Input Variable Selection in Real-time Knowledge Integration Applications: A Review, Analysis, and Recommendation Paper

S. Tavakoli, A. Mousavi, and S. Poslad

***Abstract*—The purpose of this paper is in two folds; the first is to undertake a thorough appraisal of the existing input variable selection (IVS) methods in the context of time-critical and resource-limited dimensionality reduction problems. The second is to demonstrate further improvements and the application of a recently proposed time-critical sensitivity analysis method called EventTracker in industry.**

Producing accurate knowledge about the state of a system (effect) in real-time under computational and data acquisition (cause) constraints is a major challenge. Especially if the knowledge required is critical in operations that the safety of operators and/or integrity of equipment is at stake. Understanding and interpreting, a chain of interrelated events, predicted or unpredicted, that may or may not result into a specific state of the system is the focus of this research challenge. The objective is to identify which set of input data/signal has significant impact on the set of system state information (i.e. output). Through this cause-effect analysis process, the proposed technique filters unsolicited data that may clog up communication and computational capabilities of a standard Supervisory Control and Data Acquisition System.

The outcome of this research project is a series of issues adhered to and suggesting the difficulty of finding an established method suitable for time-critical variable selection applications. However, supported by a geological drilling monitoring application, the authors are able to substantiate the aptness of the EventTracker Sensitivity Analysis method in high volume and time critical dimensionality reduction.

In order to prove the advantages gained in performance and computational efficiency by adopting the proposed sensitivity analysis method, a general comparison and evaluation of other established input variable selection techniques is conducted.

Index Terms— **Input Variable Selection, Time-critical Control, Dimensionality Reduction, Sensitivity Analysis, Supervisory Control and Data Acquisition**

1. INTRODUCTION

In a complex interrelated world, the industry is faced with ever changing performance metrics. This complexity and relentlessness of change both in substance and in presentation is forcing companies to make ever larger investments in data acquisition and interpretation technologies. The dilemma of "usefulness" and "relevance" [8], [38] for the investor still prevails. In addition, there is a direct relationship between identifying useful-relevant input data and the levels of investment on data acquisition, communication and computational capabilities for performance measurement.

The success of identifying the useful-relevant input data that affect performance metrics relies on the speed and the quality of the process that separates the useful-relevant from non-useful and non-relevant input data.

In this paper the authors will compare the existing analytical techniques for measuring usefulness and relevancy of input data. The comparison will be from two perspectives. The first will be based on how practical or applicable those techniques are for dealing with real-time systems. The second base for comparison will be between the computational overheads of different input data analysis techniques. In order to implement the comparison process, a comprehensive review of the existing literature on Input Variable Selection (IVS), Feature Selection (FS) and Sensitivity Analysis (SA) subject areas will be conducted. The paper will use a real industrial case study to examine the applicability of existing IVS, FS and SA methods against the requirements of the problem.

In the following sections the reader will be introduced first to design of an interrogation survey for system 'variable' and 'feature' extraction (section 2). In Section 3, an analysis of the computational effort required for established system variable and feature selection methodologies is offered.. At the end of Section 3 the authors conclude that the Sensitivity Analysis (SA) method, EventTracker [65] is the most suitable technique to determine input data usefulness and relevancy in real time application. In section 4, the authors will further explain how EventTracker as generic tool is applicable to real-time performance monitoring of deep drilling operations. Section 5, describes the

case study in detail and demonstrates the implementation of EventTracker and the results. And finally conclusions of the research project are provided in Section 6.

2. IVS AND FEATURE SELECTION

Although the concepts of Input Variable Selection (IVS) and feature selection appear similar, there are key aspects that differentiate them. This section provides an analysis of such aspects with focus on their role in dimensionality reduction and associated computational overhead.

Feature Selection (FS) is a well-known problem addressed by a large amount of research and literature [13, 16, 24, 26, 27, 30, 65]. The objective of this paper is to propose a technique to decide which input data sources are useful and relevant to determine the state of a given system (e.g. output). For this purpose we distinguish between FS and IVS from two aspects; one is the nature or substance of input variable and feature, and the other aspect is their selection methodologies. Both aspects are explained in the following two subsections.

2.1 Input variable and feature

Although the terms "variable" and "feature" are used and treated with marginal differences, input variables differ in nature from features. Input variables like raw input data series generally provide information about the system, whereas features are used to represent a model of the system. A feature may be locally and temporarily created to support the understanding of some specific behavior in the system. Based on the problem in hand, non-sequential data (i.e. snapshots) may be sufficient for a given data mining task, but continuous information of input variables are time-critical in certain applications [28] where system state changes rapidly in time (Volatile Systems). This conceptual difference is shown in Figure 3.

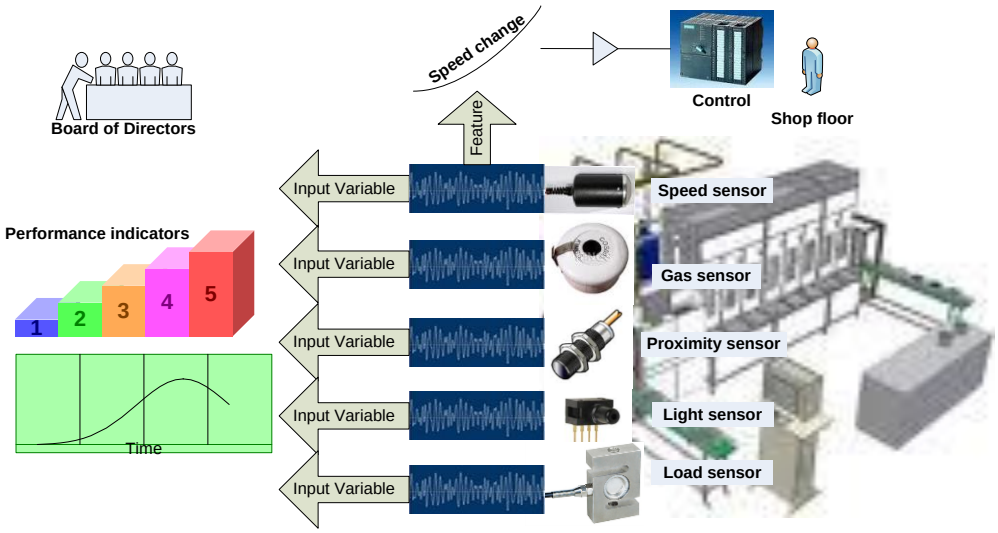

Figure 3 Input variables are used differently compared to features

Input variables represent knowledge about the direct result of aggregation – and pre-filtering - of raw input data from data sources. Like in data fusion, variable construction is an overlay task of aggregating raw input data sources.. Feature, however, is meant to add to the aggregated knowledge when input variables do not provide the required knowledge with adequate certainty, efficiency or based on other objectives.

Features are created in two ways:

- They are created either by application of specific transfer functions based on the input variables raw input data and combination of both. The so called feature construction extends the knowledgebase.
- Or they are created by finding patterns in the data series. .

With the help of an example, we describe feature construction and feature extraction in the following subsections.

2.1.1 Features derived from input variables (feature construction)

When a system is being monitored, key features that describe system status are separated from the initial input variables generated by field sensors. Each constitute a distinct layer in the data acquisition architecture, the sensor level is primary and the feature level is intermediate [1]. For

example, the input variables for a typical production line are specified to assemble the initial feature candidates that in turn are interpreted into indicators of shop floor status and job characteristics [41]. Transformations and combinations of these primary data and intermediate features are then used to extract the key performance indicators [41].

In another example, in order to detect faults in rechargeable batteries (i.e. state), two performance indicators are required, capacity and life cycle [52]. These two indicators are measured based on the amount of charge and the number of completed charge/discharge cycles (i.e. input variables) prior to the nominal capacity falling below a specified value. In order to shorten the measurement cycles and accelerate detection process, the paper reports on devising a new set of variables (features) by combining the derivatives of the two variables with different orders.

Figure 4 illustrates feature construction from input variables. The curved arrows symbolize the extra effort spent in converting input variable type knowledge to feature type knowledge. Depending on the type of process, one can expect increases in computational overhead. The cost of the computational overhead may affect further control, monitoring, or decision making process particularly from a time-critical processing point of view.

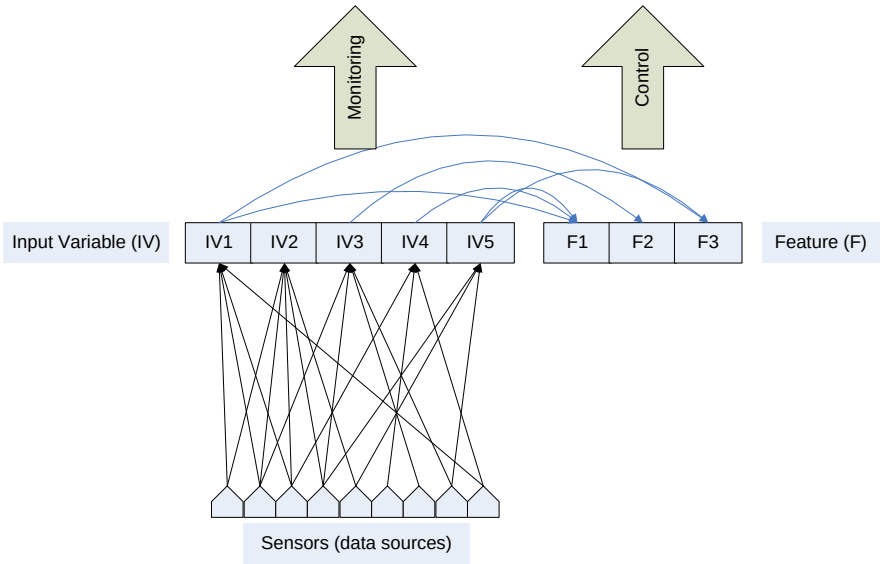

Figure 4 Feature construction from input variables

2.1.2 Features based on data mining (feature extraction)

The key challenge in any data mining and analysis process is to decrease the uncertainty associated with the relationship between input variables and the interpretation of system behavior. It is not always the case that input variables can provide a sufficiently certain view of system's behavior [12, 20]. Neither can input variables be always given adequate model to be built and evaluated. Delving into time series of acquired data (and input variables) can assist to figure out features of the data sources which are not previously known and actionable. For example, in order to understand complex characteristics of bioprocesses and enhance production robustness, [15] descriptive (e.g., frequent pattern discovery, clustering) and predictive (e.g. classification, regression) pattern recognition methods have been applied. As a result, significant trends in processing data sourced from archived temporal records of physical parameters and production scale data were discovered [15].

Furthermore, a Virtual Metrology system capable of predicting every wafer's metrology measurements based on production equipment data and metrology results [36] collected four extracted statistics, such as mean, variance, minimum, and maximum value from each sensor during two etching processes used by a Korean semiconductor manufacturing company.

Data mining and feature extraction require additional effort (see Figure 5) for data warehousing (represented by database) and historical data processing (represented by blue arrows) because they are heavily reliant on historical data. This increases computational costs and may affect time-critical aspects of monitoring.

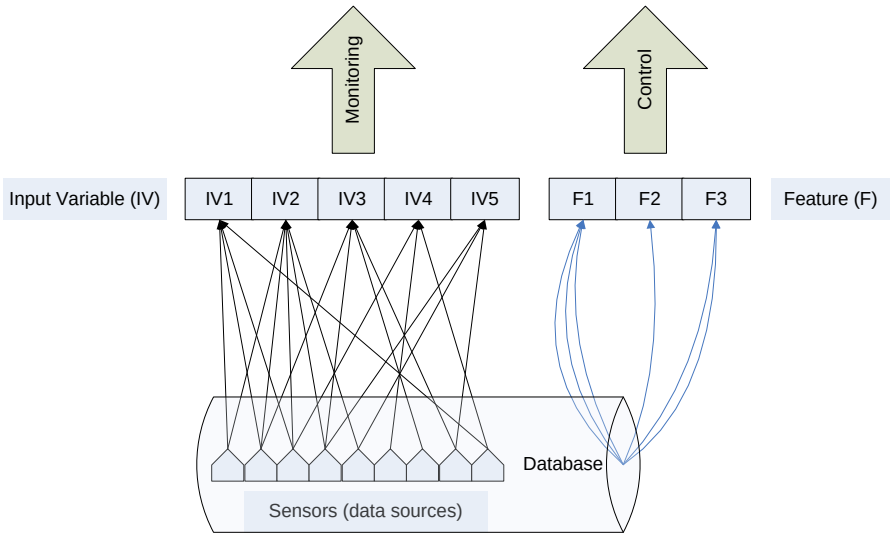

Figure 5 Feature extraction from raw data

2.2 Selection methods based on derived variables

There is evidence from research and practice of measuring computational efficiency of IVS methods by using derivations of the original inputs to select and build new variable subsets [11, 29, 37, 43]. The term "derived variable" has been used for the same purpose in the data mining literature, including [28], suggesting that techniques such as Projection Pursuit Regression (PPR) and Principle Component Analysis (PCA) can be used efficiently for variable transformation. The PPR technique is a common multivariate regression technique and the PCA is a special case of PPR. Regression and cluster analysis [32], [48] are two well-known bases for many dimensionality reduction techniques. The pros and cons of both regression and cluster analysis methodologies for IVS are discussed in the following two sections.

