A data-driven framework for identifying important components in complex systems

Citation for published version:
Lu, X, Baraldi, P & Zio, E 2020, 'A data-driven framework for identifying important components in complex systems', *Reliability Engineering and System Safety*, vol. 204. https://doi.org/10.1016/j.ress.2020.107197

Digital Object Identifier (DOI):
10.1016/j.ress.2020.107197

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Reliability Engineering and System Safety

A Data-Driven Framework for Identifying Important Components in Complex Systems

Xuefei Lu[a], Piero Baraldi[a,*], Enrico Zio[a,b,c]

[a]*Dipartimento di Energia, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy.*
[b]*MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France.*
[c]*Eminant Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea.*

Abstract

Complex technical infrastructures are systems of systems characterized by hierarchical structures, made by thousands of mutually interconnected components performing different functions. Given their complexity, it is difficult to derive their functional logic using traditional risk and reliability analysis methods based on engineering knowledge. In this work, we propose to address the problem in an innovative way that makes use of the large amount of data available from monitoring those systems. Specifically, we develop a data-driven framework to identify the critical components of a complex technical infrastructure. The criticality of a component with respect to the safe/failed state of the infrastructure is assessed considering a feature selection technique which employs Random Forest (RF) classification and a feature importance score. The proposed data-driven framework is applied to a nuclear power plant system and a synthetic case study, which mimics the complexity of a technical infrastructure.

Keywords: Importance measure; Feature selection; Random forest; Complex technical infrastructure; Auxiliary feedwater system

---

1. Introduction

Complex Technical Infrastructures (CTIs) are large-scale systems of systems, consisting of numerous mutually interconnected components. The various CTI systems perform different functions, use technologies from various domains, and are typically designed and built independently [1, 2]. Furthermore, the CTIs are evolving in time due to their continuous development, consolidation and component updating plans [3].

The identification of critical components in a CTI is necessary for improving CTI reliability and availability, while reducing maintenance and operation costs.The traditional approaches for the identification of critical components in risk assessment and reliability analysis are based on the use of

*Corresponding author
Email addresses: `xuefei.lu@polimi.it` (Xuefei Lu), `piero.baraldi@polimi.it` (Piero Baraldi), `enrico.zio@polimi.it` (Enrico Zio)

Importance Measures (IMs), which quantify the contribution of the components to a measure of system performance, such as system reliability, unreliability, unavailability or risk. Section 1.1 reports a review about IMs and their use in risk assessment and reliability analysis. Notice that the computation of IMs requires the knowledge of the functional logic of the system in the form of a structure function, which is typically difficult to acquire for CTIs due to their complexity and continuous transformations.

On the other hand, recent developments in sensors, signal processing and machine learning have opened up opportunities for analysing the large amount of data available to support cost-effective and robust decision-making for design, operation and maintenance [3, 4, 5, 6].

In this context, the objective of the present work is to develop a data-driven framework based on the definition of an importance measure which is computed making use of the operational data monitored on the CTI systems. For this, we consider the problem of classifying the CTI safe/failed state from signal measurements and adopt feature selection techniques.

Feature selection is an active research topic in the data mining, machine learning, and pattern recognition fields [5]. Considering the specific problem of building a classification model based on an inductive learning method, feature selection can have two objectives [7, 8, 5]: (1) identifying all relevant signals influencing the model outcome; (2) selecting a subset of signals, which allows developing the best performing classification model. Section 1.2 reports a literature review on feature selection methods and their applications.

In this work, we consider a feature selection method based on the random forest classification algorithm [9], which focuses on the first objective of ranking and identifying the relevant signals influencing the model output. The method quantifies the importance of a signal as proportional to the amount of uncertainty it allows to reduce when it is used in the classification. The idea behind its use is that the larger the importance of a signal in the classification of the system state, the more critical the component monitored by the corresponding signal.

The rankings of the components' criticalities obtained by the proposed data-driven importance measure are compared to those obtained by the Birnbaum importance measure on some simple systems, which are used to investigate the impact of the number of available data on the robustness of the ranking. Then, the proposed method is validated on two complex systems: an Auxiliary Feedwater (AFW) system of a nuclear power plant [10] and a 50-component system whose behavior is simulated to mimic the complexity of a CTI [11].

1.1. Importance Measures in Risk Assessment and Reliability Analysis

A relevant outcome of risk assessment and reliability analysis of complex technical systems is the quantification of the importance of the component failures with respect to the system performance. Various importance measures have been developed to quantify the criticality of the components from different perspectives. The concept of IM in a coherent system was first proposed by Birnbaum in 1960s

[12]. The Birnbaum IM is defined as the partial derivative of the system reliability with respect to the component reliability. The extension of the Birnbaum IM to non-coherent systems has been studied in [13] and [14]. Other importance measures consider the system (minimal) cut sets, i.e., (minimal) sets of components whose failures lead to system failure. The Fussell-Vesely IM, for example, measures the component importance as the probability that at least one minimal cut set containing the component has failed, conditional on the system failure [15].

Although most of the works on importance measures assess the importance of individual components, the study of the importance of group of components is fundamental for common cause failure analysis [16]. IMs that allow assessing the importance of groups of components include joint reliability importance [17], Reliability Achievement/Reduction Worth (RAW and RRW) [18], and Differential IMs (DIM) [19].

Recent advances bring IMs to broader uses, for example: [20] proposes a dynamic IM for degrading components, which is useful to address the component reassignment problem; [21] proposes a dynamic IM for time-dependent systems; [22] extends IMs to multi-state systems; [23] calibrates IMs to consider system maintenance cost and the economic dependence among components. For more details on IMs, the interested reader can refer to the review works [24, 25, 26, 27].

1.2. Feature Selection Techniques and Applications

Classification problems involve finding a mapping from the input feature space to the output class space, which minimizes the probability of classification error. In general, feature selection techniques can have two different objectives [7, 8, 5]:

(1) identification of all relevant signals (which will be referred to as features) related to the model output;

(2) identification of the (smallest) subset of features, which allows maximizing the classification accuracy of a learning machine.

Feature selection approaches fall into the three categories of wrapper, embedded and filter methods. Wrapper methods select an optimal subset of features using the learning machine itself, i.e., the learning machine is wrapped within the search algorithm, which aims at identifying the feature subset providing the ‘best’ classification performance. For wrappers, the analyst needs to specify the learning machine to be used, the searching engine and the classification performance criteria. Assuming the availability of p features, an exhaustive search among all possible combinations requires the training and testing for performance evaluation of 2^p learning machines, which is an NP-hard problem [28]. Commonly used suboptimal searching strategies include forward selection, which starts with a small number of features and adds features until the learning machine performance is decreased, backward selection, which starts with all features and removes features until the performance is decreased [29], and genetic algorithms [30]. Wrapper methods are intrinsically designed to meet objective (2), whereas their use for objective (1) can have limitations, since they may not select signals correlated to the output, whose information content is redundant with that of other selected signals.

Embedded methods perform feature selection as part of the learning machine training. Although the learning machine to be used needs to be specified by the user, and can have an impact on the feature selection results, embedded methods meet objective (1) since they allow ranking signals using importance indicators obtained during the training procedure, such as the node importance in decision trees [31, 32] and the regression coefficients in the Least Absolute Shrinkage and Selection Operator (LASSO) [33, 34].

Filter methods rank the features according to their statistical association (e.g., mutual information) with the response. Since filter methods are independent from the learning machine, they allow meeting objective (1), although the obtained feature ranking can depend on the choice of the adopted measure of relevance between input signals and output. Commonly used filters include Pearson correlation ratio [35], the FOCUS algorithm [36] and the Relief algorithm [37].

As the feature selection literature is vast and rapidly growing, we refer to the works of [4, 6, 5] for broader views on the subject.

The remainder of this paper is as follows. Section 2 provides the problem setting, the data-driven framework, the proposed data-driven importance measure and the method for critical component identification. Section 3 presents the results of the applications to simple and complex systems. Section 4 summarises the main findings of the work.

2. Methodology

Section 2.1 will illustrate the problem setting, whereas Section 2.2 will describe the proposed data-driven framework.

2.1. Problem Setting

We consider a CTI made by p components C_j, whose degradation to failure process is monitored by signals $X_j \in \mathcal{R}, j = 1 \dots p$. For the sake of notation simplicity, we assume that the j-th signal monitors the j-th component. The set of all monitoring signals is represented by the random vector $\mathbf{X} = (X_1, \dots X_p) \in \mathcal{X}, \mathcal{X} \subseteq \mathcal{R}^p$ and the overall CTI safe(0)/failed(1) state as $Y \in \mathcal{Y}$ with $\mathcal{Y} = \{0, 1\}$. A large amount of data $Data = \{(\mathbf{x}^i, y^i)\}_{i=1}^{n_{total}}$ is collected during the CTI operation, containing the measurements $\mathbf{x}^i = (x_1^i, \dots, x_p^i)$ of the p signals and the corresponding safe(0)/failed(1) states y^i of the CTI at time i. The objective of the present work is to develop a data-driven framework to identify the critical components of the CTI.

