 0,35 |

Table 7: Average Quadratic Errors for best classifiers for the five output variables prediction.

Best results have been obtained with C.4.5 decision trees for the estimation of the simplification costs ($y2$), preparation costs ($y3$) and analysis costs ($y4$), and multilayer perceptron neural networks classifiers for the estimation of the simplification impact ($y1$)and the analysis results errors ($y5$).

*4.2.3. Selection and use of input variables*

More sensitive variables had been selected with the method of section 3.2.2. The list of selected variables is given in section 3.2.3.

During the stage of classifiers selection, only more sensitive explanatory variables had been used (scenario A12 presented in the section 3.2.2). For each output variable, a series of tests allowed to identify the best scenario for using the explanatory variables among all scenarios (table 6). The table 8 gives AQE and their evolution in the different scenarios for the selected classifiers.

| | **Y1 (IS)** | **Y2 (CS)** | **Y3 (CP)** | **Y4 (CA)** | **Y5 (ARE)** |
|---|---|---|---|---|---|
| | **Neural networks** | **Decision tree** | **Decision tree** | **Decision tree** | **Neural networks** |
| **Scenario A12** | 0,196 | 0,18 | 0,29 | 0,22 | 0,31 |
| **Scenario A11** | 0,2 (+2%) | 0,171 (-6%) | 0,297 (+1%) | 0,233 (+6%) | Failed |
| **Scenario A22** | 0,191 (-2%) | 0,169 (-7%) | 0,3 (+2%) | 0,239 (+8%) | 0,315 (+1%) |
| **Scenario B1** | 0,248 (+27%) | 0,222 (+22%) | 0,299 (+2%) | 0,28 (+27%) | 0,332 (+7%) |
| **Scenario B2** | 0,238 (+22%) | 0,207 (+14%) | 0,302 (+3%) | 0,284 (+29%) | 0,333 (+7%) |

Table 8: Average Quadratic Errors and their evolution for different scenario of Known Variables (KV) and Intermediate Variables (IV) use.

The table 8 shows that intermediate variables are essential (AQE increases strongly with B1 and B2 scenarios). A22 scenario (All KV and More sensitive IV) will be used for the prediction of Y1 and Y2 output variables. A12 scenario (More sensitive KV and IV) for the prediction of Y3, Y4 and Y5 output variables.

#### *4.2.4. Distribution of Output Variables*

During the stage of classifiers selection, the values of output variables were distributed in 5 classes. The table 9 gives the percentage of misclassification for different distributions of the output variables. Models with the highest number of classes, while not exceeding a misclassification of more than 5% were chosen. Y2 and Y3 variables were distributed in 6 classes. Y1, Y4 and Y5 were distributed in 7 classes.

| Classes # | Y1 (IS) | Y2 (CS) | Y3 (CP) | Y4 (CA) | Y5 (ARE) |
|---|---|---|---|---|---|
| **3** | 1% | 0% | 2% | 0% | 1% |
| **5** | 3% | 3% | 3% | 3% | 2% |
| **6** | 5% | 2% | 5% | 4% | 4% |
| **7** | 5% | 11% | 7% | 5% | 5% |
| **9** | 7% | 11% | 7% | 7% | 8% |
| **11** | 6% | 8% | 7% | 8% | 11% |

Table 9: Percentage of misclassification depending on the number of classes of output variables.

#### *4.2.5. Optimization of classifiers*

The architecture and parameters of classifiers were optimized by using test series, as described in section 3.3.3. Table 10 gives some results for different meta-classifiers after optimization of all parameters. Finally, Stacking meta-classifier are used for the prediction of Y1, Y2 and Y5 output variables. Bagging meta-classifier are used for the prediction of Y3, Y4 output variables.

| | Y1 (IS) | Y2 (CS) | Y3 (CP) | Y4 (CA) | Y5 (ARE) |
|---|---|---|---|---|---|
| **Before optimization** | 0,191 | 0,169 | 0,294 | 0,220 | 0,312 |
| **Boosting** | 0,100 | 0,022 | 0,125 | 0,182 | 0,099 |
| **Bagging** | 0,102 | 0,088 | 0,113 | 0,142 | 0,127 |
| **Stacking** | 0,058 | 0,019 | 0,170 | 0,192 | 0,087 |
| **2-classes combination** | | 0,097 | 0,121 | 0,148 | |

Table 10: Average Quadratic Errors for different meta-classifiers after classifiers optimization.

The table11 summarizes the configuration and final evaluation of learning models.