2.2.1 Regression

The task of estimating a map of relationship between a number of independent variables and a dependent variable called regression [28]. In this paper, independent variables are regarded as input variables to the model. Dependent variables are interpreted as the model's performance indicators or system outputs. According to the regression method:

1. When there is a requirement to predict the value of a dependent variable from new values of

independent variables and no predictive model between the independent variables and dependent variable exists,

2. When a new set (usually fewer in number) of independent variables are used to replace the original set of independent variables are expected to lead to the same effect.

The heterogeneous nature of the data distributions that represent input variables can prevent assumptions about the nature of the relationship between the independent and dependent variable. This leads to the consideration of nonlinear and non-parametric regression methods. In [6] there is an explanation that by growing the dimension of input variables, the number of possible regression structures increases faster than exponentially. The faster than exponential increase in number of exponential structure, makes regression methods extremely unreliable for input data analysis.

Reference [6] compares the performance of regression techniques by conducting simulation experiments on ten prominent regression methods. They considered the effect of six factors in their experiments: regression method; embedded functional relationship between the data of independent and dependent variables; number of variables (dimension); sample size; added noise to the sample data; portion of involved variables in the function (model sparseness). Of the ten regression methods examined, none is applicable to all conditions [6]. They report that Recursive Partitioning Regression (RPR) is able to cope with high numbers of dimensions (12 variables) when all variables were involved (explanatory). They however, recommend analysts to engage in the process of selection of the regression method by trying portions of data on each method until they seem to be a reasonable fit.

From above analysis and overview, one can conclude that no particular regression method is capable of covering the scale and heterogeneity of variables involved in volatile situation of industrial systems whilst keeping the computational cost low. Although some guidance is given concerning the selection of a suitable IVS method, e.g. to be incremental rather than memory-based, it is vital to explore other dimensionality reduction methods, such as clustering.

2.2.2 Cluster analysis

Cluster analysis methods use “similarity criteria” so that a group of data values that are “similar” can either be replaced by a new value representing the group (clumping) or assigned a unique type of label (partitioning) [32]. As a basic example, K-Means clustering [48] divides a set of data entities into K non-overlapping clusters of similar data and each cluster is represented by the mean value of its data (the centroid). The choice of the number of clusters (K) and similarity criteria are the two main challenges in this approach.

As a type of clustering method with specific similarity criteria and automatic selection of number of clusters, Principal Component Analysis (PCA) replaces a number of input variables that are correlated by a smaller number of variables which are not correlated (principal components) and which at the same time keep the same variability of the original input variables [35]. Given a fixed set of input variables PCA always produces a unique set of new variables independent of the analysis of model performance factors. A key issue here is at the choice of using one of the actual (genuine) entities in each group to represent the group instead of generating an artificial one, e.g., a mean value [48].

In [32] the fundamental challenges associated with clustering are highlighted as data representation, the purpose of grouping, cluster validity, and cluster algorithm comparison. It is understood that although there is generally no universally good data representation, domain knowledge aids the clustering process. In the general case of this paper where input variables are assumed to hold heterogeneous data series in the time domain, there is no clear choice of changing and selecting a particular type of data representation. In terms of grouping, the added dimension in the time domain, as is the general assumption of this paper, raises potential issues (see Figure 6); To start with, one common indicator or variable is required to represent the time series of input variables (see Figure 7) or alternatively, a similarity criteria could be applied between each two input variable time series as shown in Figure 8. This on the one hand is computationally exhaustive, and on the

other hand imposes a limitation on the choice of similarity criteria between the time series due to the different sampling rates of different input variables, and therefore, the different size of the collected samples.

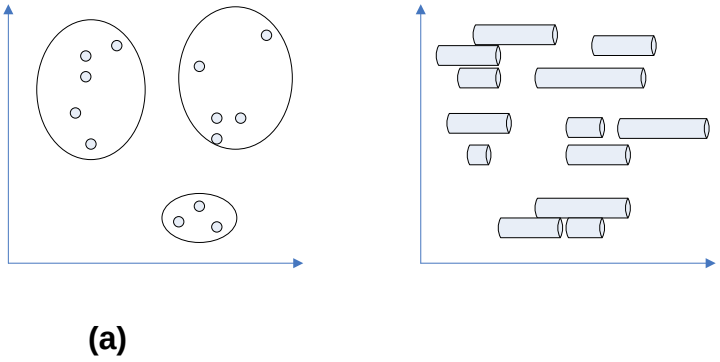

Figure 6 Each data entity in (a) is singular while in (b) is a time series of data

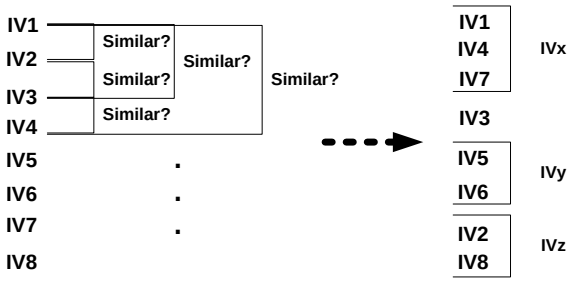

Figure 7 Input Variables IV1-8 are grouped based on their similarity and represented by one common Input Variable IVx–z

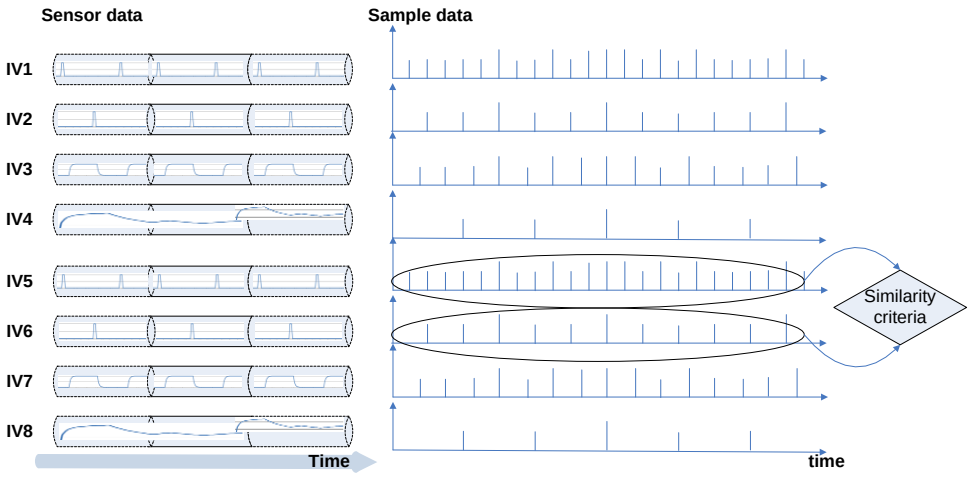

Figure 8 A similarity criteria should decide on the grouping of data time series by working on chunks of each two time series acquired at identical time intervals

Reference [32] concludes that due to the unknown prior knowledge about the structure of data, no best clustering algorithm exists and a diverse set of approaches often need be tried before determining an appropriate algorithm for the clustering problem at hand.

In general, DVIVS find the most appropriate set of input variables according to some criteria that may or may not consider the output performance of the system. [25] and [68] introduced a variable selection criteria based on numeric data based fuzzy modeling, which was explored by [62]. [25] performed an input-output sensitivity analysis to remove input variables holding the least maximum normalized sensitivity index in a backward elimination fashion. [68] proposed an Input Variable Selection Criterion (IVSC) function, which estimates the importance of each input variable numerically (most important input variables are kept using forward selection scheme).

In DVIVS methods, as classified in this paper, to evaluate newly (artificial) created variables, the values of original variables need to be acquired when the new variables are evaluated. This adds a burden to real-time data integration systems that wish to employ an IVS method with a low computational overhead. OVIVS, in contrast to DVIVS, avoids this burden.

2.3 Selection methods based on original variables

OVIVS tends to reduce the computational cost of data integration by avoiding high rate of sampling and processing the less important portion of input data. It is however important to invalidate this tendency by exploring OVIVS methods.

2.3.1 Variable ranking

Variable ranking has proved to be a sound approach in many variable selection algorithms due to its simplicity and scalability [26]. In variable ranking, a scoring (or scaling) function is used to compute and assign a quantitative score [48] for each input variable in relation with a target class, e.g. correlation coefficient [54], and then variables are sorted based on their score [26].

Authors in [54] present a simple ranking algorithm that seems to be superior to more complex

state-of-the-art ranking algorithms. The powerful computation time performance of their ranking method mainly comes from the explicit and sequential implementation of two modules in a cycle of selection and elimination tasks [54]. First, a correlation-based criterion examines and ranks input variables with respect to output variables. Then an orthogonalization module applies redundancy detection and variable elimination by normalizing input variables according to the top ranked variable at the same cycle. This sequence is shown in Figure 9.

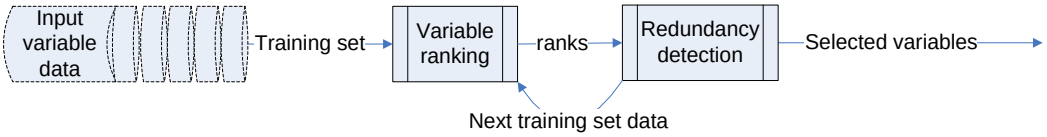

Figure 9 Cyclic ranking and selection of variables based on sequential data sets

The loop stops when either all input variables are processed, or when the number of training data values is smaller than the number of input variables.

In [54] a Simplified Polynomial Expansion (SPE) as a sufficiently good approximation of general nonlinear models that map input variables to output variables is used. Therefore, an initial computational effort is required at the beginning of each cycle (embedded in variable ranking module) to accomplish SPE between output variables and remaining input variables. They experimented using an SPE-ranker algorithm on artificially generated data sets. The stoppage criterion, however safe, leads to long runs for systems with a high number of input variables, since for example, for a system with 100 input variables the algorithm needs at least 100 execution cycles for processing 100 samples. With a typical sampling rates of up to 5 samples per second, this takes at least 20 seconds before one full decision on the elimination (or change of acquisition settings) of input variables could be taken.

The type of approach to measure the performance of a selected set of variables could affect the accuracy of the resulting selected variable set as well as the computational cost. As seen in Figure 10, approaches differ in the way variable selection module and selection validation modules interact,

i.e. if the loops are embedded, or are they a one-off incident. They also differ in the number of data sets involved from input or output variables i.e. none, once, or multiple times that data sets for training or test purposes are used.

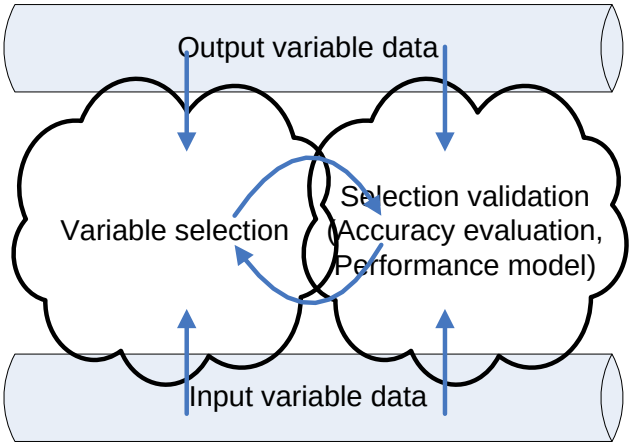

Figure 10 Different approaches could exist for interaction between variable selection task and selection validation task

Any of the above variable selection methods may adopt a wrapper, embedded or filter approach.

2.3.2 Wrapper methods

The wrapper approach [38] measures and compares the usefulness of different subsets of input variables with respect to decision making parameters. To achieve this, as shown in Figure 11, the approach selects the input variable and measures their influence on the performance model.