2.2. The Proposed Framework

The basic idea behind the proposed framework is to measure the contribution of the components to the CTI safe/failed state by introducing a data-driven IM. To this aim, we consider the use of a feature selection technique for the identification of the subset $\mathbf{X}^\star = (X_{r_1}, \ldots, X_{r_q})$ of all monitoring signals $\mathbf{X}$ relevant to infer the CTI safe/failed state Y.

In this work, embedded feature selection methods are preferred to wrapper methods since they allow identifying all the relevant signals related to the system state, whereas wrapper methods do not select signals – although relevant to the output – whose information content is redundant with that of other already selected signals. Consider, for example, a system whose failure requires the failure of components C_1 and C_2, and in which component C_2 belongs to a safety system activated only in the case of failure of C_1: a wrapper feature selection method would select only monitoring signal X_1 of C_1, which is sufficient to reproduce the system state, whereas an embedded method, similarly to importance measures in reliability and risk analysis, would select both signals X_1 of C_1 and X_2 of C_1.

The proposed data-driven framework for critical component identification is shown in Figure 1. In Step 1, a large amount of $Data = \{(\mathbf{x}^i, y^i)\}_{i=1}^{n_{total}}$ containing the signal measurements $\mathbf{x}^i$ and the corresponding CTI state y^i is collected during the CTI operation. In Step 2, a classification model based on the random forest algorithm is developed using as training data the subset $Data_{train}$ formed by n_{train} instances randomly sampled from $Data$. The model classifies the CTI safe/failed state Y on the basis of the monitoring signals, i.e., $Y = \mathcal{T}(\mathbf{X}; \Theta)$, being Θ the vector of the model parameters. The classification performance of $\mathcal{T}(\cdot; \Theta)$ is evaluated on the subset $Data_{test} = Data \setminus Data_{train}$ and if the classification accuracy is satisfactory, Step 3 is performed, otherwise more data in Step 1 should be collected. Section 2.2.1 will describe the RF classification model. In Step 3, an embedded feature selection method based on RF classification is applied. The obtained importance scores of features $X_j, j = 1 \ldots p$, are taken as the importance measures of the corresponding C_j components, $\mathrm{DDIM}(C_j), j = 1 \ldots p$. The feature selection method, the data-driven importance measure and its theoretical interpretation are discussed in Sections 2.2.2, 2.2.3 and 2.2.4, respectively. Finally, in Step 4, the components are ranked according to the computed data-driven importance measure.

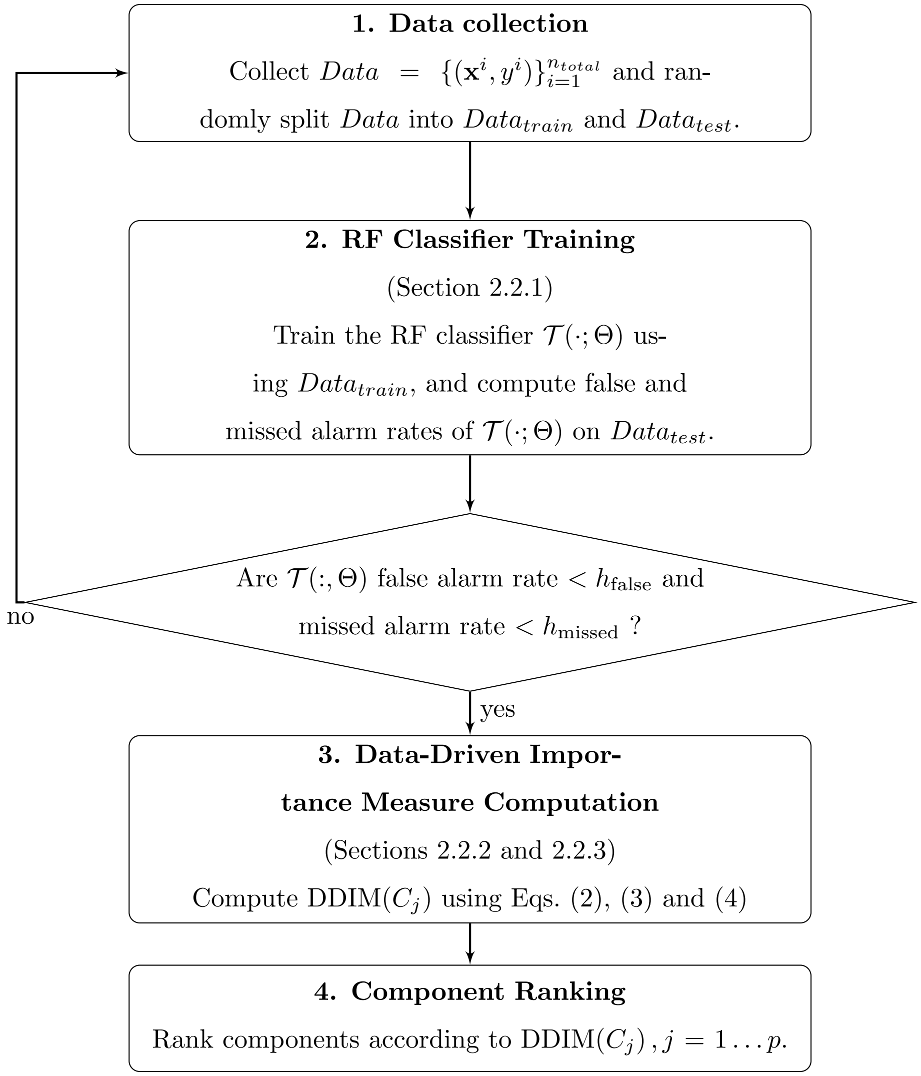


Figure 1: The proposed data-driven framework for critical component identification.

2.2.1. Random Forest Classifier Development

Tree-based models approximate an unknown mapping g via recursive binary partitioning of the feature space into sets of hyperrectangles. The random forest algorithm builds multiple decision trees and merges them to achieve more accurate and stable predictions. A random forest for classification is a collection of n_{tree} randomized classification trees, which are constructed based on the following steps: 1) n_{tree} bootstrap samples $\mathcal{D}_t, t = 1 \ldots n_{tree}$ are drawn from the dataset $Data_{train}$; 2) an unpruned decision tree $\mathcal{T}(\cdot;\Theta_t)$ with parameters Θ_t is grown for each bootstrap sample $\mathcal{D}_t$ by choosing at each node the best splitting feature from a random subset of size m_{try} of all the p features. Once the random forest $\mathcal{T}(\cdot;\Theta = (\Theta_1, \ldots, \Theta_{n_{tree}}))$ is built, the classification is made by aggregating the classification provided by the n_{tree} trees using the majority vote rule. In this work, the RF parameters n_{tree} and m_{try} are set according to the indications in [38, 8, 39]. Specifically, the value of n_{tree} is fixed larger than 100 and $m_{try} = \max\{2, \lfloor\sqrt{p}\rfloor\}$. The classifier $\mathcal{T}(\cdot;\Theta_t)$ is built using the $n_{train} = n_{total}/2$ instances of $Data_{train}$. To evaluate whether the classifier provides a good empirical approximation of the system safe/failed state, out-of-sample classification performances are computed on the $n_{test} = n_{total}/2$ instances of $Data_{test}$. Specifically, we consider two metrics of false and missed alarm rates:

$$\text{False Alarm Rate} = \frac{FP}{FP+TN}, \qquad \text{Missed Alarm Rate} = \frac{FN}{TP+FN}, \tag{1}$$

where TP denotes the number of positive instances ($y^i = 1$) correctly classified, FN the number of positive instances incorrectly classified, FP the number of negative instances ($y^i = 0$), incorrectly classified, and TN the number of negative instances correctly classified.

The setting of the thresholds, h_{false} and h_{missed} (Figure 1) depends on the characteristics of the dataset, such as the total number of instances n_{total} and the unbalanced rate between safe and failed data. In this work, both thresholds have been set equal to 10^{-2} to guarantee that accurate classifiers are used to compute the importance measure.