*4.2.6. Quality of models*

Table 12 gives the percentage of correctly classified instances and the number of unacceptable errors (when the estimated value is more optimistic than the real value) on 80 new cases that have not been used for learning. These confidence rates are satisfactory with regard to the estimation of costs ($y2$, $y3$ and $y4$). The confidence rates are satisfactory but should be improved with regard to the estimation of error on analysis ($y1$ and $y5$).

| | y1 (IS) | y2 (CS) | y3 (CP) | y4 (CA) | y5 (ARE) |
|---|---|---|---|---|---|
| **Correctly classified instances** | 94% | 98% | 100% | 100% | 91% |
| **Unacceptable errors** | 0 | 0 | 0 | 0 | 2/80 |

Table 12: Classifier confidence rates.

*4.3. Validity domain*

From the perspective of the objective of the preparation, the proposed method was applied to CFD analysis. Thus, this approach can be used for all preparation objectives for which the simulation is applied to a fluid volume. In this case, only the explanatory variables are different and should be selected among the set of potential variables. For other preparation objectives, it will be necessary to propose a new description of the preparation process but the main strategy to find the classifiers remains valid. From the perspective of the preparation operations, the study was limited to six simplification operations

that are described by "all or nothing" parameters. The adaptation and meshing processes were the same for all examples. If we wish take into account a greater number of preparation operations and describe them more precisely, it will be necessary to add new variables that describe these operations. Each variable will be described by a greater number of values. Knowing that it is necessary to have at least ten significant examples for each value, carefully selected examples will be added.

## 5. Conclusion and perspectives

In this paper, a new approach to evaluate a priori the impact of CAD model simplification processes on simulation results has been developed. The idea is to make use of Machine Learning Techniques to configure a set of classifiers from a set of known examples. Once configured, the classifiers can be used to estimate a priori what would be the impact of a new unknown preparation process on the simulation results. Thus, it is possible to evaluate a preparation process without doing it. Engineers can thus save a lot of time and test several preparation processes before focusing on a particular one that they will anyhow have to do. Five output variables are used to evaluate a preparation process: the preparation cost, the meshing cost, the simulation cost, the simplification impact on sub-assembly and on the overall analysis result error. Data have been extracted from preparation processes description and CAD models. They have been implemented in vectors in order to be used by classifiers.

The choice of input variables has been a real challenge. The algorithm used to find the relevant input data has been validated by using classifiers. Another algorithm has been proposed to test different classifiers or several configurations and criteria have been proposed to identify the best configured classifier for each output variable.

The satisfactory ratings of the classifiers confidence indicator show that using AI techniques is a good mean for the a priori estimation of preparation processes costs and analysis result error. However, there were some deficits in

some classes of the learning base. The creation of additional models in these classes should improve the classifiers' confidence indicator. One of the major difficulties encountered during this study was the small size of the learning database even if 325 examples have been built and simulated. This problem has been solved with a robust method to build a representative database and by learning with cross-validation method. Further studies, should also treat more complex examples with a larger number of sub-assemblies and parts. In order to optimize algorithms of classifiers, it then could be envisaged to use genetic algorithms combined to the classifiers.

However, the global preparation process proposed at the end of our workflow can still be optimized. Actually, our workflow estimates the impact of a given process but does not directly identify the best process. Further studies should therefore focus on an optimization loop so that using the developed indicators, the best process can be suggested to the designers. The combined use of classifiers such as neural networks with genetic algorithms allows optimizing the design of mechanical products. It could be one way in future studies to identify the optimal preparation process with respect to costs and errors minimization. The preparation process quality takes into account the impact of simplification on analysis result. The proposed method could be extended to other steps of preparation model like meshing.

At the end, the proposed approach and the developed tools reduce the time spent to adapt a complex DMU to a particular simulation while controlling the quality of the analysis results. More broadly, the approach could be extended to other applications which require a preparation process such as the visualization of large DMU or the detection of collisions in large DMU.

## References

[1] T. M. Mitchell, Machine Learning, 1st Edition, McGraw-Hill, Inc., New York, NY, USA, 1997.

[2] A. Thakur, A. G. Banerjee, S. K. Gupta, A survey of cad

model simplification techniques for physics-based simulation applications, Computer-Aided Design 41 (2) (2009) 65–80. doi:http://dx.doi.org/10.1016/j.cad.2008.11.009.

[3] NX5, Siemens plm software, https://www.plm.automation.siemens.com.

[4] GPURE, Deltacad, http://gpure.net.

[5] R. Sun, S. Gao, W. Zhao, An approach to b-rep model simplification based on region suppression, Computers and Graphics (Pergamon) 34 (5) (2010) 556–564. doi:10.1016/j.cag.2010.06.007.

[6] C. Barber, D. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Transactions on Mathematical Software 22 (4) (1996) 469–483.

[7] A. Sheffer, Model simplification for meshing using face clustering, CAD Computer Aided Design 33 (13) (2001) 925–934.