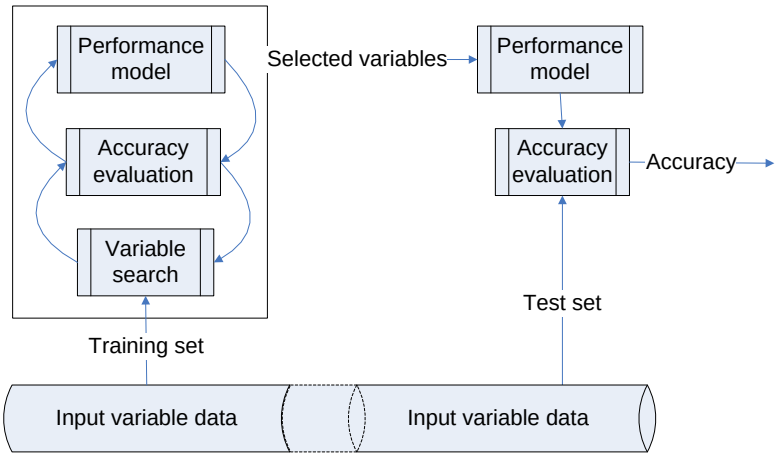

Figure 11 Conceptual representation of a wrapper approach to variable selection

Therefore, similar to a learning machine, execution of iterative runs of the performance model

using different variables (training set) at each run [54], and finally examination of selected variables with a different set of variable data (test set) contributes to the computational overhead [63]. [44] introduced a wrapper method for measuring the importance of input variables based upon predictive models, and probability distribution of input variables. In a wrapper methodology, at occasions when computation is too exhaustive for large number of variables, heuristics such as backward elimination and forward selection [50] may reduce but not necessarily eliminate the overall computational cost [68].

2.3.3 Embedded methods

In an embedded method, similar to wrapper method, an iterative learning mechanism is applied to determine the importance of the subsets of input variables. However, in contrast to a wrapper method, an embedded method does not use a closed learning mechanism which works independent of the actual variable selection function. Instead, as shown in Figure 12, an embedded method incorporates variable selection in the learning mechanism. This accelerates the finding of solutions as well as leading to more accurate outcome [26].

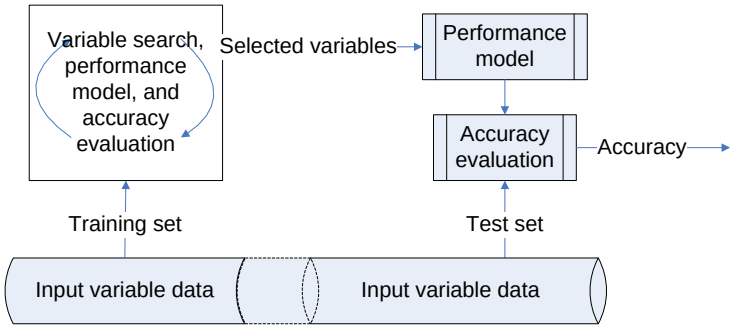

Figure 12 Conceptual representation of an embedded approach to variable selection

Nevertheless, both embedded and wrapper methods require iterative executions of the performance model as well as historical variable datasets for the training and test phases. Hence, the associated computational overhead is still of concern.

2.3.4 Filter method

Filtering of irrelevant and redundant variables, if accurate enough, could help with lowering the computational complexity of data acquisition [54] and the search space [55]. In a filter method, data and its properties are assessed for relevance using an independent criterion function. Therefore, the details of algorithms and models that govern the output variables (system performance data) have no effect on the variable selection. Since a filter method incorporates a one-step learning-based wrapper method, as shown in Figure 13, it is a low computationally complex process [63]. However, for the same reason - of not implementing a multi-step learning process - filter methods do not necessarily find the most accurate and useful subset of variables [26].

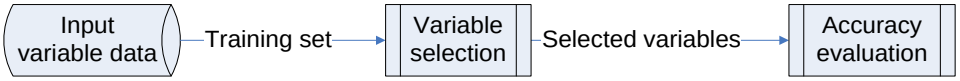

Figure 13 Conceptual representation of filter approach to variable selection

Selection of an IVS method encompasses as well as assessment of performance, understanding of the type and amount of computational effort that must be spent to implement and execute it. Questions like ‘should exclusive knowledge of the variables be incorporated?’, ‘should there be an automatic learning approach?’, ‘should a simulation support observation of performance with different sets of variables?’ and so on, come from different points of view. This leads us to contain the aforementioned methodologies in a set of viewpoints or perspectives.

3. IVS METHODOLOGIES FROM A COMPUTATIONAL PERSPECTIVE

This section provides an overview to the IVS methodologies from several perspectives. The various perspectives reported in this section explore the initial assumptions about the nature dependent and independent variables within a search space. The report also covers the efforts made in the literature to describe the deterministic or stochastic nature of the relationships between input variables and output performance measures.

3.1 Heuristic approaches

The scale of the problem and the time constraints on the IVS as well as the discrete nature of the search space seems predisposed to a heuristic approach [4]. Heuristics refers to the search strategies that use problem-specific knowledge to find a solution [50]. The way heuristic can help with solving complex problems is through ignoring some aspects of the problem avoiding further computational processes and therefore complexity. Therefore, in using a heuristic approach for IVS, some input variables to a problem are known to be less important and more computationally expensive. These could be ignored in the process of ranking or redundancy detection. Thus, the heuristic function which evaluates the cost of the input variable acquires extra information about the computational method applied to the input variable to the system.

Coding and maintaining heuristic rules are easier than optimization procedures [4]. However expert knowledge of the factors to determine rules of heuristics is required. This limits this approach limited to well-defined and known problems [4].

3.2 IVS through Optimization and Simulation

It may be argued that input variable selection is similar to an optimization problem. Hence the objective of IVS could be rephrased to "minimizing the computational overhead caused by sampling, processing, and storage of unnecessary values of inputs to a system subject to not losing the accuracy of system state interpretation". The amount of computational overhead is understood to be directly proportional (linear or non-linear) to the number of input variables [54]. Therefore, the objective leads to strategies to minimize the number of input variables, or more actually, to minimize their sampling rates while maintaining the accuracy of performance variables of the model.

The difficulty in adopting this approach for use with IVS arises out of the stochastic nature of the values of objective i.e. model's performance variables. The accuracy in these values cannot be evaluated exactly due to its non-deterministic nature. [5] introduced some remedies to the issues of estimated values of stochastic parameters in a simulation based on the execution of multiple

replications and / or longer runs of the simulation. Such solution does not comply with the time constraints of real-time analysis situations.

In other words, optimization via simulation seeks to build a prescriptive model (answering how to set the input variables) from an existing descriptive model (system simulation) [4]. Prescriptive models can grow in size and become non-linear very quickly when details are incorporated, making them virtually impossible to solve optimality [4].

3.3 Statistic perspective

Statistic approaches support extraction of probability distributions and their associated properties from data. For input variables, the approach replaces each input variable with the probabilistic distribution that can represent the time series of the associated input variable with a close enough accuracy (using an error estimation method). For the selection of input variables, the allocated probability distributions may be clustered or otherwise shortlisted according to the result of the analysis against the degree of their influence on the performance of the model.

Statistic perspective is erected based on two features that input variables hold in the generic view that this paper commits to; one is the heterogeneity of data type and therefore, the expected uncertainty about the data series of the input variables. Nothing is assumed to be known about the distribution of data sets. The other is the unlimited nature of the data series that enables a statistic approach to look at them as a sample of a larger distribution [48].

The difficulty with using statistic approach for IVS arises from the limitation on the number of distributions that could be applied to fit the sampled input variable data. It is therefore safe to doubt the suitability of statistical techniques to assess the influence of an input variable on system state within the limited time (real-time). This unsuitability is compounded when a known distribution cannot be ascertained (unaware – also see [65]).

3.4 Machine learning perspective

Application of machine learning necessitates a repetitive use of consecutive values of data in order to set some prediction parameters which could later be used to categorize newly created data [75]. Sufficient samples from each input variable should be tried against a sufficient number of samples of output performance parameters. This requires exhaustive number of executions.

Supervised machine learning that uses performance parameters in the analysis plays an important role in wrapper and embedded-based variable selection methods. The major disadvantages when using a learning mechanism is the requirement to accumulate data about the input variables using repetitive performance model executions making it computationally expensive [76].

The unsupervised learning machine approach has the same on-by-one, repetitive comparing process similar to the supervised learning algorithm. The unsupervised learning algorithm also relies on historical input variable data and conducts goodness-of-fit experiments to verify the relationship between system input and output. One can deduce that such technique would be unsuitable for real-time applications.

3.5 Data mining perspective

Unlike statistical approaches, data mining methods search for patterns and regularities within the available data [48] that are natural and unknown [32]. The data mining methods do not seek to provide a consistent statistical distribution as in statistical analysis or a learning function as in machining learning methods. . Instead, they focus on producing a more compact representation of the given data [73].

In general, data mining-based input variable selection methods focus on the similarity of the input variables independent to their importance or influence on the model performance. Therefore, although a group of similar input variables can be identified as a result of data mining, further effort is needed to determine their importance with respect to the output variables and to decide on the acquisition attributes.

3.6 Classification / knowledge-discovery perspective

Advocates of data mining methods consider data classification as a task in data mining. However, there seems to be a subtle difference in the essence of the knowledge acquired from the two methods. Within the context of IVS, the cause-effect relationship between the input variables and performance variables could be considered as knowledge about the system. Input variables with different levels of cause-effect relationships could be classified into different categories. In classification, in contrast to data mining, categories are pre-defined or synthesized to support finding and grouping data [74]. From a discriminative viewpoint [28], some function is required to maximize a measure of separation between the variables. Such a discriminant function is explored through studying the level of impact that each independent variable has on the dependent variable. The following section opens a discussion on this issue and its available options, under the impression of sensitivity analysis.

4. SENSITIVITY ANALYSIS

Sensitivity analysis (SA) has been discussed by [9, 17, 18, 22, 58] as a technique to minimize the computational overhead by eliminating the input variables that have the least impact on the system. Sensitivity analysis techniques can help with focusing only on the most valuable information that has a significant impact on behavior of systems. The measurement of the true impact of an input on the output of a system becomes challenging due to the epistemic uncertainty of the relationship between the two variables [39]. Selection of an appropriate method for sensitivity analysis therefore, depends on a set of factors and assumptions. A framework is structured to allow us to analyze and compare the performance of SA methods. This framework takes into account at first priority the efforts for exploring the type of relationship between inputs and outputs. The framework also covers the nature of the input and output data series. Last but not least, the framework provides means to exhibit the computational efforts that each SA method requires. These main issues of the framework are introduced in the context of the following three main sub-sections.

4.1 The analytical relationship between the input and the output

The majority of sensitivity analysis methods tend to demonstrate the impact of change in one variable on the other by means of the mathematical equation that describes the relationship between them. Methods such as differential analysis [31], Green's function [70], and coupled/decoupled direct [23] are classified as analytical sensitivity analysis methods by [31]. However, the non-linear and non-monotonic relationship between inputs and outputs for a given system may not necessarily lend themselves to the use of such analytical methods [39]. The reasons for this follow.

4.1.1 Differential analysis

In differential analysis, the impact of an independent variable on the dependent variable is assessed by identification of the perturbation behavior of the dependent variable due on the changes of the independent variable [2]. This is achieved by finding the coefficients of the differential equation that explain the relationship between the independent and dependent variables [2]. Methods such as Neumann expansion [42] and the perturbation method [14] could help extract these coefficients by approximating the differential equation. However, it is cannot be guaranteed that the often complex and nonlinear relationship between system variables could be approximated with differential equations with sufficiently low error margins [31].

4.1.2 Green's function

When differentiating model equations is difficult use of Green's function could act as a catalyst to help achieve the sensitivity equations [31]. Effectively, in this method, differentiation operation is replaced by the sequence of finding the impulse response of the model [21], and the subsequent integration operations. This introduces an auxiliary function proposing that a linear and time-invariant system could only benefit from this solution. Another disadvantage of the application of Green's function method is its ability to work only with the ordinary type of differential equations that govern dependent variables with respect to independent variables. In real applications it is

difficult to separate the relationships of independent variables with dependent variables. Additionally, working one variable at a time for high dimensional systems could be computationally expensive.

4.1.3 Coupled/decoupled direct method

In a coupled direct method, after differentiation of the model equations, the subsequent sensitivity equations are solved together with the original model equations [31]. In decoupled direct methods, they are solved separately. This gives the impression that a decoupled direct method is advantageous in terms of its reduced computational cost. Although a decoupled direct method is reported to be more efficient than use of Green's function, as are other analytical methods, knowledge of the model equations is required. This contributes to the two features of model-oriented and expertise-hungry of the analytical methods, putting them at a disadvantage compared to sensitivity analysis methods that do not require model equations.

4.1.4 Sampling-based methods

Where no mathematical equation is defined for the model variables, or when it is not preferred to work on the existing model equations, other sensitivity analysis methods that do not care about the computational overhead tend to establish a model equation by identification of some statistical features in the distribution of the data series of the two variables. The general shortcoming of these methods such as Fourier Amplitude Sensitivity Test (FAST) [18, 47], Morris [34, 10], Monte-Carlo [61] and Latin Hypercube [19] is their heavy reliance on historical data. This dismisses them from being good candidates for time-constrained applications.