2.2.2. Random Forest-based Feature Selection

[40] define the importance of a feature X_j as the increase in misclassification error when the values of that feature are permuted while the others remain unchanged. Such feature importance is estimated via an 'Out-Of-Bag' (OOB) procedure, where the misclassification errors are calculated on data not used for growing the tree. Specifically, the misclassification error of each tree, denoted by $\mathcal{E}(\mathcal{T}(\cdot;\Theta_t), \mathcal{D}_t^{OOB})$, is evaluated on the OOB data $\mathcal{D}_t^{OOB}$ not included in the bootstrap sample used for constructing $\mathcal{T}(\cdot;\Theta_t)$; then, the dataset $\mathcal{D}_t^{OOB,j}$ is built by permuting the values of feature X_j in $\mathcal{D}_t^{OOB}$ and the misclassification error $\mathcal{E}(\mathcal{T}(\cdot;\Theta_t), \mathcal{D}_t^{OOB,j})$ of the tree $\mathcal{T}(\cdot;\Theta_t)$ is computed. Notice that feature X_j in $\mathcal{D}_t^{OOB}$ is characterised by the same distribution of $\mathcal{D}_t^{OOB,j}$, although its relation with the output and the other features is completely modified by the permutation. The misclassification error $\mathcal{E}(\mathcal{T}(\cdot;\Theta_t), \mathcal{D})$ of the tree $\mathcal{T}(\cdot;\Theta_t)$ on the dataset $\mathcal{D} = \{(\mathbf{x}^i, y^i)\}_{i=1}^{n_\mathcal{D}}$ is defined by

$$\mathcal{E}\left(\mathcal{T}(\cdot;\Theta_t), \mathcal{D}\right) = \frac{1}{n_\mathcal{D}} \sum_{i:(\mathbf{x}^i,y^i)\in\mathcal{D}} \mathbb{1}\left(y^i \neq \mathcal{T}(\mathbf{x}^i;\Theta_t)\right), \tag{2}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. The importance score of feature X_j is obtained by averaging the differences in OOB errors before and after the permutation of values of signal X_j in the dataset over all the trees fo the random forest:

$$\text{VI}\left(X_j\right) = \frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} \left(\mathcal{E}(\mathcal{T}(\cdot;\Theta_t), \mathcal{D}_t^{OOB,j}) - \mathcal{E}(\mathcal{T}(\cdot;\Theta_t), \mathcal{D}_t^{OOB})\right), j = 1 \dots p. \tag{3}$$

Since repeating multiple times the random permutation of the same feature leads to different datasets, which generates different $\text{VI}\left(X_j\right)$ values, the *bootstratp* technique [41] has been used to compute confidence intervals of $\text{VI}\left(X_j\right)$. The use of permutation schemes for importance score computation has been investigated in [42, 43].

When facing imbalanced datasets, sampling or weighting strategies are typically used to improve the classification accuracy [44]. In case of sampling, $\mathcal{D}_t^{OOB}$ and $\mathcal{D}_t^{OOB,j}$ should be properly modified in Eq. (3) to consider the oversampling of the minority class or the undersampling of the majority class, whereas in case of weighting, the sum in Eq. (3) should become a weighted sum.

Alternative definitions of feature importance scores have been provided by [45, 46] and [39]. In [39], instead of the mean decrease in accuracy, the mean decrease in impurity measured by the Gini importance is used to evaluate the importance of a feature. In [45], the definition of importance score is based on the computation of the error $\mathcal{E}(\mathcal{T}^{*i}(\cdot;\Theta_t), \mathcal{D}_t^{*OOB,j})$ of a new model $\mathcal{T}^{*i}$ built using all the data $\mathcal{D}_t^{OOB}$ except those measured by X_j. This definition requires the development of a dedicated RF model to assess the importance of each component, which can be time-consuming and does not measure how individual models rely on X_j [47]. The various definitions of RF importance scores reflect modellers' difference perspectives on the meaning of feature importance. Reference [42] points out that the RF permutation importance in Eq. (3) is a relatively reliable indicator for categorical features. Since this is the case in this work, we restrict our attention to the permutation-based importance score.

It has also been shown that, although random forest classification does not require the independence among features, correlations among features may induce bias in the evaluation of feature importance [46]. For more discussion on RF-based importance measures, we refer to the works of [43, 48, 46, 49, 47].

2.2.3. Data-Driven Importance Measure Definition

The data-driven importance measure (DDIM) of component C_j is defined equal to the importance score of the corresponding component C_j:

$$\text{DDIM}(C_j) := VI(X_j). \tag{4}$$

2.2.4. DDIM Interpretation

Roughly speaking, the term:

$$\frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} \mathcal{E}(\mathcal{T}(\cdot;\Theta_t), \mathcal{D}_t^{OOB,j})$$

in Eq. (3), can be seen as an in-sample approximation of the expected error $\mathbb{E}[Y \neq \mathcal{T}(\mathbf{X}_j^{'};\Theta)]$, of the RF model using the random vector $\mathbf{X}_j^{'} = (X_1, \ldots, X_j^{'}, \ldots, X_p)$, where $X_j^{'}$ is an independent copy[1] of X_j and the term:

$$\frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} \mathcal{E}(\mathcal{T}(\cdot;\Theta_t), \mathcal{D}_t^{OOB})$$

[1] '$X_j^{'}$ is an independent copy of X_j' means that the distribution of $X_j^{'}$ is the same as the distribution of X_j, and that $X_j^{'}$ and X_j are independent.

is an in-sample approximation of the same model error on the original random vector $\mathbb{E}\left[Y \neq \mathcal{T}\left(\mathbf{X};\Theta\right)\right]$. Therefore, $\mathrm{VI}\left(X_j\right)$ computed using Eq. (3) can be seen as an estimate of the difference, $\mathrm{VI}^*\left(X_j\right)$, of expected errors of the RF classification model:

$$\mathrm{VI}^*\left(X_j\right) = \mathbb{E}[Y \neq \mathcal{T}(\mathbf{X}_j^{'};\Theta)] - \mathbb{E}\left[Y \neq \mathcal{T}\left(\mathbf{X};\Theta\right)\right]. \quad (5)$$

Then, assuming that $\mathcal{T}\left(\cdot;\Theta\right)$ is a 'perfect' classifier, which implies that $\forall \mathbf{X}, \mathcal{T}\left(\mathbf{X};\Theta\right) = Y$ and $\mathbb{E}\left[Y \neq \mathcal{T}\left(\mathbf{X};\Theta\right)\right] = 0$, Eq. (5) becomes:

$$\mathrm{VI}^*\left(X_j\right) = \mathbb{E}\left[Y \neq \mathcal{T}(X_1,\ldots,X_j^{'},\ldots,X_p)\right]. \quad (6)$$

Therefore, $\mathrm{VI}^*(X_j)$ is related to the quality of classification on the system state when the knowledge on the state of component C_j is non-informative. With respect to the DDIM, Eq. (6) becomes

$$\begin{aligned} \mathrm{DDIM}^*(C_j) := VI^*(X_j) &= \mathbb{E}\left[Y \neq \mathcal{T}(X_1,\ldots,X_j^{'},\ldots,X_p)\right] & (7)\\ &= \mathbb{P}\left[Y \neq \mathcal{T}(X_1,\ldots,X_j^{'},\ldots,X_p)\right]. & (8) \end{aligned}$$

According to Eq. (8), we can interpret the DDIM of Component C_j as the probability of misclassifying the system state when the knowledge on component C_j becomes non-informative, i.e., the state of the component is taken from a component of the same type operating in another virtual twin system.

3. Case Studies

In all the case studies considered in this work, we assume that the monitoring signals $X_j, j = 1\ldots,p$ are binary variables directly indicating the safe(0)/failed(1) state of the corresponding components. Although in a general case it is not always possible to directly know the component states, we expect that a fault diagnostic system able to infer the component state from monitoring signals can be built. Then, the outcome of the fault diagnostic system can be used as signal X_j.

The open-source R package `randomForest` [39] with the default parameter values is used to compute the importance scores and conduct the experiments. The number of trees is set equal to $n_{tree} = 500$ and the number of candidate features at each node where a tree is grown is set equal to $m_{try} = \max\{2, \lfloor\sqrt{p}\rfloor\}$. The use of default setting of the RF parameters has been verified by repeating the experiments with different combinations of n_{tree} and m_{try}, without obtaining a relevant improvement of the classification accuracy. Appendix A provides further details on the influence of the RF parameters on the classification accuracy and importance measure computation.

In the next Sections 3.1, 3.2 and 3.3, the proposed DDIM is applied to case studies of increasing complexity with respect to the number of components in the system. Section 3.1 considers some simple systems and investigates the robustness of the DDIM with respect to the quantity of data. Sections

3.2 and 3.3 present the application to complex systems such as the auxiliary feedwater system of a nuclear pressurized water reactor [10] and a synthetic case study which mimics the complexity of a real CTI, respectively.

In Sections 3.1 and 3.2, the rankings obtained by using the DDIM are compared with the corresponding rankings obtained by using the Birnbaum IM [12], which is defined by:

$$\mathrm{IM}^B(C_j) = \frac{\partial R(r_1, \ldots, r_p)}{\partial r_j}, \tag{9}$$

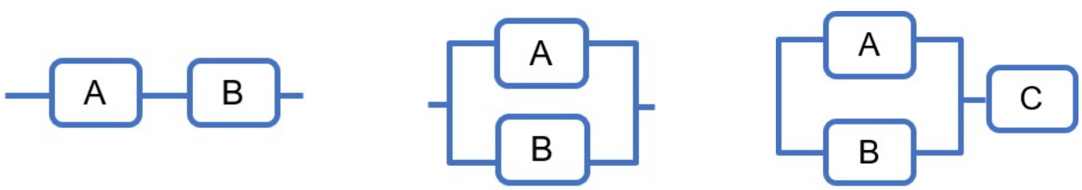
where $R(\cdot)$ denotes the system reliability function and r_j denotes the reliability of component C_j. Under assumptions of independence among component failures and binary safe(0)/failed(1) state X_j, $\mathrm{IM}^B(C_j)$ is:

$$\mathrm{IM}^B(C_j) = \mathbb{E}\left[\phi(X_1, \ldots, X_j = 0, \ldots, X_p) - \phi(X_1, \ldots, X_j = 1, \ldots, X_p)\right], \tag{10}$$

where ϕ denotes the system structure function. Notice that for CTIs such as the one considered in Section 3.3, the structure functions are typically not known, and, therefore, Birnbaum IM cannot be computed.