[8] K. Inoue, T. Itoh, A. Yamada, T. Furuhata, K. Shimada, Face clustering of a large-scale cad model for surface mesh generation, Computer-Aided Design 33 (3) (2001) 251 – 261. doi:http://dx.doi.org/10.1016/S0010-4485(00)00124-X.

[9] CATIAV5, Dassault systems, http://www.3ds.com/fr/produits-et-services/catia/produits/catia-v5.

[10] J. Tang, S. Gao, M. Li, Evaluating defeaturing-induced impact on model analysis, Mathematical and Computer Modelling 57 (3-4) (2013) 413–424. doi:10.1016/j.mcm.2012.06.019.

[11] R. Ferrandes, P. Marin, J.-C. Lon, F. Giannini, A posteriori evaluation of simplification details for finite element model preparation, Computers and Structures 87 (1-2) (2009) 73–80. doi:10.1016/j.compstruc.2008.08.009.

[12] S. Gopalakrishnan, K. Suresh, A formal theory for estimating defeaturing-induced engineering analysis errors, CAD Computer Aided Design 39 (1) (2007) 60–68. doi:10.1016/j.cad.2006.09.006.

[13] K. Lee, T. Chong, G.-J. Park, Development of a methodology for a simplified finite element model and optimum design, Computers and Structures 81 (14) (2003) 1449–1460. doi:10.1016/S0045-7949(03)00084-1.

[14] G. Foucault, J.-C. Cuillire, V. Franois, J.-C. Lon, R. Maranzana, Adaptation of CAD model topology for finite element analysis, Computer-Aided Design 40 (2) (2008) 176 – 196. doi:http://dx.doi.org/10.1016/j.cad.2007.10.009.

[15] F. Danglade, J.-P. Pernot, P. Vron, On the use of machine learning to defeature cad models for simulation, Computer-Aided Design and Applications 11 (3) (2014) 358–368. doi:10.1080/16864360.2013.863510.

[16] A. Jahangirian, A. Shahrokhi, Aerodynamic shape optimization using efficient evolutionary algorithms and unstructured cfd solver, Computers & Fluids 46 (1) (2011) 270 – 276, 10th {ICFD} Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010). doi:http://dx.doi.org/10.1016/j.compfluid.2011.02.010.

[17] A. Kharal, A. Saleem, Neural networks based airfoil generation for a given using bezierparsec parameterization, Aerospace Science and Technology 23 (1) (2012) 330 – 344, 35th ERF: Progress in Rotorcraft Research. doi:http://dx.doi.org/10.1016/j.ast.2011.08.010.

[18] X. Yuan, J. Hongfan, W. Yu, A neural network approach to surface blending based on digitized points, Journal of Materials Processing Technology 120 (1-3) (2002) 76–79. doi:10.1016/S0924-0136(01)01105-0.

[19] H. M. Gomes, A. M. Awruch, P. A. M. Lopes, Reliability based optimization of laminated composite structures using genetic algorithms and

artificial neural networks, Structural Safety 33 (3) (2011) 186 – 195. doi:http://dx.doi.org/10.1016/j.strusafe.2011.03.001.

[20] W. Chan, M. Fu, J. Lu, An integrated fem and ann methodology for metal-formed product design, Engineering Applications of Artificial Intelligence 21 (8) (2008) 1170–1181. doi:10.1016/j.engappai.2008.04.001.

[21] Y. Chen, G. Kopp, D. Surry, Prediction of pressure coefficients on roofs of low buildings using artificial neural networks, Journal of Wind Engineering and Industrial Aerodynamics 91 (3) (2003) 423–441. doi:10.1016/S0167-6105(02)00381-1.

[22] F. Mazhar, A. M. Khan, I. A. Chaudhry, M. Ahsan, On using neural networks in uav structural design for cfd data fitting and classification, Aerospace Science and Technology 30 (1) (2013) 210–225. doi:http://dx.doi.org/10.1016/j.ast.2013.08.005.

[23] M. A. A. Oroumieh, S. M. B. Malaek, M. Ashrafizaadeh, S. M. Taheri, Aircraft design cycle time reduction using artificial intelligence, Aerospace Science and Technology 26 (1) (2013) 244–258. doi:http://dx.doi.org/10.1016/j.ast.2012.05.003.

[24] Y. Uematsu, R. Tsuruishi, Wind load evaluation system for the design of roof cladding of spherical domes, Journal of Wind Engineering and Industrial Aerodynamics 96 (10-11) (2008) 2054–2066. doi:10.1016/j.jweia.2008.02.051.

[25] P. Lopes, H. Gomes, A. Awruch, Reliability analysis of laminated composite structures using finite elements and neural networks, Composite Structures 92 (7) (2010) 1603–1613. doi:10.1016/j.compstruct.2009.11.023.