For example, [17] applied their model to 1280 sample values of 20 input parameters. Each cycle of sample generation and model execution took between 2 to 52 hours per set of samples. The overall execution cycles took almost 46 days. This example illustrates the significant impact of the sampling based analysis on the computational overhead. The following subsections analyze the major

sampling based methods.

4.1.4.1.Monte Carlo and Latin Hypercube methods

Random sample generation, as the main characteristic of a Monte Carlo method, provides the values for the independent variables from which dependent variables are produced, based on the execution of the model on sampled input data [57]. The random sampling scheme occurs either in no particular manner, or with some criteria that could help with the efficiency of computation [31]. For example, in Latin Hypercube Sampling (LHS) method [19], the range of each input parameter is divided into intervals of equal probability. In each set of samples of input parameters, each input parameter takes a random value from one of its intervals with no repeat of the same interval for one full sampling cycle [19]. This way, it would be more likely to work on all segments of data in the distribution. Therefore, more informative distribution parameters for generated output could be achieved in a shorter period [31].

An overview of Monte Carlo methods is given in Figure 15. First the probability distribution of input variables is estimated from the available data stream, i.e. using curve fitting blocks. Then based on these distributions, random sample generation occurs (sample generation blocks). After the model is applied to the generated samples, the produced output values are processed to estimate and extract the distribution attributes [60].

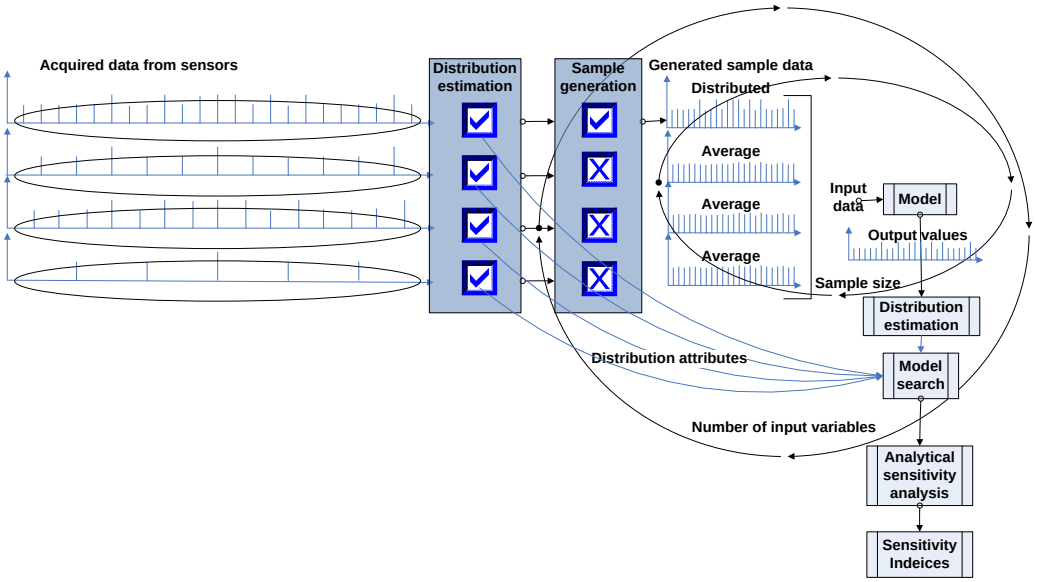

Figure 15 Iteration view of Monte Carlo method for sensitivity analysis

An important issue with the use of Monte Carlo methods in time-sensitive applications is the effort required to estimate the distribution of the input variables prior to sample generation.

For sensitivity analysis purposes based on Monte Carlo sampling method, in order to infer the impact of each input variable on the output variable, data samples of only one input variable (the checked box in Figure 15) are generated at a time while the other input variables are set to a fixed, for example average , value (the cross-marked boxes in Figure 15). This cycle repeats itself for each input variable. Reference [40] called this feature a 'double-loop nested sampling procedure' which could be computationally very expensive, particularly with a high dimension for the input variables.

4.1.4.2.Morris method

In the Morris method, a parameter screening method [19] changes in the value of output variable and is measured for each change in the input variable. Changes to only one input variable are applied to calculate values of an elementary effect [34]. The resulting set is then processed to estimate the distribution. This implies that for each cycle of the output distribution, the estimation takes $M = 2rn$ model executions if r is the number of required output values for an estimation of a stable distribution and n the number input variables [19]. Even though more economical extensions of the Morris method could reduce the total number of cycles, for example by using each generated model

output in more than one calculation [10], a typically low value for M is as high as 21000 executions 1000 output values and 20 inputs applied to $M = r(n + 1)$ in an improved Morris method [19]. The Morris method thus cannot support sensitivity analysis in time-constrained applications.

4.1.4.3.Analysis of variance (ANOVA) methods

One-At-a-Time (OAT) identification of the dependency of the output variables to inputs does not support the detection of higher order interactions between multiple input variables (i.e. second order and higher) and outputs. To overcome this issue, a series of sensitivity analysis methods are used to measure and decompose the variance of output distribution to some elements each of which could separately represent these input-output interactions [58]. ANOVA based sensitivity analysis methods are in general computationally more efficient for this reason [56].

Variance decomposition is defined as follows [61]:

$$V(y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \cdots + V_{12\ldots n} \qquad (3)$$

in which, the left hand side shows the total variance of model output y and the right hand side is a sequence of summation terms of the first order influences of input variables, second order influences, and so on. $V_{12\ldots k}$ represents the portion of variance of the output for interaction of all n input variables together. Based on this variance decomposition, a sensitivity index for the output with respect to each input variable is defined as:

$$S_i = {}^{V_i}/_{V(y)} \qquad (4)$$

To decompose the elements with respect to equation 3 and from there the sensitivity indices, when no explicit relationship exists between inputs and output (i.e. when an analytical approach is not possible), a numerical approach based upon sample generation (e.g., Monte Carlo) could be adopted [9]. The amount of computational overhead, in terms of the number of model runs (for producing output values per each input sample set) can be derived using the following equation:

$$M = N \times \sum_{i=0}^{n} \frac{n!}{(n-1)!i!} \qquad (5)$$

Where N is the sample size, and n is the number of input variables. For example, with 10 input variables and 1000 samples, the number of model executions would make 1024000 which is high. Thus, the computational cost for time-constrained applications needs to be improved. The following subsections cover this improvement and the overall drawback of all sampling based methods.

4.1.4.4.Fourier amplitude sensitivity test (FAST)

Fourier Amplitude Sensitivity Test (FAST) [18] and its extended version [69] are examples of improvements in computational efficiency for ANOVA-based sensitivity analysis methods. FAST (and extended FAST) can be distinguished from other ANOVA methods by means of its input data sample generation scheme, in which, samples for each input variable are generated according to a periodic function within the limits of the input variable [19]. In other words, in a FAST method, the data distribution of input variables cannot be estimated from the acquired historical data. Instead, all distributions of input variables are considered to be uniform and within a range which should be specified. The subsequent generated samples on this range follow a periodical function [59]. The periodic nature of a sample generation scheme causes the model output values to be periodic. Therefore, by using a numerical Fourier analysis on the values of output, the magnitude of Fourier spectrum at each frequency represents the sensitivity index of the corresponding input variable. Components of this process are shown in Figure 16.

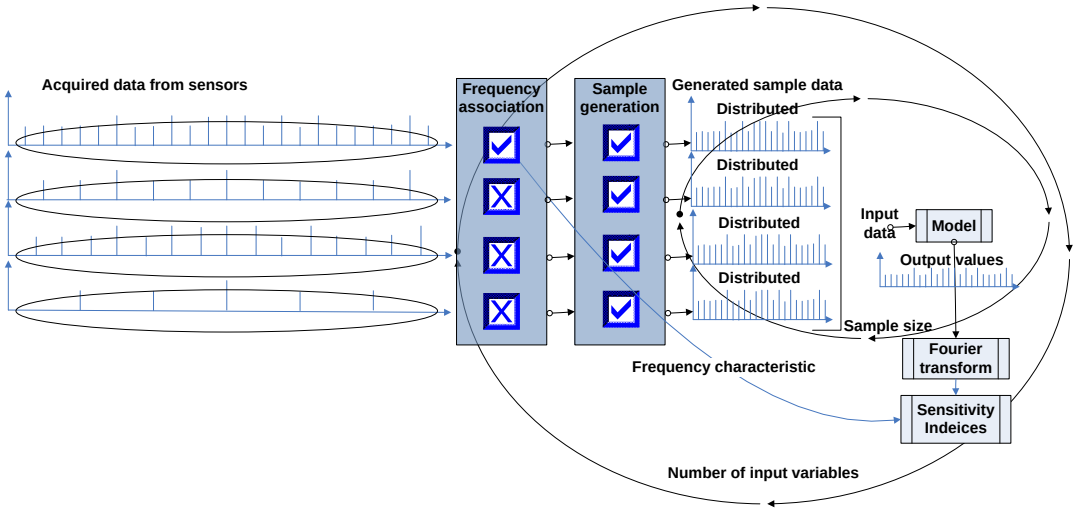

Figure 16 Iteration view of the use of a FAST method for sensitivity analysis

As shown in Figure 16, some aspects of computational cost that existed in the Monte Carlo method, i.e. distribution estimation, are omitted in the FAST method and replaced with simple tasks for boundary detection and frequency association. Furthermore, the output value distribution estimation is also replaced by a numerical Fourier Transform (FT) method for finding Fourier spectrum. In order to explicitly identify the power coefficient associated with the frequency of each input variable, a proper choice of the unique frequencies is required. For this, the range of frequencies is divided into high and low ranges. A high frequency is assigned to the input variable subject to power spectrum coefficient identification and the rest of input variables are assigned a frequency from the low range. This way the distance between the high frequency and all other low frequencies on the spectrum allows clear identification of the coefficient, or sensitivity index. In Figure 16, the checked box in the frequency association module shows that the input variable number 1 is assigned a very different (high) frequency for the sample generation compared to the frequency of the others (those with crossed boxes). As a result the power coefficient of the first frequency can be detected with sufficient confidence.

This type of frequency association adds a new loop for the sample generation and model execution process to the analysis. The number of model runs using FAST is obtained as follows:

$$M = Nn(8w_{max} + 1) \qquad (7)$$

where N is the sample size, n the number of input variables and w_{max} the largest among the assigned frequencies [59].

The number of model executions in FAST does not seem low comparing to the alternative sampling-based SA methods, as it still features the 'double-loop nested sampling procedure' according to [40]. However, the computational overhead could be lower due to the simpler tasks included in the nested loops. The sample generation and Fourier transform in FAST is usually less computationally costly than the collection of sample generation, distribution estimation, and distribution-based function fitting (model search).

4.1.4.5.Time uncertainty in generated sample data

A major drawback of sampling-based SA methods is in the concurrency of use of generated sample data by the model. This is not always the case in the use of real data in a system where all data values may not be generated and used in the model at the same time. For example, a thermostat generates an input event once the temperature passes a certain threshold. The input event to warning dissemination system then generates an alarm output. In the same system, a water level-gauge generates a trigger input event if a water tanks is filled up above a certain level. The two event inputs are not guaranteed to be simultaneous. The simultaneous generation of values of data and their connection to the model to produce the corresponding next output value is a rare event in systems with stochastic events and non-deterministic responses.

As another example, consider the application of stochastic simulation for statistical analysis and characterization of Resin Transfer Molding processes [45]. When calculating the mould pressure profile and resin advancement progress, they consider the measurement of time as a source of uncertainty with respect to measurements of viscosity, pressure, displacement, surface density, compressing variation, stacking sequence, race tracking, and human error.

In sample generation for measured data series, only magnitude of input variables is concerned to

follow a fit distribution. The timestamp when each new data is generated obeys a fixed frequency. When no value of a data series has actually entered the system, a data entity of zero or a low value is generated based on the fitted probability distribution. This concept is shown in Figure 18.