3.1. Simple Systems

We consider the simple systems shown in Figure 2.

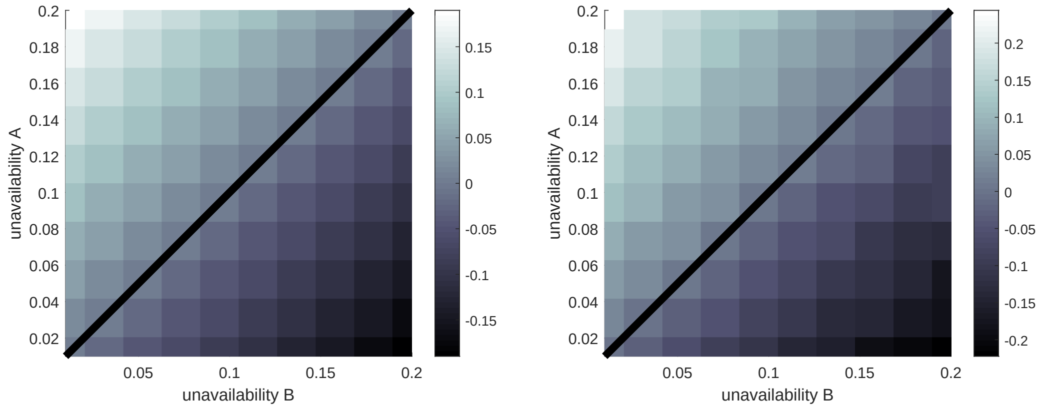


(a) 2-component series system (b) 2-component parallel system (c) 3-component system

Figure 2: Simple systems

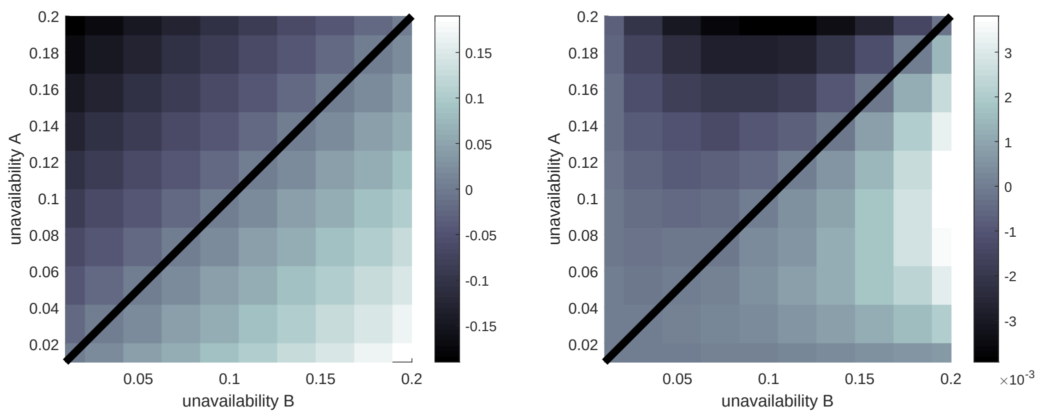
For the 2-component series system (Figure 2(a)), different cases characterized by components A and B unavailabilities in the range $[0.01, 0.2]$ have been considered. In each case, a training and a test datasets, $Data_{train}$ and $Data_{test}$, formed by $n_{train} = n_{test} = 5 \times 10^4$ instances, respectively, have been simulated by randomly sampling components A and B safe(0)/failed(1) states (X_A^i, X_B^i) and computing the corresponding system state y^i using the system structure function. Notice that, the simulated data allows to build a RF classifier, which does not make any error in the classification of the system state. For comparison, we compute the Birnbaum IMs, $\mathrm{IM}^{\mathrm{B}}(A)$ and $\mathrm{IM}^{\mathrm{B}}(B)$ for the considered systems. The obtained importance measures are shown in Figure 3. We observe that DDIM behaves very similar to the Birnbaum IM.

The above experiment has been repeated for the 2-component parallel system (Figure 2 (b)). Figure 4 shows that DDIM provides the same ranking of the two components as the Birnbaum importance



(a) $\mathrm{IM}^{\mathrm{B}}(A) - \mathrm{IM}^{\mathrm{B}}(B)$ (b) $\mathrm{DDIM}(A) - \mathrm{DDIM}(B)$

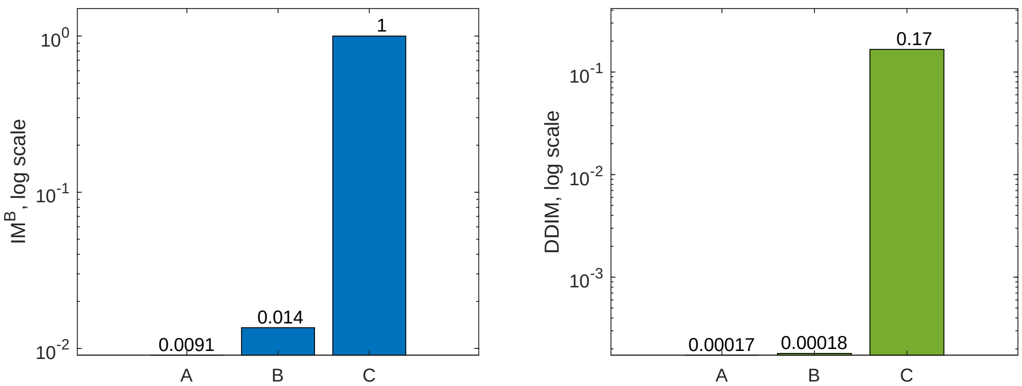
Figure 3: Difference between the Birnbaum IM of components A and B, $\mathrm{IM}^{\mathrm{B}}(A) - \mathrm{IM}^{\mathrm{B}}(B)$ (left), and difference between the data-driven IM of components A and B, $\mathrm{DDIM}(A) - \mathrm{DDIM}(B)$ (right), for the 2-component series system. The black line divides the upper zone where $\mathrm{DDIM}(A) \geq \mathrm{DDIM}(B)$ from the bottom zone where $\mathrm{DDIM}(A) < \mathrm{DDIM}(B)$.



(a) $\mathrm{IM}^{\mathrm{B}}(A) - \mathrm{IM}^{\mathrm{B}}(B)$ (b) $\mathrm{DDIM}(A) - \mathrm{DDIM}(B)$

Figure 4: Difference between the Birnbaum importance measure of components A and B, $\mathrm{IM}^{\mathrm{B}}(A) - \mathrm{IM}^{\mathrm{B}}(B)$ (left), and difference between the data-driven importance measure of components A and B, $\mathrm{DDIM}(A) - \mathrm{DDIM}(B)$, the 2-component parallel system. The black line divides the upper zone where $\mathrm{DDIM}(A) < \mathrm{DDIM}(B)$ from the bottom zone where $\mathrm{DDIM}(A) \geq \mathrm{DDIM}(B)$.

measure. When, the differences between the importance measures of the two components are considered (Figure 4), then, it is interesting to observe that, whereas the maximum difference of the Birnbaum importance measure is obtained when one component has the largest unavailability and the other the smallest, the maximum difference of the DDIM is in correspondence of the unavailability of a component equal to 0.2 (maximum considered value) and the unavailability of the other equal to 0.1. This is due to the fact that $\mathrm{DDIM}(C_j)$ measures the importance of a component according to its contribution to the classification error, which depends from the unavailabilities of both components, whereas the Birnbaum IM depends only on the unavailability of the other component.



(a) IM^{B}, Y-axis in log scale

(b) DDIM, Y-axis in log scale

Figure 5: Birnbaum IM^{B} and DDIM importance measures for the 3-component system with unavailabilities of components A, B and C equal to $0.015, 0.01$, and 0.095, respectively.

We consider the 3-component system of Figure 2(c) with the unavailabilities of components A, B and C equal to $0.015, 0.01$ and 0.095, respectively [50]. The training and test datasets, $Data_{train}$ and $Data_{test}$, of sizes $n_{train} = n_{test} = 10^4$, respectively, are used to compute DDIM(A), DDIM(B) and DDIM(C). The out-of-sample classification performance is satisfactory with no misclassification. Figure 5 shows that the proposed data-driven importance measure clearly identifies component C as the most critical, which is consistent with the ranking of the Birnbaum importance measure. This 3-component system is used to investigate the following two practical issues.