[26] T. Buar, M. Nagode, M. Fajdiga, An improved neural computing method for describing the scatter of s-n curves, International Journal of Fatigue 29 (12) (2007) 2125–2137. doi:10.1016/j.ijfatigue.2007.01.018.

[27] S. Jayanti, Y. Kalyanaraman, K. Ramani, Shape-based clustering for 3d cad objects. a comparative study of effectiveness, CAD Computer Aided Design 41 (12) (2009) 999–1007. doi:10.1016/j.cad.2009.07.003.

[28] V. Sunil, S. Pande, Automatic recognition of machining features using artificial neural networks, International Journal of Advanced Manufacturing Technology 41 (9-10) (2009) 932–947. doi:10.1007/s00170-008-1536-z.

[29] I. Gonzalez-Carrasco, A. Garcia-Crespo, B. Ruiz-Mezcua, J. L. Lopez-Cuadrado, An optimization methodology for machine learning strategies and regression problems in ballistic impact scenarios, Applied Intelligence 36 (2012) 424–441. doi:10.1007/s10489-010-0269-5.

[30] N. Iyer, K. Jayanti, S.and Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, Computer Aided Design 37 (5) (2005) 509 – 530.

[31] C. Lemke, M. Budka, B. Gabrys, Metalearning: a survey of trends and technologies, Artificial Intelligence Review 44 (2015) 117–130. doi:10.1007/s10462-013-9406-y.
URL http://dx.doi.org/10.1007/s10462-013-9406-y

[32] WEKA, University of waikato, http://www.cs.waikato.ac.nz/ml/weka.

| Classifier | Classifiers architecture and learning initilization | Parameters to optimize or refine by learning |
|---|---|---|
| Neural Network | Neural network model (Perceptron or Adaline)<br>Structure (multilayer feedforward network, fully recurrent network, recurrent netwok with self connections, ...)<br>Activation function (Step, Linear, Log-Sigmoid or Tan-Sigmoid)<br>Number of hidden layers<br>Value of momentum applied to the weights during updating<br>Number of iterations<br>Validation threshold<br>Stop condition (from number of iteration or validation threshold) | Number of nodes by layer<br>Weights of the connections betweem nodes |
| Decision tree | Decision tree model (CART or C4.5.)<br>Pruning strategy (no-pruning, post-pruning, pre-pruning<br>Node selection criterion (entropy measure, Fisher test, Gini index,...)<br>Stop condition on terminal nodes | Tree architecture<br>Number of nodes<br>Classes on terminal nodes |
| Support vector machine | Kernel function (Linear, Polynomial, Gaussian radial basis function,...)<br>Parameters depending of the Kernel function (Bias-variance compromise , Gamma,)<br>Tolerance of the termination criterion | Optimal hyperplan |
| Naive Bayes methods | Type: classifier or net<br>Estimator algorithm<br>Method used for searching network structures | Rules or net architecture<br>Models parameters |

Table 4: Main classifiers and learning parameters.

| **Y1 (IS)** | **Y2 (CS)** | **Y3 (CP)** | **Y4 (CA)** | **Y5 (ARE)** |
|---|---|---|---|---|
| **Classifiers :** | | | | |
| Neural Network Multi-Layer Perceptron<br>• HL: 1<br>• N : 15<br>• LR : 0,1<br>• M : 0,1 | Decision tree C4.5<br>• CFP : 0,75<br>• L : 27<br>• S : 53<br>• P: no | Decision tree C4.5<br>• CFP : 1<br>• L : 12<br>• S : 23<br>• P : yes | Decision tree C4.5<br>• CFP: 0.75<br>• L : 8<br>• S : 15<br>• P : yes | Neural Network Multi-Layer Perceptron<br>• HL: 1<br>• N : 5<br>• LR : 0,1<br>• M: 0,1 |
| **Meta-classifier :** | | | | |
| Stacking | Stacking | Bagging | Bagging | Stacking |
| **Use of variables :** | | | | |
| All KV + More sensitive IV | All KV + More sensitive IV | More sensitive KV and IV | More sensitive KV and IV | More sensitive KV and IV |
| **Classes #:** | | | | |
| 7 | 5 | 5 | 7 | 7 |
| **Learning models evaluation (AQE):** | | | | |
| 0,058 | 0,019 | 0,113 | 0,142 | 0,087 |

Table 11: Learning model configuration and evaluation (AQE)after optimization. HL = number of Hidden Layer. N= nuber of nodes in hidden layer. LR = Learning Rate. M = Momentum applied to the weights. CFP = Confidence Factor for pruning. L# = Number of Leaves. S = Size of the tree. P = Pruning