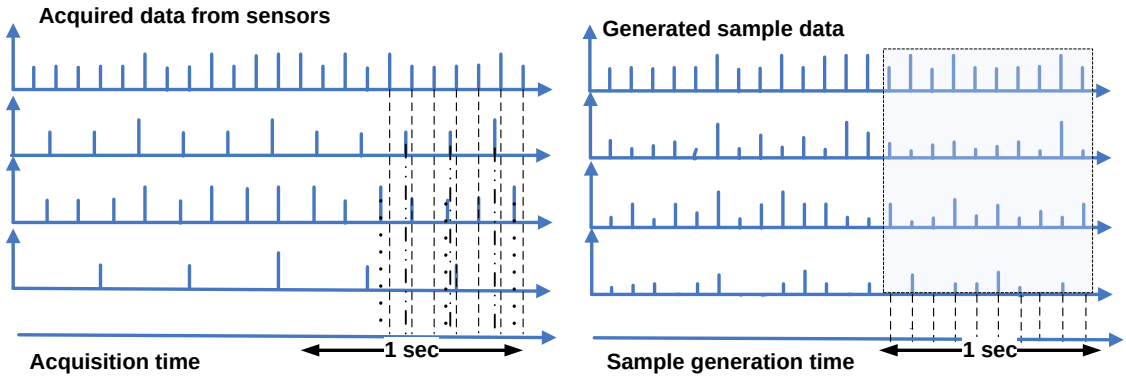

Figure 18 Sample generation may generate more data entities than the original distribution

In the example shown in Figure 18, four data time series from four sensors are acquired with sampling rates of 10, 4, 6, and 2 samples per second. Therefore, in one second, 22 data entities enter the system. However, sample generation with generation rate of 10 samples per second, generates 40 data entities in one second. Hence, it seems very likely that a portion of computational overhead of sampling-based sensitivity analysis methods results from the generation of extra data. To avoid this, one may consider the randomness or frequency of the data entering the system, and simulating it by applying the realized frequency or random function to the rate of sample generation. This additional stochastic modeling effort [51] has not been reported in sampling-based sensitivity analysis methods. Even though if applied, it adds up to the computational efforts of distribution estimation and sample generation tasks, leaving a the trade-off between the added computational overhead and the reduction of generated sample data.

4.1.4.6.Entropy-based epistemic sensitivity analysis

[39] and [40] tackled the issue of computational cost of the ‘double loop sample generation strategy’ and restrictive conditions of evaluation of dependent variables based upon independent

variables in the sampling-based SA methods by proposing an approximation approach that measures the entropy of variable distributions from the original samples. The method used the same decomposition equation as in equation 3 in section 4.1.4.3 but uses entropy instead of the variance of sample data distributions.

The method avoids the time consuming sample generation of independent variable and the evaluation of dependent variable by 'Simple-Random Sampling (SRS)' using piecewise uniform density function estimations. The approach is shown in Figure 19.

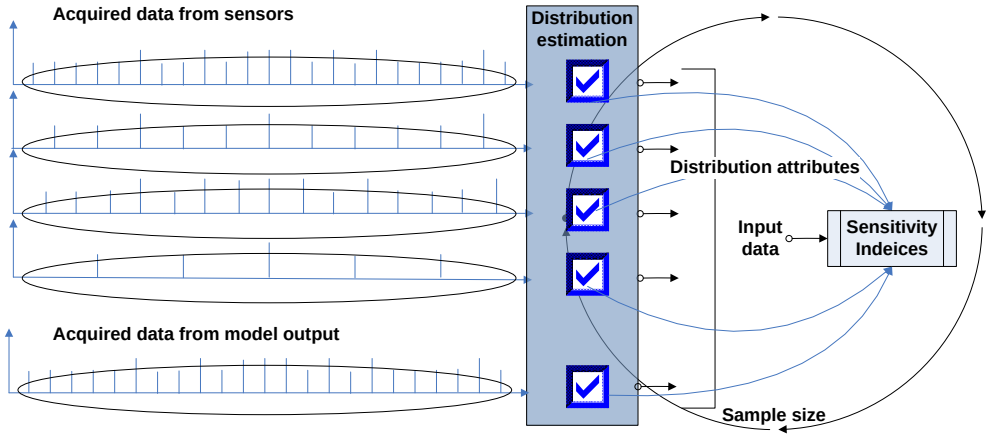

Figure 19 Iteration view of Entropy-based method for sensitivity analysis

Only one sample size execution is sufficient to obtain the samples and to approximate the sensitivity indices (see Figure 19). [40] demonstrated the feasibility of the estimation approach in a test case with fifteen independent and two dependent variables. Reasonable results were provided with far less computational cost. However, obtaining the appropriate indicator functions for each independent variable requires knowledge of their distribution probabilities [40].

4.2 The Statistical distribution of input variables

The second factor in the selection of sensitivity analysis method involves the characteristics of the data distributions of the input and output variables. The sensitivity indices are normally influenced by the distribution of the corresponding data series. For example, nonlinear relationships between

input and output series of a model cannot be recognized by correlation-based sensitivity analysis methods [3]. Variance-based and Entropy-based indices are expected to be more sensitive to heteroscedastic data [39], whilst the homoscedasticity of a data series can be higher among discrete signals and much higher between binary signals.

In this paper, assumptions concerning the generality of type and characteristics of data, discourage the use of such SA methods, e.g., correlation-base SA cannot be considered as a suitable approach. However, variance-based or entropy-based SA methods look promising from the point of view of data heterogeneity.

4.3 The Computational overhead of SA methods

Sensitivity analysis is a computation-hungry process. In domain-wide (global) sensitivity analysis methods, large batches of input variables are captured at each time interval and levels of sensitivity are measured based on historical data analysis. For example, sampling-based methods need to generate new and rather large sizes of sample values of both output and input data regardless of the original sample sizes.

The amount of resources needed by the SA algorithm and its associated data can be compared with the amount of savings that may occur as a result of the applied algorithm. Correlation-based methods [3] need equal sizes of data batches for the input and output series of the model. Therefore, the sampled data series needs to use either interpolation or extrapolation to maintain an equal size, subsequently adding an extra computational load onto the system. ANOVA based SA methods save the computational effort on the analytical analysis side, but instead contribute to the computational cost via the sample generation efforts.

According to section 4.1, FAST performs better than other ANOVA-based SA methods because it is independent of the detection of input variable distribution characteristics. [56] argues that FAST perform is the most efficient of all SA methods. However, having investigated the ability of several global SA methods, including Morris, correlation, and FAST, they also stress that none of the

existing SA methods maintain their competency with an increasing number of input variables.

It is obvious that there is wide scope for exploring methodologies for sensitivity analysis that could perform as efficiently for use in time-constrained applications.

5. EventTracker Sensitivity Analysis

It is shown in this section that the event-based sensitivity analysis method (EventTracker) proposed by [65] is feasible to be used for variable selection for time-constrained applications. The EventTracker SA method is applied in this paper to the sensory system in a well-drilling system.

EventTracker is an input variable selection method that assembles together information about the trace of the changes in system inputs and outputs using sensitivity analysis. For this to happen, an input/output trigger/event matrix is produced for all inputs and outputs of the system. This matrix is designed to map the relationships between input data as causes that trigger events and the output data that describes the actual events. The cause-effect relationship between the causes of state change i.e. input variables and the effect system output parameters determines which set of inputs have a genuine impact on a given output. In this way the 'EventTracker' method is able to construct a discrete event framework where events are loosely coupled with respect to their triggers for the purpose of sensitivity analysis. The method has a clear advantage over analytical and computational IVS method since it tries to understand and interpret system state change in the shortest possible time with minimum computational overhead. The process of input variable organization for knowledge creation on performance measures (system output) is explained in [65].

5.1 Experiment on Drilling Disaster Knowledge Support

During the drilling of a well, the driller needs continuous knowledge about the borehole stability and even more urgently about any significant borehole instability. The actual state of the borehole is used for any immediate, eventually urgent, counteractions, or to make any revisions to improve the drilling plan respectively. The state is typically evaluated using a number of different information

sources and that incorporates a geological model. Such information sources usually supply real-time data, measurements at regular intervals and drilling reports. Real-time data originates typically from sensors mounted at the surface as well as in some special cases from down hole; these sensor inputs are introduced in Table 2 in the Appendix. The data are sampled and provided with an interval in the magnitude of seconds. Figure 22 shows a portion of such data together with the associated information about the state of the drilling rig.

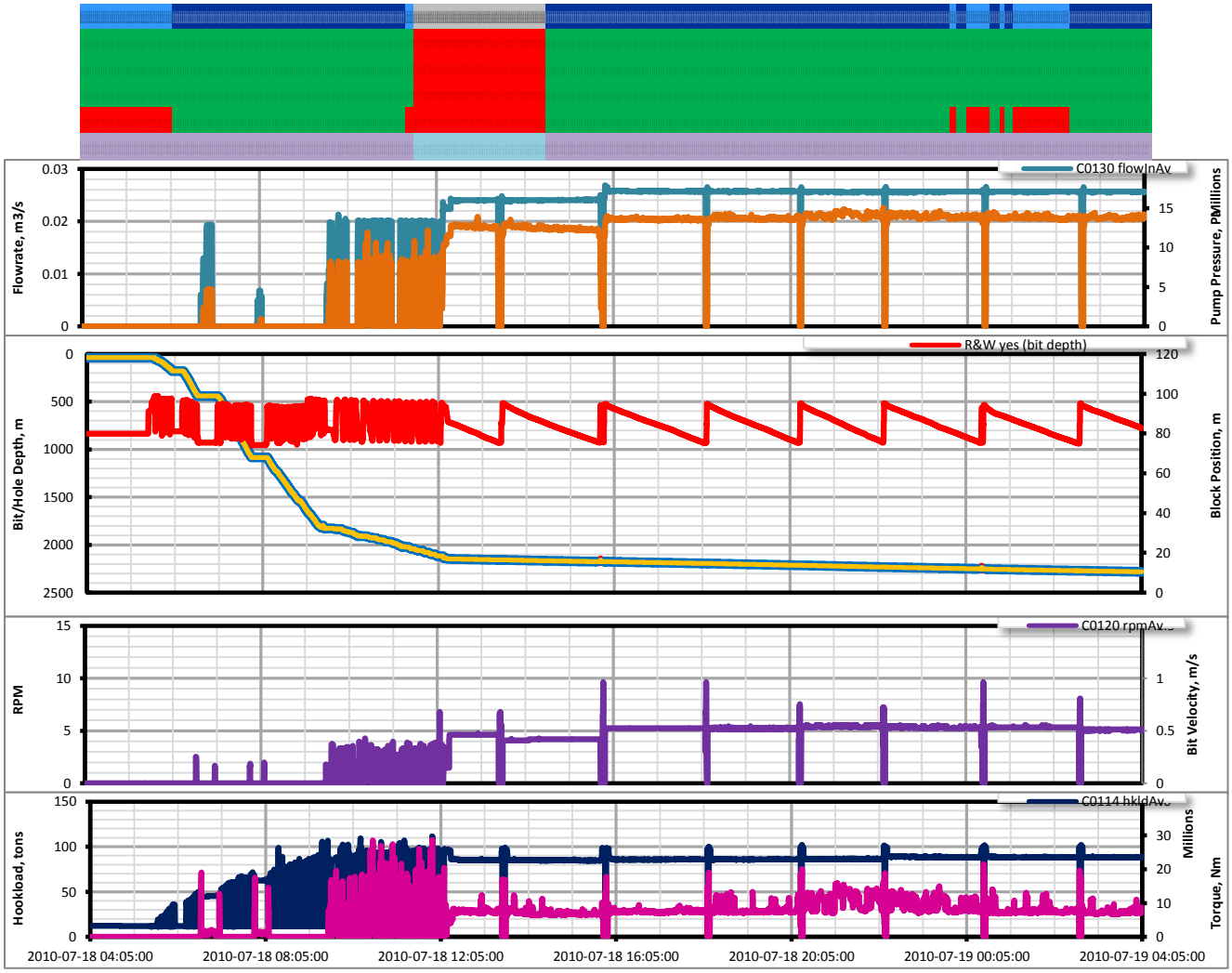

Figure 22 Sensor data time series (four lower charts) for drilling the state (upmost chart) in a rig, sampled at 0.1 Hz. The non-numerical and colour-coded drilling states include one OperationCode (CodeOS) and five other atomic codes

5.2 Time constraint

Each decision support function has to meet response time requirements which are not available per

se, but as an estimate the following response times in Table 1 could be considered;

Table 1 Required Response Time in Drilling Disasters

Event	Response time
Kick	magnitude of minutes
Stuck pipe support	magnitude of seconds
Pump start up	magnitude of seconds
Lost circulation	magnitude of seconds
Deviation from normal situation	magnitude of seconds

The required response time depends mainly on the critical event to be detected. The earlier a counteraction is initialized; the better is the chance to manage an emerging crisis. Thus the question for the response time is not permissible per se, answering 'the response time should be as short as possible' is of course insufficient. Nevertheless it can be stated that the overall magnitude of a response time is of the order of seconds. Especially for a kick event the response time is strongly dependent upon the formation parameters such as permeability, pressure conditions and fluid properties and thus the required response time may be extended to an order of minutes. On the other hand, for stuck pipe prevention, ream and wash operations are usually applied after connecting a new stand to the drill string. Thus, waiting for a decision which either recommends or not recommends such an operation makes no sense if the response time is the magnitude of the duration of that operation.

5.3 Variable construction

In terms of real world tasks, stuck pipe detection is an actual but rather simple problem; the real and more sophisticated task is the stuck pipe prediction and as a consequence stuck pipe prevention by some precautionary counteractions. The assumption that hook-load and torque contain information about emerging stuck pipe is still valid, how that information is covered is actually unknown and therefore a challenge whose importance should not be underestimated.