Does the dataset contain sufficient information for estimating DDIM and ranking the components? A common practice to assess the uncertainty of the estimate of data-driven approaches is to compute the associated confidence interval. In practical cases, where only a dataset with finite observations is available, *bootstratp* or *jackknife* are typically employed [41]. Figure 6 shows the bootstrap confidence intervals associated with DDIM estimates for the 3-component system (unavailabilities of components A,B and C equal to 0.2, 0.3 and 0.1, respectively), using 30 bootstrap replicates. When a training dataset of size $n_{train} = 50$ is used, it is not possible to rank the data-driven criticality of components A and B, whereas, when the training dataset size increases, the overlapping among the confidence intervals decreases. For example, when the training sample size increases to $n_{train} = 1361$, the ranking becomes clear.

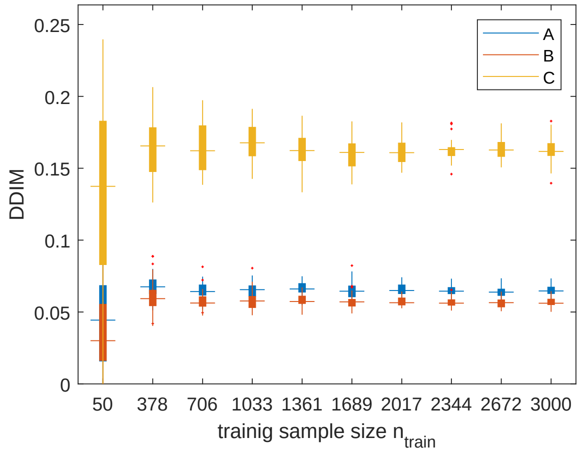
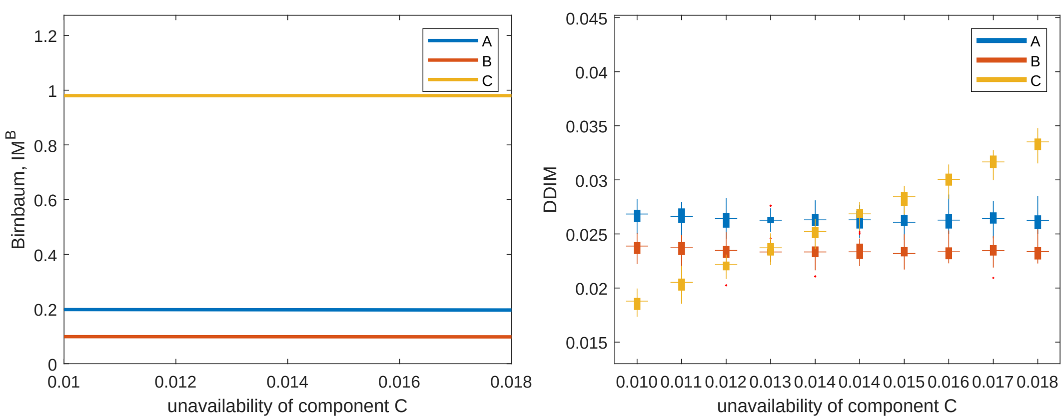


Figure 6: DDIM for the 3-component system when the number of instances increases from 50 to 3000.



(a) Birnbaum IM$^{\mathrm{B}}$

(b) DDIM

Figure 7: Birnbaum IM$^{\mathrm{B}}$ and DDIM importance measures for the 3-component system with unavailability of component C varying in the interval $[0.01, 0.018]$.

When do DDIM and Birnbaum IM^B disagree? To investigate the conditions under which the ranking provided by the DDIM differs from the one provided by Birnbaum IM, the following experiment has been performed. The unavailabilities of components A and B have been set equal to 0.1 and 0.2, respectively, whereas the unavailability of component C has been varied in the interval $[0.01, 0.018]$. The boxplot in Figure 7 (b) shows the obtained DDIM and IM$^{\mathrm{B}}$ using a training dataset of size $n_{train} = 10^4$ and performing 30 macro-replicates of the experiment. Since the unavailabilities of components A and B are fixed, IM$^{\mathrm{B}}(C)$ remains unchanged although the unavailability of component C varies. On the other side, DDIM(C) increases as the unavailability of component C increases, and DDIM ranks component C as less critical than components A and B when its unavailability is smaller than 0.012. This is due to the fact that DDIM measures the importance of a component with respect to its contribution to the classification error. In this case, the system has two minimal cut sets: {C} and {A,B}. Therefore, when the unavailability of component C becomes smaller than the product of the unavailabilities of components A and B, the impact on the classification error of permuting the states of component C in the dataset is reduced since the correct knowledge of the states of components A and B allows classifying the majority of the system states in the dataset.

3.2. The Auxiliary Feedwater System in a Pressurized Water Reactor Plant

We consider the auxiliary feedwater system of a nuclear pressurized water reactor, whose reliability block diagram is shown in Figure 8 [10]. All $p = 14$ components are in standby during plant operation

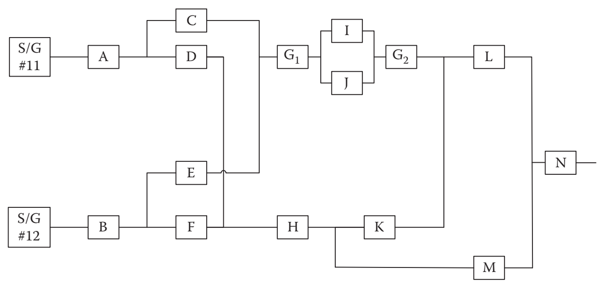


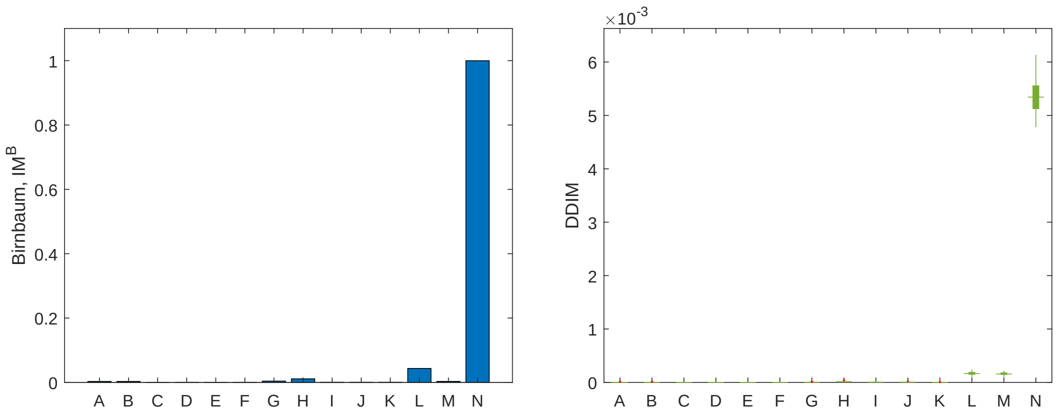
Figure 8: The reliability block diagram of the auxiliary feedwater system [10].

in normal condition and can randomly fail. The failure time of component C_j is assumed to be exponentially distributed with constant failure intensity λ_j (h^{-1}). Components are periodically tested at intervals $T(h)$, and the average test and repair durations are $T_{t,j}(h)$ and $T_{R,j}(h)$, respectively.

The specific parameters defining the testing policy are taken from [10] and reported in Table 1. The safe/failed states of the components have been simulated and the corresponding system state obtained by using the system structure function. The obtained training dataset is formed by $n_{train} = 9 \times 10^4$ instances, and 30 macro-replicates are used to obtain the confidence intervals. The estimated data-driven IMs and the analytically computed Birnbaum IMs are shown in Figure 9.

Table 1: Characteristics of an auxiliary feedwater system

Component name	Failure intensity λ_j (h^{-1})	Average test duration $T_{t,j}$ (h)	Average repair time $T_{R,j}$ (h)	Test interval T (h)
A	1×10^{-7}	2	5	720
B	1×10^{-7}	2	5	720
C	1×10^{-6}	2	10	720
D	1×10^{-6}	2	10	720
E	1×10^{-6}	2	10	720
F	1×10^{-6}	2	10	720
G	1×10^{-7}	2	15	720
H	1×10^{-7}	2	24	720
I	1×10^{-4}	4	36	720
J	1×10^{-4}	4	36	720
K	1×10^{-5}	4	24	720
L	1×10^{-7}	2	10	720
M	1×10^{-4}	2	10	720
N	1×10^{-7}	2	5	720



(a) IM^B (b) DDIM

Figure 9: Birnbaum IM^{B} and DDIM importance measures of the AFW system components.