Figure ? sketches a typical borehole drilled nearly vertical at its beginning and that then changes direction to nearly horizontal. The main components of the hook load F_{Hook} are the acceleration

force F_A, the weight of the whole drill string F_W, the friction forces F_F and some other non-quantifiable forces denoted as ε. In case of creating a deterministic model, the mass influx of the drill string, the borehole trajectory, especially the inclination and friction factors, amongst others, need to be known.

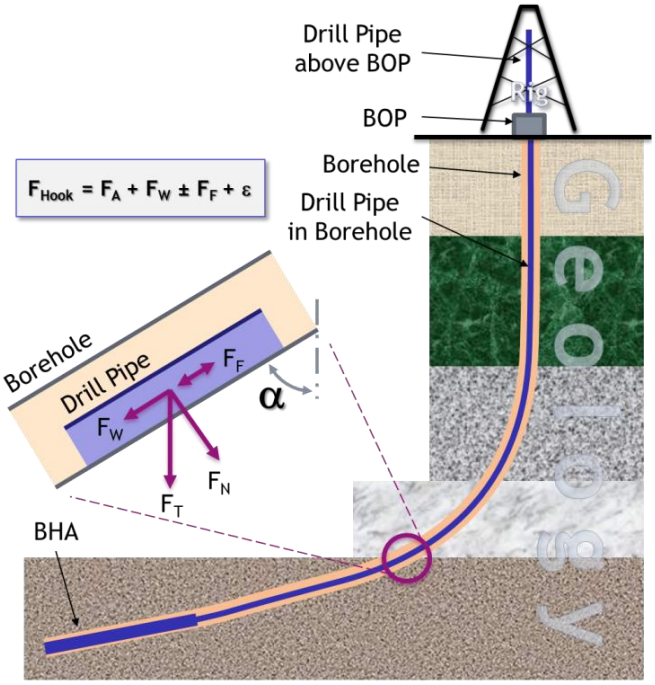

Figure 1 Forces influencing the hook load

Since it appears unpromising to identify and estimate all factors with reasonable certainty and accuracy to predict and thus prevent stuck pipe (as well as other crises), a heuristic approach incorporating deterministic know-how seems to be the best feasible solution statement. To incorporate as much deterministic knowledge as possible, a systematic approach in building variables based on some laws of physics appears to be an appropriate solution here. Consider the case for each of the ten sensors installed in each well, that 100 variables are constructed. Figure 23 shows a portion of such variables generated for one of the sensor data sources for duration of 10 minutes with a 0.1 Hz generation rate.

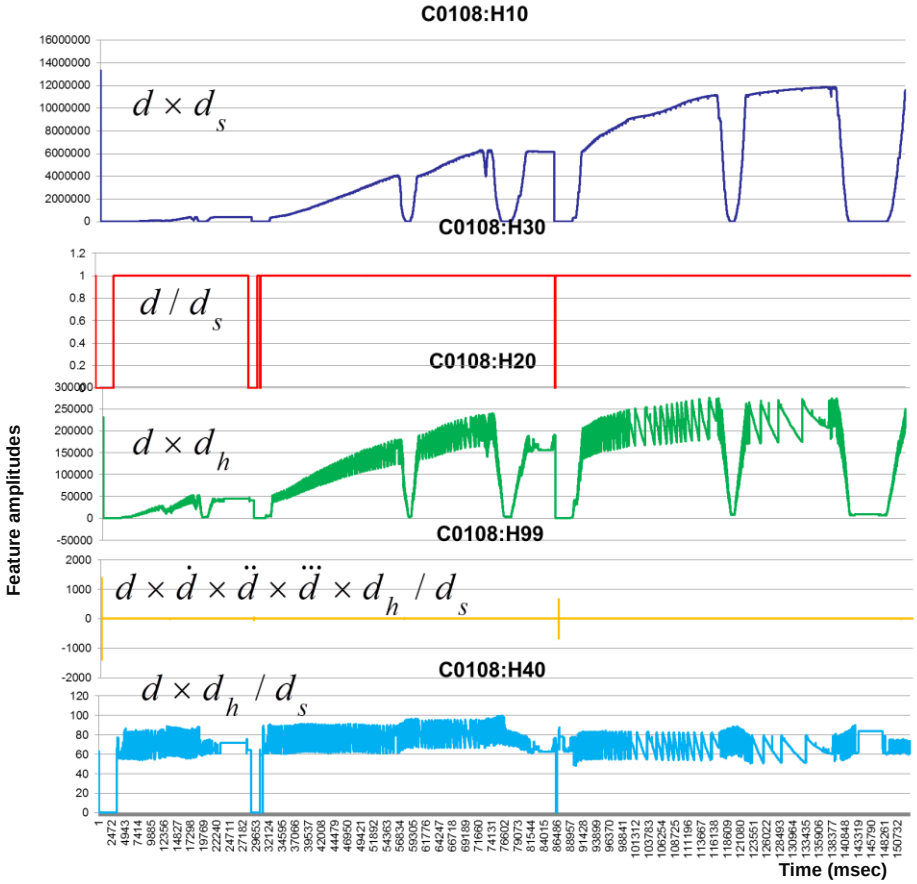

Figure 23 Some of the constructed input variables based on sensor inputs from sensor C0108 on the drilling rig. The symbol *d* denotes deviation function

5.4 Experiment data and results

Time series from these 100 variable data sources were collected for each of the ten sensors in each of the four drilling wells. Also the generated six system states of each of the four drilling processes that provided the Instantaneous State of the drilling scenario were collected. EventTracker SA method was executed on the collected samples and created a matrix of sensitivity indexes to associate each output to each of the 100 input time series for each of the ten sensors. A similar task was performed for a 100 variables for 10 sensors in the other three rigs.

Figure 24 shows the sensitivity indexes of the Operation Code (CodeOS) system state with respect to all 100 variables in each of the four well datasets. Correlations in the chart mean that some variables have a less significant role than some others in determining the state of the well.

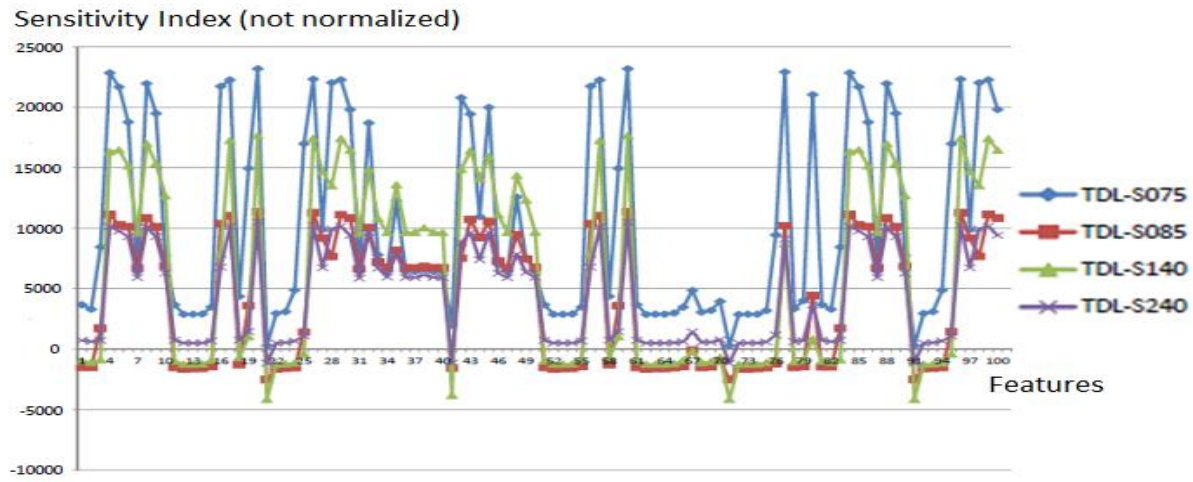

Figure 24 Sensitivity indexes of state of the well with respect to the 100 variables in four wells

By grouping and sorting the normalized sensitivity indexes of all Operation Codes with respect to all variables for each sensor data in separate diagrams, we obtain significance plots as shown in Figure 25. Obviously the order of the 100 variables in these plots are not the same as the one in figure 24 as variables are sorted based on their normalized sensitivity index. These plots indicate that large proportion of variables has low normalized sensitivity index. This could be interpreted as not important for none of the Operation Codes with respect to the associated sensor data. Eventually, if a significant threshold of for example 0.45 is considered (dashed horizontal line in figure 25), by looking at figure 25 and similar results for the other variables in the Appendix, it could be realized that more than 50% of the generated knowledge could be ignored, i.e. not generated.

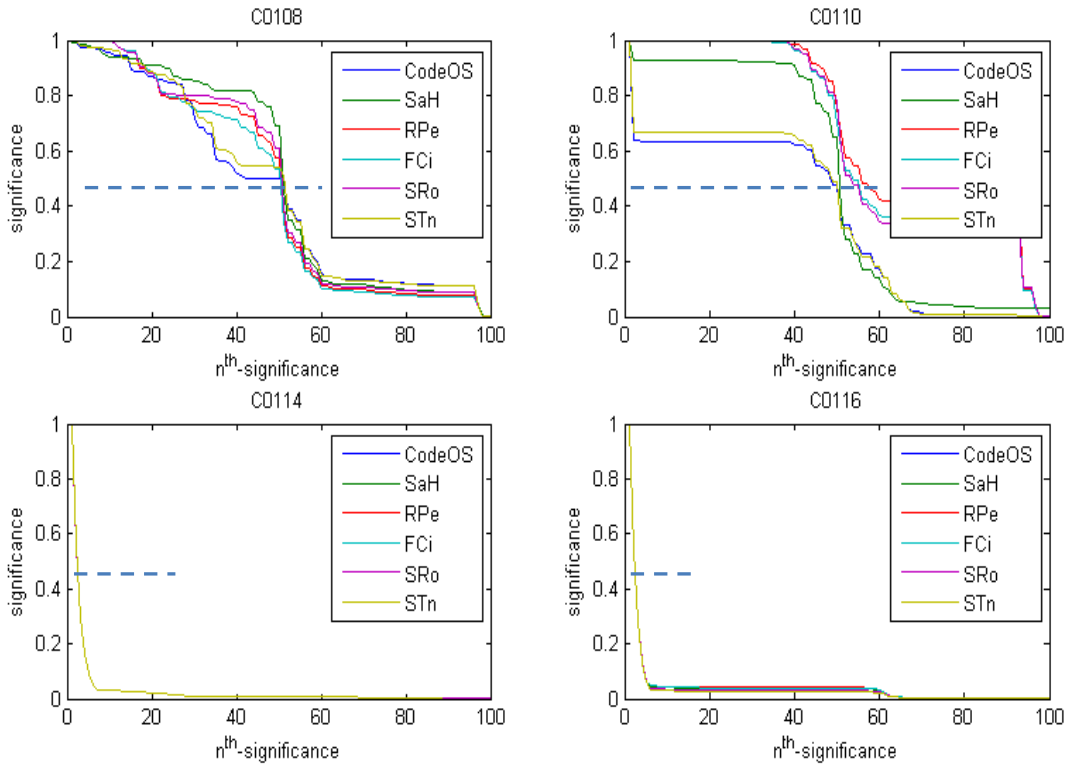

Figure 25 Normalized sensitivity indexes representing 'significance' value of variables for all Operation Codes with respect to each sensor data. Plots are sorted against significance values. The plots show only the significance of variables generated from sensor data C108, C110, C114, and C116. Similar plots for the variables from the other sensors of the drilling system are shown in the Appendix. Dashed horizontal line shows the 0.45 significance threshold level.

In terms of time efficiency, the execution of the proposed SA method took a few seconds in comparison with the sampling window (10 minutes). Each scoring search slot took a fraction of a second making it feasible for time-constrained applications that generate data with order of seconds or higher to use.

$$T_{Score_i} \leq \frac{(Search\ Slot)}{N_{Trigger} N_{Event}} \qquad (17)$$

$$\boldsymbol{T_{Score_i} \leq 160\ \text{msec}, where\ N_{Trigger} = 100, N_{Event} = 6, SS = 100\text{sec}} \qquad (18)$$

6. CONCLUSION

The opportunity to bring new light to the problem of input variable selection (IVS) has been explored

and its feasibility for use in time-constrained dimensionality reduction applications has been demonstrated. In this paper the authors attempt to highlight the importance of chosing the appropriate Input Variable Selection method with respect to the type of applications. The applications we focused on are industrial situations in which the accuracy and timeliness of input data is vital for the integrity of equipment and safety of operators. A number of well-established IVS methodologies (i.e. Heuristics, Optimization and Simulation, Statistics, Machine learning, Data mining, Classification / knowledge-discovery) were analyzed and compared with the proposed EventTarcker method. them from perspectives that have not been explored with the presented spectrum i.e.. The analysis and comparison conducted in this paper enables the reader to choose the appropriate method with respect to the application in hand. From reviewing the related work, we conclude that based upon their effect on the structure of the original set of input variables, IVS problems can be classified into two distinct groups of methods: primary or original variable methods (OVIVS), and secondary or derived variable methods (DVIVS). The major distinction between OVIVS and DVIVS lies with keeping the variables as they are and only deciding on their relevancy, or transforming the system input variables into new subsets of. The main shortcoming of DVIVS concerns data storage and processing of the new variables. These issues contribute to the computational overhead of DVIVS methods which makes them less attractive. It was concluded that it would be difficult to find an IVS method that could fulfill promptness, accuracy, and computational efficiency of time-critical and resource-limited applications.

Sensitivity analysis (SA) methods were investigated due to their intention to measure and support selection criteria for input variables based on their influence on system outputs. Moreover, a framework was introduced to analyze and evaluate the computational complexity and efficiency of SA methods. There is as yet no sensitivity analysis method that competently works with a complex system in terms of the heterogeneity and large number of input variables and time constraints. Simpler methodologies that do not compromise speed of dealing with historical data still pose a

challenge to existing SA methods. Implementation of EventTracker sensitivity analysis has supported this statement. The paper presented a solution i.e. EventTracker SA within the presented scope is able to handle time-critical dimensionality reduction problems with respect to limited computational resources. The insutrial case study presented in this paper is event-based, and the proposed SA method was demonstrated to be suitable for such industrial cases whilst other methods fell short of suitability tests.

ACKNOWLEDGMENT

The authors acknowledge the financial and continued support provided by the Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom. The pilot case drilling system data in this work has been taken in the context of, and supported by, the EU FP7 funded project TRIDEC (FP7-258723-TRIDEC).

APPENDIX

Table 2 Sensor data inputs in a rig drilling scenario

Input ID	**Unit**	**Description**
C0108	m	Total (measured) depth of bit
C0110	m	Total (measured) depth of hole
C0112	m	Block position
C0113	m/s	Drill Rate
C0114	kg	Hookload, measured at surface
C0116	kg	Weight on bit, measured at surface
C0118	J	Rotary torque, measured at surface
C0120	rad/s	Rotary speed, measured at surface (revs per minute)
C0121	Pa	Pump (standpipe) pressure, measured at surface
C0130	m3/s	Mud flow into the hole
D0101	m	mdHole - mdBit

D0201	m	mdHole + posBlock
D0301	m	mdBit + posBlock
E0101	W	tqAv * rpmAv
E0201	W	pressPumpAv * flowInAv
E0301	W	ropAv * wobAv

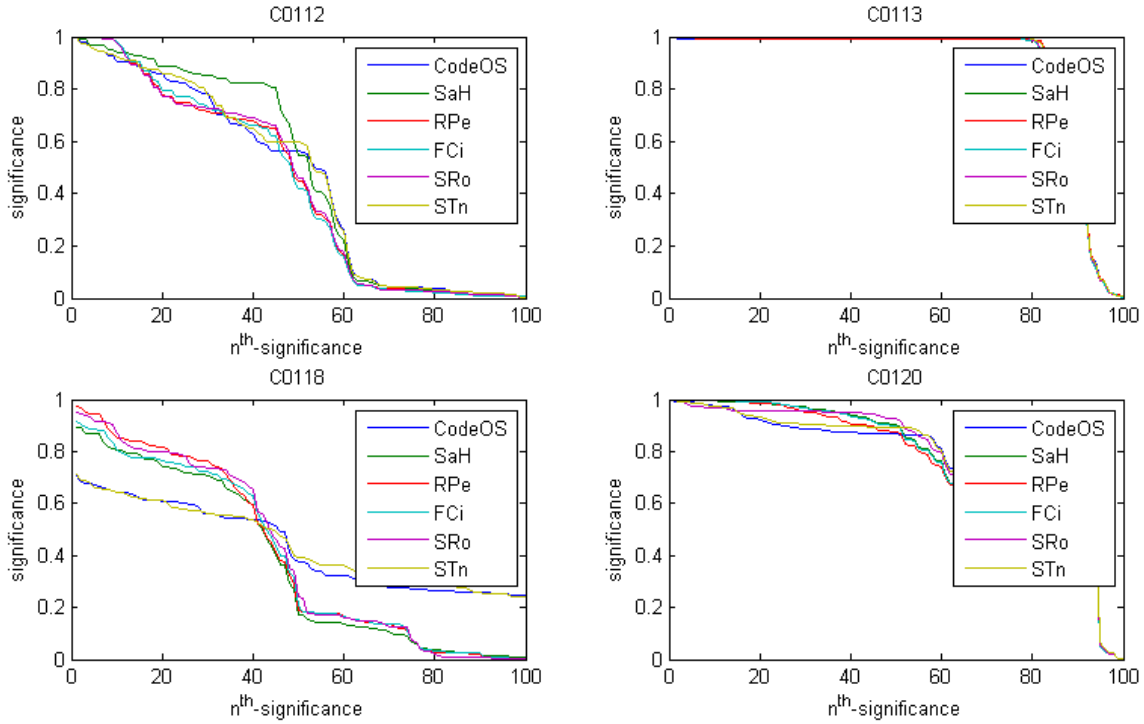

Figure 26 Sorted input significance plots of variables generated from sensor data C112, C113, C118, and C120

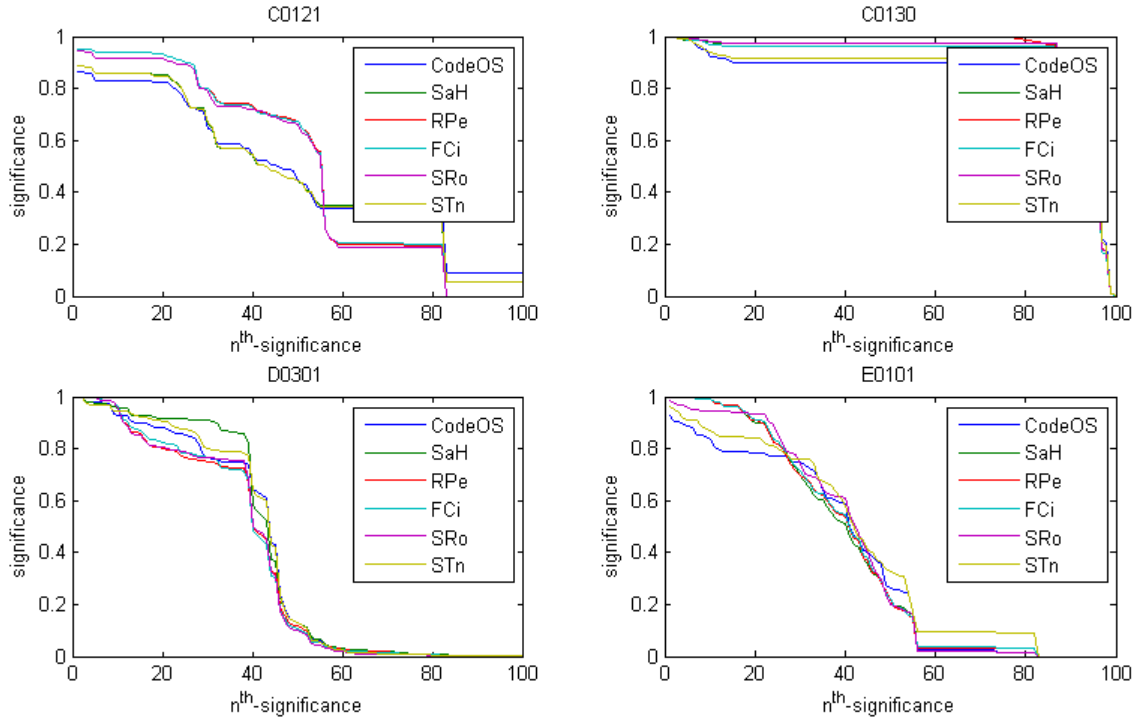

Figure 27 Sorted input significance plots of variables generated from sensor data C121, C130, D0301, and E0101

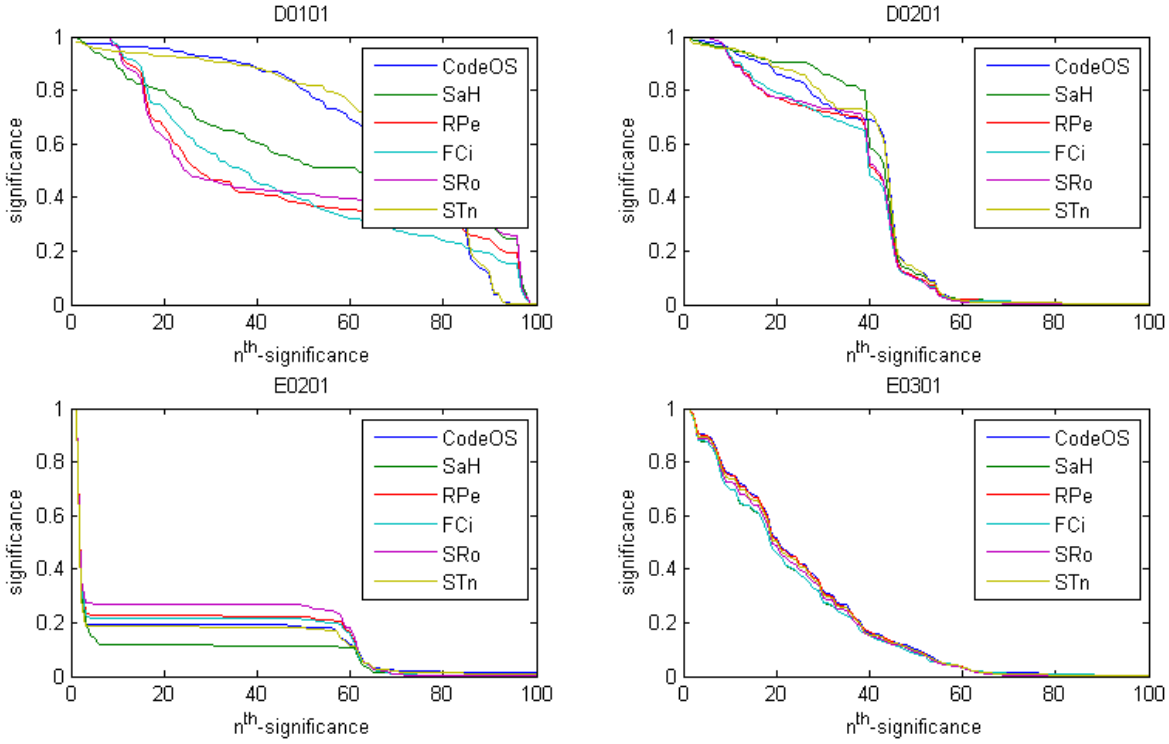

Figure 28 Sorted input significance plots of variables generated from sensor data D0101, D0201, E0201, and E0301

REFERENCES

[1] M. Acheroy, Introduction to Data Fusion, [Homepage of Signal and Image Centre (SIC), Electrical Engineering Department of the Faculty of Applied Sciences of the Royal Military Academy], [Online]. Available: http://www.sic.rma.ac.be/Research/Fusion/Intro/ 27/01/1999-last update.

[2] A., Ambrosetti, A. Malchiodi, Nonlinear analysis and semilinear elliptic problems, Cambridge University Press, Cambridge, UK, 1997.

[3] C. Annis, Correlation, [Homepage of statisticalengineering.com], [Online]. Available: http://www.statisticalengineering.com/correlation.htm, 01/01/2008-last update.

[4] R.G. Askin, C.R. Standridge, Modeling and analysis of manufacturing systems, Wiley, New York, 1993.

[5] J. Banks, Discrete-Event System Simulation, 3rd ed., Prentice Hall, Upper Saddle River, NJ, USA, 2001.

[6] D.L. Banks, R.T. Olszewski, R.A. Maxion, Comparing methods for multivariate nonparametric regression, Communications in Statistics Part B: Simulation and Computation, vol. 32, no. 2, pp. 541-571, 2003.

[7] G. Beylkin, C. Kurcz, L. Monzón, Fast algorithms for Helmholtz Green's functions, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 464, no. 2100, pp. 3301-3326, 2008.

[8] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence, vol. 97, no. 1-2, pp. 271-275, 1997.

[9] E. Borgonovo, L. Peccati, Global sensitivity analysis in inventory management, International Journal of Production Economics, vol. 108, no. 1-2, pp. 302-313, 2007.

[10] R.D. Braddock, S.Y. Schreider, Application of the Morris algorithm for sensitivity analysis of the REALM model for the Goulburn irrigation system, Environmental Modeling and Assessment, vol. 11, no. 4, pp. 297-313, 2006.

[11] K.H., Brodersen, F. Haiss, C.S. Ong, F. Jung, M. Tittgemeyer, J.M. Buhmann, B. Weber, K.E. Stephan, Model-based feature construction for multivariate decoding, NeuroImage, 2010.