Both importance measures recognize component N remarkably more important than the other components and component L as the second most important. The importance of components A, B, C, D, E, F, I, J, K are negligible for both methods. The main difference among the two rankings involves component M, which is considered more important by the DDIM (third position in the ranking) than by the IM$^{\mathrm{B}}$ (fifth position in the ranking). Notice that the Birnbaum IM measures the criticality of a component as the increase of the CTI unreliability when the component state switches from safe to failed, while DDIM assesses the component criticality considering the error in the classification of the system state when the component state is taken from a virtual twin system. The difference in the criticality ranking of component M is due to the fact that it is characterised by the largest failure intensity among the system components. Therefore, its replacement with a component taken from a virtual twin system is causing a change of the instances to be classified in much more cases than what is occurring for other components characterised by smaller failure intensities, for which the probability that the component is in the safe state in both the real and the virtual twin systems is very large.

3.3. The 50-component System

We consider a CTI formed by $p = 50$ components, in which each component can be in five states, $D \in \{1, 2, 3, 4, 5\}$ corresponding to healthy, partially degraded, degraded, very degraded and failed, respectively [11]. The components perform transitions among the states at random times. Figure 10 shows the possible stochastic state transitions corresponding to: degradation (from $D = 1$ to $D = 2$, from $D = 2$ to $D = 3$ and from $D = 3$ to $D = 4$), partial restoration (from $D = 4$ to $D = 3$, from $D = 3$ to $D = 2$ and from $D = 2$ to $D = 1$), failure (from $D = 4$ to $D = 5$) and complete repair (from $D = 5$ to $D = 1$). Table 2 reports the time-invariant transition rates, $\lambda_j^{D^{'} \to D^{''}}$ of component j from state $D^{'}$ to state $D^{''}, D^{'} \neq D^{''}$. Each component C_j is monitored by a signal X_j directly measuring its state D.

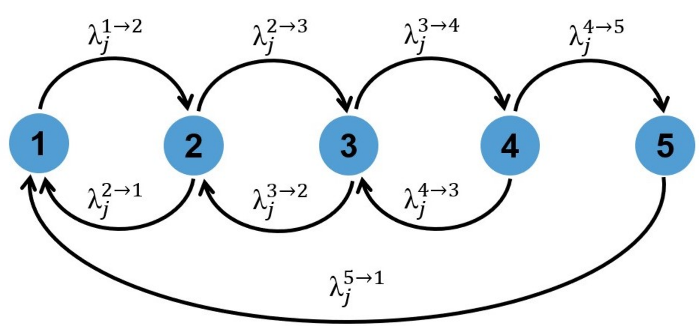


Figure 10: State transitions of a CTI component.

We assume that the CTI can fail due to two cascading failures:

(1) Component C_{11} performs a transition from state 4 to state 5, which can cause an ordered sequence of events leading to the transitions of components C_{12}, C_{13}, C_{14}, C_{15} and C_{16} into state 5 and the consequent failure of the CTI. The probability of failure propagation between any two components in the sequence is set to 0.95 and the time necessary for the malfunction propagation follows a uniform distribution in the interval $[1, 20]$ minutes;

Table 2: Transition rates in hours^{-1}

Component C_j	Transition rates			
$j = 1, 2, 3, 6 \ldots, 10,$	$\lambda_j^{1\to 2} = 0.5$	$\lambda_j^{2\to 3} = 0.02$	$\lambda_j^{3\to 4} = 0.5$	$\lambda_j^{4\to 5} = 0.01$
$11, 12, 17, 21, 22, 35, 36$	$\lambda_j^{2\to 1} = 0.5$	$\lambda_j^{3\to 2} = 0.01$	$\lambda_j^{4\to 3} = 0.4$	$\lambda_j^{5\to 1} = 0.2$
$j = 4, 5, 13, 14, 18, 19, 20, 23,$	$\lambda_j^{1\to 2} = 0.3$	$\lambda_j^{2\to 3} = 0.005$	$\lambda_j^{3\to 4} = 0.4$	$\lambda_j^{4\to 5} = 0.01$
$24, 27, \ldots, 34, 38, 39$	$\lambda_j^{2\to 1} = 0.3$	$\lambda_j^{3\to 2} = 0.01$	$\lambda_j^{4\to 3} = 0.4$	$\lambda_j^{5\to 1} = 0.2$
$j = 15, 16, 25, 26, 37,$	$\lambda_j^{1\to 2} = 0.4$	$\lambda_j^{2\to 3} = 0.005$	$\lambda_j^{3\to 4} = 0.4$	$\lambda_j^{4\to 5} = 0.01$
$40, \ldots, 50$	$\lambda_j^{2\to 1} = 0.4$	$\lambda_j^{3\to 2} = 0.01$	$\lambda_j^{4\to 3} = 0.4$	$\lambda_j^{5\to 1} = 0.2$

(2) Component C_{21} performs a transition from state 4 to state 5, which can cause an ordered sequence of events leading to the transitions of components C_{22}, C_{23}, C_{24}, C_{25} and C_{26} into the state 5 and the consequent failure of the CTI. The probability of failure propagation between any two components in the sequence is set to 0.95 and the time necessary for the malfunction propagation follows a uniform distribution in the interval $[1, 30]$ minutes.

The CTI critical components are those involved in the two cascading failures, i.e. C_{11} C_{12}, C_{13}, C_{14}, C_{15} ,C_{16}, and C_{21}, C_{22}, C_{23}, C_{24}, C_{25} , C_{26}.

The CTI behaviour is simulated for 720 days and the multi-state signals $X_j, j = 1 \ldots 50$ assessing the j-th component degradation state are collected every 2 hours, together with the corresponding CTI safe(0)/failed(1) state Y. Therefore, a dataset $Data = \{(\mathbf{x}^i, y^i)\}_{i=1}^{n_{total}}$ formed by $n_{total} = 8640$ instances is obtained. The simulated dataset is imbalanced, being the fraction of positive instances ($y^i = 1$) over the total number n_{total} of simulated instances equal to 5.3%.

3.3.1. Importance Measures

The computation of the importance measures is performed using 50% of the $n_{total} = 8640$ instances of $Data$. The remaining 50% of instances are used for testing the classification performance. The defined RF classifier is characterised by a satisfactory accuracy, being the missed alarm and false alarm rates equal to 0 and 0.00024, respectively.

Figure 11 shows the estimated DDIMs for the 50-component CTI. The components from the first cascading chain are coloured in white, and those from the second cascading chain in black. DDIM correctly assigns the largest importance values to the 12 critical CTI components. Note that the proposed DDIM tends to assign larger criticality values to the components at the end of the failure chains, e.g., C_{16} and C_{26}, than to those at the beginning. This is due to the fact that the probability of having a cascading failure given the failure of a component at the end of the chain is larger than the same probability given a failure of a component at the beginning of the chain. Therefore, from a feature selection point of view, the state of the component at the end of the chain is more informative.

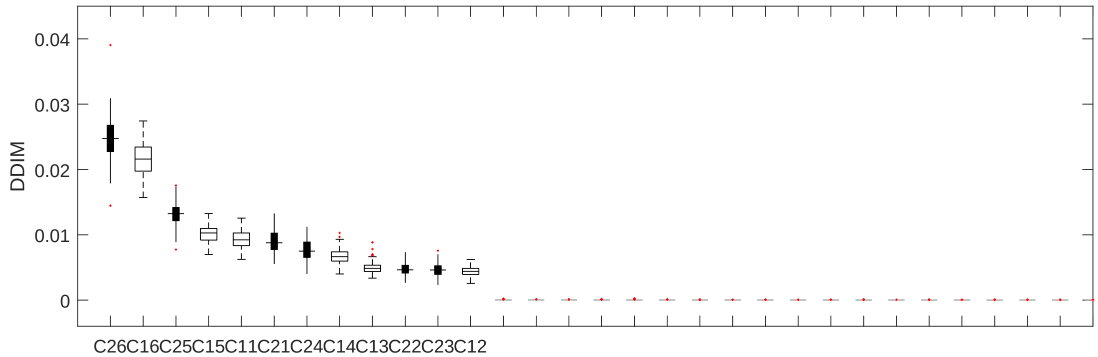


Figure 11: Top 30 components in the 50-component CTI ranked by DDIM. Boxplots are obtained via bootstrapping with bootstrap size 100.

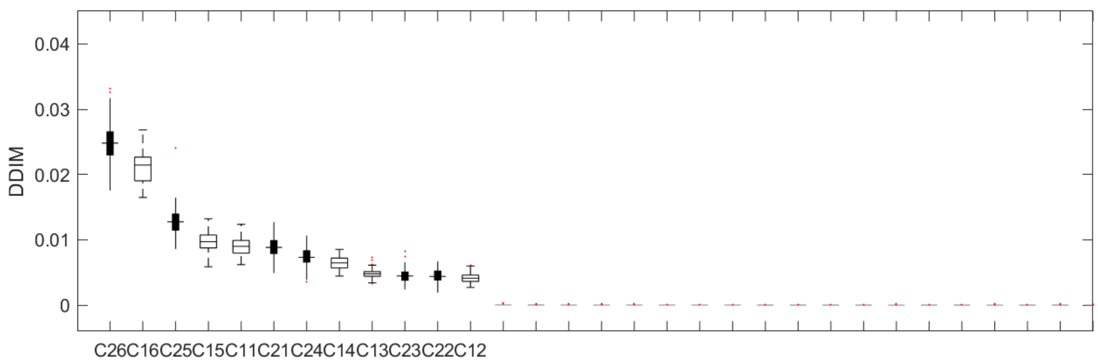


Figure 12: Top 30 components in the 330-component CTI ranked by DDIM. Boxplots are obtained via bootstrapping with bootstrap size 100.