[12] K. Buchenneder, Processing of myoelectric signals by feature selection and dimensionality reduction for the control of powered upper-limb prostheses, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-38449120756&partnerID=40&md5=d0035422173fb3941a3905e9e3e13d65, 2007.

[13] H. Bunke, K. Riesen, Improving vector space embedding of graphs through feature selection algorithms, Pattern Recognition. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-77953078185&partnerID=40&md5=2fe02081a72afabb36b576444fb8e630, 2010.

[14] A. Buonomo, A. Lo Schiavo, Nonlinear distortion analysis via perturbation method, International Journal of Circuit Theory and Applications, vol. 38, no. 5, pp. 515-526, 2010.

[15] S. Charaniya, H. Le, H. Rangwala, K. Mills, K. Johnson, G. Karypis, W. Hu, Mining manufacturing data for discovery of high productivity process characteristics, Journal of Biotechnology, 2010.

[16] F., Chen, F. Li, Combination of feature selection approaches with SVM in credit scoring, Expert Systems with Applications, vol. 37, no. 7, pp. 4902-4909, 2010.

[17] H.L. Cloke, F. Pappenberger, J. Renaud, Multi-Method Global Sensitivity Analysis (MMGSA) for modelling floodplain hydrological processes, Hydrological Processes, vol. 22, no. 11, pp. 1660-1674, 2008.

[18] R.I. Cukier, H.B. Levine, K.E. Shuler, Nonlinear sensitivity analysis of multiparameter model systems, Journal of Computational Physics, vol. 26, no. 1, pp. 1-42, 1978.

[19] D.J.W. De Pauw, K. Steppe, B. De Baets, Unravelling the output uncertainty of a tree water flow and storage model using several global sensitivity analysis methods, Biosystems Engineering, vol. 101, no. 1, pp. 87-99, 2008.

[20] N.S. Dias, M. Kamrunnahar, P.M. Mendes, S.J. Schiff, J.H. Correia, Variable subset selection for brain-computer interface: PCA-based dimensionality reduction and feature selection, Proceedings of the 2nd International Conference on Bio-Inspired Systems and Signal Processing, pp. 35, 2009.

[21] D.G. Duffy, Advanced engineering mathematics with MATLAB, 2nd ed., Chapman & Hall/CRC Press, Boca Raton, FL, 2003.

[22] D.P. Durkee, E.A. Pohl, E.F. Mykytka, Sensitivity analysis of availability estimates to input data characterization using design of experiments, Quality and Reliability Engineering International, vol. 14, no. 5, pp. 311-317, 1998.

[23] F. Faghihi, K. Hadad, Numerical solutions of coupled differential equations and initial values using Maple software, Applied Mathematics and Computation, vol. 155, no. 2, pp. 563-572, 2004.

[24] H. Gao, M.K. Mandal, J. Wan, Classification of hyperspectral image with feature selection and parameter estimation, International Conference on Measuring Technology and Mechatronics Automation, ICMTMA, pp. 783, 2010.

[25] A.E.Gaweda, J.M. Zurada, R. Setiono, Input selection in data-driven fuzzy modeling, IEEE International Conference on Fuzzy Systems, pp. 1251, 2001.

[26] S. Gunasekaran, K. Revathy, Content-based classification and retrieval of wild animal sounds using feature selection algorithm, The 2nd International Conference on Machine Learning and Computing, pp. 272, 2010.

[27] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, The Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[28] D.J.Hand, H. Mannila, P. Smyth, Principles of data mining, MIT Press, London, 2001.

[29] G. He, W. Liao, Feature construction method of combined subdivision surface", Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS, vol. 16, no. 3, pp. 507-512, 2010.

[30] X. Huang, W. Wang, Electroencephalography based feature selection for multi-intelligence activity", 2nd International Workshop on Education Technology and Computer Science, ETCS 2010, pp. 808, 2010.

[31] S.S. Isukapalli, Uncertainty Analysis of Transport-Transformation Models, New Burnswick Rutgers, The State University of New Jersey, New Jersey, 1999.

[32] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, 2010.

[33] G. James, Advanced modern engineering mathematics, 3rd ed., Pearson Prentice Hall, Harlow, 2004.

[34] Y. Jin, H. Yue, M. Brown, Y. Liang, D.B. Kell, Improving data fitting of a signal transduction model by global sensitivity analysis, Proceedings of the American Control Conference, pp. 2708, 2007.

[35] I.T. Joliffe, Principal component analysis", 2nd ed., Springer, 2002.

[36] P. Kang, H. Lee, S. Cho, D. Kim, J. Park, C. Park, S. Doh, A virtual metrology system for semiconductor manufacturing, Expert Systems with Applications, vol. 36, no. 10, pp. 12554-12561, 2009.

[37] E. Kim, S. Lee, K. Kwon, S. Klm, Feature construction scheme for efficient intrusion detection system, Journal of Information Science and Engineering, vol. 26, no. 2, pp. 527-547, 2010.

[38] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence, vol. 97, no. 1-2, pp. 273-3[27], 1997.

[39] B. Krzykacz-Hausmann, Epistemic Sensitivity Analysis Based on the Concept of Entropy, Inter'l Symp. Sensitivity Analysis of Model Output (SAMO2001), pp. 53, 2001.

[40] B. Krzykacz-Hausmann, An approximate sensitivity analysis of results from complex computer models in the presence of epistemic and aleatory uncertainties, Reliability Engineering and System Safety, vol. 91, no. 10-11, pp. 1210-1218, 2006.

[41] C. Kwak, Architecture of a dynamic production controller in CIM enterprise environments, International Journal of Production Research, vol. 48, no. 1, pp. 167-182, 2010.

[42] B. Lallemand, G. Plessis, T. Tison, P. Level, Neumann expansion for fuzzy finite element analysis, Engineering Computations (Swansea, Wales), vol. 16, no. 5, pp. 572-583, 1999.

[43] N. Lavrač, J. Fürnkranz, D. Gamberger, Explicit feature construction and manipulation for covering rule learning algorithms, Studies in Computational Intelligence, Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-74049153719&partnerID=40&md5=f361c4a49bc45c810c35f311957bc100., 2010

[44] V. Lemaire, F. Clérot, An input variable importance definition based on empirical data probability distribution", Studies in Fuzziness and Soft Computing, Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-34047164934&partnerID=40&md5=7bd07bcd90ae8b4c8a078f6969716f16, 2006.

[45] J. Li, C. Zhang, Z. Liang, B. Wang, Stochastic simulation based approach for statistical analysis and characterization of composites manufacturing processes, Journal of Manufacturing Systems, vol. 25, no. 2, pp. 108-121, 2006.

[46] K. Li, Fast computation technique of Green's function and its boundary integrals for multilayered medium structures by using Fourier series expansion and method of images, International Journal of RF and Microwave Computer-Aided Engineering, vol. 8, no. 6, pp. 474-483, 1998.

[47] G.J. McRae, J.W. Tilden, J.H. Seinfeld, Global sensitivity analysis-a computational implementation of the Fourier Amplitude Sensitivity Test (FAST), Computers and Chemical Engineering, vol. 6, no. 1, pp. 15-25, 1982.

[48] B.G. Mirkin, Clustering for data mining: a data recovery approach, Chapman & Hall/CRC, Boca Raton, FL, 2005.

[49] D. Mladenić, Feature selection for dimensionality reduction, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-33745840855&partnerID=40&md5=eb339e5bc2b37fe281c7ed9fc517a8f0, 2006.

[50] P. Norvig, S.J. Russell, Artificial Intelligence: Modern Approach, 1st ed., Prentice Hall, 1995.

[51] B.K. Øksendal, Stochastic differential equations: an introduction with applications, 5th ed., Springer, London, 1998.

[52] J.I. Park, S.H. Baek, M.K. Jeong, S.J. Bae, Dual features functional support vector machines for fault detection of rechargeable batteries, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 39, no. 4, pp. 480-485, 2009.

[53] L. Qi, C. Kambhamettu, Y. Jieping, Integrating spatial and discriminant strength for feature selection and linear dimensionality reduction, Conference on Computer Vision and Pattern Recognition Workshops, 2006.

[54] J.R. Quevedo, A. Bahamonde, O. Luaces, A simple and efficient method for variable ranking according to their usefulness for learning, Computational Statistics and Data Analysis, vol. 52, no. 1, pp. 578-595, 2007.

[55] T. Ragg, M. Granzow, Normalization, variable ranking and model selection for high-dimensional genomic data – Design and Implementation of automated analysis strategies", Workshop on State-of-the-art in Scientific Computing PARA'04, Springer, 2004.

[56] J.K. Ravalico, H.R. Maier, G.C. Dandy, J.P. Norton, B.F.W. Crokef, A Comparison of Sensitivity Analysis Techniques for Complex Models, Int'l Cong. Modeling and Simulation MODSIM'05, eds. A. Zerger & R.M. Argent, pp. 2533, 2005.

[57] C.P. Robert, G. Casella, Monte Carlo statistical methods", 2nd ed., Springer, New York, 2004.

[58] A. Saltelli, Sensitivity analysis for importance assessment, Risk Analysis, vol. 22, no. 3, pp. 579-590, 2002.

[59] A. Saltelli, S. Tarantola, K.P. Chan, A quantitative model-independent method for global sensitivity analysis of model output", Technometrics, vol. 41, no. 1, pp. 39-56, 1999.

[60] R.W. Shonkwiler, F. Mendivil, Explorations in Monte Carlo methods, Springer, London, 2009.

[61] I.M. Sobol, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, Mathematics and Computers in Simulation, vol. 55, no. 1-3, pp. 271-280, 2001.

[62] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control", IEEE Transactions on Systems, Man and Cybernetics, vol. 15, no. 1, pp. 116-132, 1985.

[63] L. Talavera, An evaluation of filter and wrapper methods for feature selection in categorical clustering, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-33745202625&partnerID=40&md5=cc0d9fca9e79e5adeb4857661a5ed560, 2005.

[64] S. Tavakoli, A. Mousavi, A. Komashie, A Generic Framework for Real-Time Discrete Event Simulation (DES) Modelling, *Proc. The 2008 Winter Simulation Conference*, Dec. 2008.

[65] S. Tavakoli, A. Mousavi, P. Broomhead, , Event Tracking for Real-Time Unaware Sensitivity Analysis (EventTracker), IEEE Trans. on Knowledge and Data Engineering, Vol. PP, Issue 99, 2011.

[66] A. Unler, A. Murat, A discrete particle swarm optimization method for feature selection in binary classification problems, European Journal of Operational Research, vol. 206, no. 3, pp. 528-539, 2010.

[67] I. Uysal, H.A. Güvenir, An overview of regression techniques for knowledge discovery, Knowledge Engineering Review, vol. 14, no. 4, pp. 319-340, 1999.

[68] Z. Xing, L. Jia, Y. Qin, T. Lei, Research on input variable selection for numeric data based fuzzy modeling", International Conference on Machine Learning and Cybernetics, pp. 2737, 2003.

[69] C. Xu, G.Z. Gertner, A general first-order global sensitivity analysis method", Reliability Engineering and System Safety, vol. 93, no. 7, pp. 1060-1071, 2008.

[70] X. Yang, Green's function and positive solutions for higher-order ODE, Applied Mathematics and Computation, vol. 136, no. 2-3, pp. 379-393, 2003.

[71] S. Zaman, F. Karray, Features selection using fuzzy ESVDF for data dimensionality reduction, Proceedings - 2009 International Conference on Computer Engineering and Technology, ICCET 2009, pp. 81, 2009.

[72] H. Zhang, Y. Hu, Study of feature selection for the stored-grain insects based on artificial immune algorithm, CAR 2010 - 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics, pp. 140, 2010.

[73] Seokho Chi, Sung-Joon Suk, Youngcheol Kang, Stephen P. Mulva, Development of a data mining-based analysis framework for multi-attribute construction project information, Elsevier Advanced Engineering Informatics, Volume 26, Issue 3, August 2012, Pages 574–581.

[74] S. Liu, C.A. McMahon, M.J. Darlington, S.J. Culley, P.J. Wild, 2006, A computational framework for retrieval of document fragments based on decomposition schemes in engineering information management, Elsevier Advanced Engineering Informatics, Volume 20, Issue 4, October 2006, Pages 401–413.

[75] Grant H. Krugera, Albert J. Shiha, Danie G. Hattinghb, Theo I. van Niekerk, 2011, Intelligent machine agent architecture for adaptive control optimization of manufacturing processes, Elsevier Advanced Engineering Informatics, Volume 25, Issue 4, October 2011, Pages 783–796.

[76] Y.H. Hung, 2009, A neural network classifier with rough set-based feature selection to classify multiclass IC package products, Elsevier Advanced Engineering Informatics, Volume 23, Issue 3, July 2009, Pages 348–357.