To test the stability of the proposed method, a further experiment is performed by adding 280 non-critical components, i.e., components whose failure operation state has no influence on the system state, to the previous 50-component CTI. Figure 12 shows that DDIM is still able to correctly identify the 12 critical components. Comparing to the ranking obtained in the previous case without noise components, the only difference is the order of components C_{22} and C_{23}, which have very similar values of DDIM with overlapping bootstrap confidence intervals, which do not allow ordering them. Notice that the bootstrap confidence intervals for the 330-component CTI are slightly larger than the 50-component CTI. This observation is consistent with the experiments in Section 3.1, a larger number of components p to investigate would require a larger number of instances n_{train} in order to achieve the same level of uncertainty.

4. Conclusion

This work has addressed the problem of identifying the critical components of a complex technical infrastructure in the challenging case in which the system structure function is unknown. A data-driven framework is introduced to allow using the information contained in the available monitoring data. Component ranking is based on a data-driven importance measure which associates to each component the importance score of the associated features. It can be interpreted as the probability of misclassifying the system state if monitoring data coming from an identical component operating in a virtual twin system were used. The proposed framework is able to estimate whether the available monitoring data contain enough information to assess component criticality by observing the accuracy of the RF classification of the CTI safe/failed state. Also, the use of a bootstrap technique allows estimating the confidence intervals associated to the obtained importance measure.

Empirical experiments on simple systems show that the proposed data-driven framework provides ranking of the components' importance similar to those obtained by the Birnbaum IM. The applications to a nuclear power plant system and to a case study which mimics the complexity of a CTI shows that the most critical components are correctly identified. The obtained results encourage the use of data-driven methods for investigating the risk and reliability of CTIs, whose components are normally monitored.

Future research directions will include the investigation of alternative permutation schemes for feature importance score computation and the development of a procedure for the selection, from the obtained data-driven importance measures and component ranking, of the set of critical components for the CTI reliability and availability.

Appendix A. Analysis of the Influence of the RF Parameters on the Classification Accuracy and Importance Measure Computation

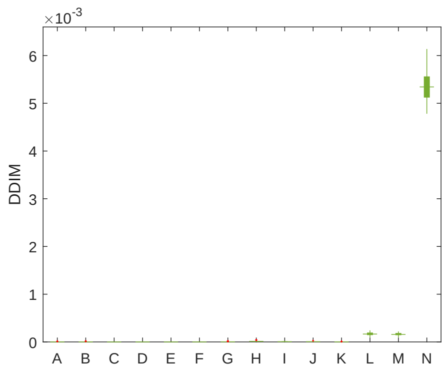
We consider the case study of Section 3.2 about the auxiliary feedwater system of a nuclear pressurized water reactor. Table A.3 shows the performance of the RF classifier when varying the two key RF

Table A.3: Analysis of the Influence of the RF Parameters on the Classification Accuracy

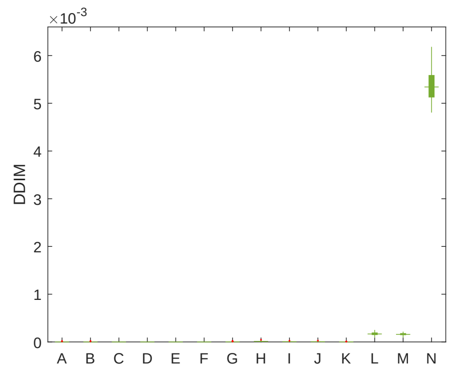
Experiment	RF parameter	Classification performance
(a)	$n_{tree} = 500, m_{try} = 3$	False Alarm Rate = 0, Missed Alarm Rate = 0.014
(b)	$n_{tree} = 800, m_{try} = 3$	False Alarm Rate = 0, Missed Alarm Rate = 0.016
(c)	$n_{tree} = 100, m_{try} = 3$	False Alarm Rate = 0, Missed Alarm Rate = 0.015
(d)	$n_{tree} = 500, m_{try} = 2$	False Alarm Rate = 0, Missed Alarm Rate = 0.050

parameters, n_{tree} and m_{try}. In Experiment (a), default RF parameter values, $n_{tree} = 500, m_{try} = 3$, are used. In the Experiments (b) and (c), the RF parameter n_{tree} has been varied from the original value of 500 to 800 and 100, respectively, while keeping the parameter $m_{try} = 3$. The obtained results show that the performance of the RF only slightly decreases (the missed alarm rate increases from 0.014 to 0.016 and 0.015, respectively). In Experiment (d), the RF parameter n_{mtry} has been varied from the original value of 3 to 2, while keeping the parameter $n_{tree} = 500$. This setting has led to a worsening of the performance (the missed alarm rate increases from 0.014 to 0.050). In all experiments, the false alarm rate remains zero.

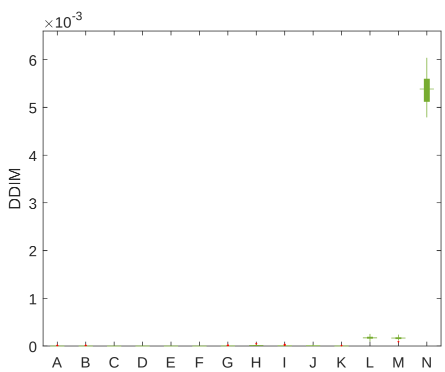
Figure A.13 shows that the ranking of the components is not significantly affected by the parameter variation.



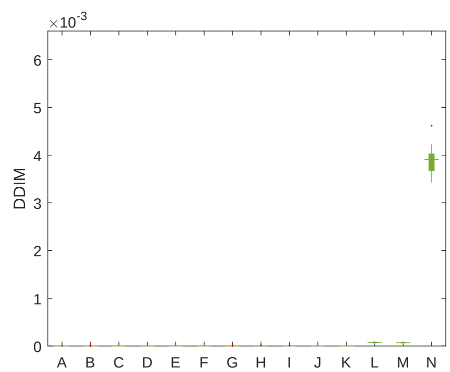
(a) $n_{tree} = 500, m_{try} = 3$



(b) $n_{tree} = 800, m_{try} = 3$



(c) $n_{tree} = 100, m_{try} = 3$



(d) $n_{tree} = 500, m_{try} = 2$

Figure A.13: Sensitivity Analysis of the RF Parameters for the auxiliary feedwater system of a nuclear pressurized water reactor case study in Section 3.2.

The analysis has shown that the performance of the RF classifier is not significantly affected by variations of the parameters n_{tree} and m_{try}, and that the component ranking is stable with respect to the parameter setting.

Appendix B. Pseudo-code for the Data-driven Importance Measure Computation

Algorithm 1 Pseudo-code for the Data-driven Importance Measure Computation

Require: $Data_{train}$, RF parameters m_{try}, n_{tree}

1. **Train Random Forest**
   (a) Draw n_{tree} bootstrap samples $\mathcal{D}_t, t = 1 \ldots n_{tree}$ from the dataset $Data_{train}$.
   (b) For $t = 1 \ldots n_{tree}$, grow an unpruned decision tree $\mathcal{T}(\cdot; \Theta_t)$ on $\mathcal{D}_t$: at each node, choose the best splitting feature from a random subset of size m_{try} of all the p features.
   (c) Compute the misclassification error $\mathcal{E}(\mathcal{T}(\cdot; \Theta_t), \mathcal{D}_t^{OOB})$ evaluated on $\mathcal{D}_t^{OOB} = Data_{train} \backslash \mathcal{D}_t$ using

$$\mathcal{E}\left(\mathcal{T}(\cdot; \Theta_t), \mathcal{D}^{OOB}\right) = \frac{1}{n_{\mathcal{D}}} \sum_{i:(\mathbf{x}^i, y^i) \in \mathcal{D}^{OOB,j}} \mathbb{1}\left(y^i \neq \mathcal{T}(\mathbf{x}^i; \Theta_t)\right), \quad \text{(B.1)}$$

   where $\mathbb{1}(\cdot)$ denotes the indicator function.
2. **Compute the importance score of feature X_j: VI (X_j)**
   (a) For $j = 1 \ldots p$, do
      i. For $t = 1 \ldots n_{tree}$, obtain the dataset $\mathcal{D}_t^{OOB,j}$ by permuting the values of feature X_j in $\mathcal{D}_t^{OOB}$ and calculate $\mathcal{E}(\mathcal{T}(\cdot; \Theta_t), \mathcal{D}_t^{OOB,j})$.
      ii. Compute:

$$\mathrm{DDIM}(C_j) = \mathrm{VI}\left(X_j\right) = \frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} \Big(\mathcal{E}(\mathcal{T}(\cdot; \Theta_t), \mathcal{D}_t^{OOB,j}) - \mathcal{E}(\mathcal{T}(\cdot; \Theta_t), \mathcal{D}_t^{OOB})\Big). \quad \text{(B.2)}$$

Output: $\mathrm{DDIM}(C_j), j = 1 \ldots p$

Acknowledgement

The authors would like to thank Federico Antonello (Politecnico di Milano) for providing the dataset of the 50-component case study in Section 3.3.

References

[1] C. B. Keating, J. J. Padilla, K. Adams, System of systems engineering requirements: challenges and guidelines, Engineering Management Journal 20 (4) (2008) 24–31.

[2] I. Eusgeld, C. Nan, S. Dietz, System-of-systems approach for interdependent critical infrastructures, Reliability Engineering & System Safety 96 (6) (2011) 679–686.

[3] E. Zio, Challenges in the vulnerability and risk analysis of critical infrastructures, Reliability Engineering & System Safety 152 (2016) 137–150.

[4] S. Salcedo-Sanz, L. Cornejo-Bueno, L. Prieto, D. Paredes, R. García-Herrera, Feature selection in machine learning prediction systems for renewable energy applications, Renewable and Sustainable Energy Reviews 90 (2018) 728–741.

[5] V. Bolón-Canedo, A. Alonso-Betanzos, Ensembles for feature selection: A review and future trends, Information Fusion 52 (2019) 1–12.

[6] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, G. Nenadic, Machine learning methods for wind turbine condition monitoring: A review, Renewable Energy 133 (2019) 620–635.

[7] R. Genuer, J. M. Poggi, C. Tuleau-Malot, Variable selection using random forests, Pattern Recognition Letters 31 (14) (2010) 2225–2236.

[8] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, VSURF: Variable selection using random forests, The R Journal 7 (December) (2015) 19–33.

[9] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[10] M. Modarres, M. P. Kaminskiy, V. Krivtsov, Reliability Engineering and Risk Analysis: A Practical Guide, CRC Press, 2017.

[11] F. Antonello, P. Baraldi, A. Shokry, E. Zio, U. Gentile, L. Serio, Data-driven extraction of association rules of dependent abnormal behaviour groups, in: Proceedings of the 29th International European Safety and Reliability Conference, ESREL 2019, 2019.

[12] Z. W. Birnbaum, On the importance of different components in a multicomponent system, Multivariate Analysis (1969) 581–592.

[13] J. D. Andrews, S. Beeson, Birnbaum's measure of component importance for noncoherent systems, IEEE Transactions on Reliability 52 (2) (2003) 213–219.

[14] J. K. Vaurio, Importances of components and events in non-coherent systems and risk models, Reliability Engineering & System Safety 147 (2016) 117–122.

[15] J. B. Fussell, How to hand-calculate system reliability and safety characteristics, IEEE Transactions on Reliability R-24 (3) (1975) 169–174.

[16] J. Mi, M. Beer, Y.-f. Li, M. Broggi, Y. Cheng, Reliability and importance analysis of uncertain system with common cause failures based on survival signature, Reliability Engineering & System Safety 201 (106988) (2020).

[17] J. S. Hong, C. H. Lie, Joint reliability-importance of two edges in an undirected network, IEEE Transactions on Reliability 42 (1) (1993) 17–23.

[18] A. Gandini, Importance & sensitivity analysis in assessing system reliability, IEEE Transactions on Reliability 39 (1) (1990) 61–70.

[19] E. Borgonovo, G. E. Apostolakis, A new importance measure for risk-informed decision making, Reliability Engineering & System Safety 72 (2) (2001) 193–212.

[20] Y. Fu, T. Yuan, X. Zhu, Importance-measure based methods for component reassignment problem of degrading components, Reliability Engineering & System Safety 190 (2019) 106501.

[21] D. Lyu, S. Si, Dynamic importance measure for the k-out-of-n: G system under repeated random load, Reliability Engineering & System Safety 195 (2020) 106720.

[22] Y. Sheng, H. Ke, Reliability evaluation of uncertain k-out-of-n systems with multiple states, Reliability Engineering & System Safety 195 (2020) 106696.

[23] P. Do, C. Bérenguer, Conditional reliability-based importance measures, Reliability Engineering & System Safety 193 (2020) 106633.

[24] M. C. Cheok, G. W. Parry, R. R. Sherry, Use of importance measures in risk-informed regulatory applications, Reliability Engineering & System Safety 60 (3) (1998) 213–226.

[25] W. Kuo, X. Zhu, Some recent advances on importance measures in reliability, IEEE Transactions on Reliability 61 (2) (2012) 344–360.

[26] H. Peng, D. W. Coit, Q. Feng, Component reliability criticality or importance measures for systems with degrading components, IEEE Transactions on Reliability 61 (1) (2012) 4–12.

[27] X. Zhu, W. Kuo, Importance measures in reliability and mathematical programming, Annals of Operations Research 212 (1) (2014) 241–267.

[28] E. Amaldi, V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems, Theoretical Computer Science 209 (1-2) (1998) 237–260.

[29] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324.

[30] E. Zio, P. Baraldi, N. Pedroni, Selecting features for nuclear transients classification by means of genetic algorithms, IEEE Transactions on Nuclear Science 53 (3) (2006) 1479–1493.

[31] A. L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1-2) (1997) 245–271.

[32] I. Guyon, S. Gunn, A. Ben-Hur, G. Dror, Result Analysis of the NIPS 2003 Feature Selection Challenge, in: L. K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing Systems 17, MIT Press, 2005, pp. 545–552.

[33] R. Tibshirani, Regression shrinkage and selection via the LASSO: A retrospective, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73 (3) (2011) 273–282.

[34] G. M. Abdella, M. R. Maleki, S. Kim, K. N. Al-Khalifa, A. M. S. Hamouda, Phase-I monitoring of high-dimensional covariance matrix using an adaptive thresholding LASSO rule, Computers and Industrial Engineering 144 (2020) 106465.

[35] K. Pearson, On the General Theory of Skew Correlation and Non-linear Regression, Vol. XIV of Mathematical Contributions to the Theory of Evolution, Drapers' Company Research Memoirs, Dulau & Co., London, 1905.

[36] H. Almuallim, T. G. Dietterich, Learning with many irrelevant features, in: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), Anaheim, CA, 1991, pp. 547–552.

[37] D. N. Reshef, Y. a. Reshef, H. K. Finucane, S. R. Grossman, G. Mcvean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, P. C. Sabeti, Detecting novel associations in large data sets, Science 334 (2011) 1518–1524.

[38] K. J. Archer, R. V. Kimes, Empirical characterization of random forest variable importance measures, Computational Statistics and Data Analysis 52 (4) (2008) 2249–2260.

[39] L. Breiman, A. Cutler, A. Liaw, M. Wiener, randomForest: Breiman and Cutler's random forests for classification and regression. https://cran.r-project.org/package=randomForest (2018).

[40] L. Breiman, A. Cutler, Online manual for random forests. https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm, 2002.

[41] B. Efron, The Jackknife, the Bootstrap and Other Resampling Plans, Society for Industrial and Applied Mathematics, Philadelphia, 1982.

[42] C. Strobl, A. L. Boulesteix, A. Zeileis, T. Hothorn, Bias in random forest variable importance measures: Illustrations, sources and a solution, BMC Bioinformatics 8 (1) (2007) 25.

[43] A. Altmann, L. Toloşi, O. Sander, T. Lengauer, Permutation importance: A corrected feature importance measure, Bioinformatics 26 (10) (2010) 1340–1347.

[44] C. Chen, A. Liaw, L. Breiman, Using random forest to learn imbalanced data, Tech. rep., University of California, Berkeley (2004).

[45] M. Gevrey, I. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models, Ecological Modelling 160 (3) (2003) 249–264.

[46] B. Gregorutti, B. Michel, P. Saint-Pierre, Correlation and variable importance in random forests, Statistics and Computing 27 (3) (2017) 659–678.

[47] A. Fisher, C. Rudin, F. Dominici, All models are wrong but many are useful: variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance, arXiv preprint arXiv:1801.01489 (2018).

[48] G. Louppe, L. Wehenkel, A. Sutera, P. Geurts, Understanding variable importances in forests of randomized trees, in: C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 431–439.

[49] S. J. Kazemitabar, A. A. Amini, A. Bloniarz, A. Talwalkar, Variable importance using decision trees, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, G. R. (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 426–435.

[50] P. Baraldi, M. Compare, E. Zio, Component ranking by Birnbaum importance in presence of epistemic uncertainty in failure event probabilities, IEEE Transactions on Reliability 62 (1) (2013) 37–48.