Giovanni Bechini
Siemens Energy,
Munich 81739, Germany
e-mail: giovanni.bechini@siemens-energy.com

Enzo Losi
Department of Engineering,
University of Ferrara,
Ferrara 44121, Italy
e-mail: enzo.losi@unife.it

Lucrezia Manservigi
Department of Engineering,
University of Ferrara,
Ferrara 44121, Italy
e-mail: lucrezia.manservigi@unife.it

Giovanni Pagliarini
Department of Mathematics and
Computer Science,
University of Ferrara,
Ferrara 44121, Italy
e-mail: giovanni.pagliarini@unife.it

Guido Sciavicco
Department of Mathematics and
Computer Science,
University of Ferrara,
Ferrara 44121, Italy
e-mail: guido.sciavicco@unife.it

Ionel Eduard Stan
Department of Mathematics and
Computer Science,
University of Ferrara,
Ferrara 44121, Italy
e-mail: ioneleduard.stan@unife.it

Mauro Venturini
Department of Engineering,
University of Ferrara,
Ferrara 44121, Italy
e-mail: mauro.venturini@unife.it

Statistical Rule Extraction for Gas Turbine Trip Prediction

Gas turbine trip is an operational event that arises when undesirable operating conditions are approached or exceeded, and predicting its onset is a largely unexplored area. The application of novel artificial intelligence methods to this problem is interesting both from the computer science and the engineering point of view, and the results may be relevant in both the academia and the industry. In this paper, we consider data gathered from a fleet of Siemens industrial gas turbines in operation that includes several thermodynamic variables observed during a long period of operation. To assess the possibility of predicting trip events, we first apply a new, systematic statistical analysis to identify the most important variables, then we use a novel machine learning technique known as temporal decision tree, which differs from canonical decision tree because it allows a native treatment of the temporal component, and has an elegant logical interpretation that eases the posthoc validation of the results. Finally, we use the learned models to extract statistical rules. As a result, we are able to select the five most informative variables, build a predictive model with an average accuracy of 73%, and extract several rules. To our knowledge, this is the first attempt to use such an approach not only in the gas turbine field but also in the whole industry domain. [DOI: 10.1115/1.4056287]

Introduction

Gas turbines have reached a primary position in thermal power generation field thanks to their fast deliveries of power and the availability of natural gas. Given the ever-growing amount of data produced and collected in industrial processes in general, and in turbine operations in particular, it is natural to consider bottom-up learning tasks (that is, machine learning tasks) as a primary tool to extract knowledge from such processes, with the aim of designing systems that may help to ensure a reliable predictive maintenance program. A *trip* event in a gas turbine is an unscheduled operational event during which a turbine abnormally shuts down from a certain operation state, thus leading to a direct impact on its lifespan and revenue [1]. The reasons that lead to a trip event can be many, including, for example, too high vibrations, abnormal deviations and/or gradients of exhaust gas temperatures, and problems within the fuel spray nozzles. A trip can also occur when a turbine is unable to reach the self-sustained speed so that the startup process has to be repeated. In any case, each trip occurrence entails an increase in costs, due to the subsequent necessary repair and maintenance, as well as due to the production inter-ruption. Given the dynamic nature of such an event, a trip may also lead to compressor surge, so that a proper control must be activated to avoid unsafe consequences for the whole machinery [2–4]. Thus, a reliable method that is able to predict, in some form, the occurrence of a trip would be extremely beneficial for both the gas turbine manufacturer and its users.

Recording the values of several variables during a gas turbine operation gives rise to a multivariate time series, and virtually all engineering tasks of interest in gas turbines, from the machine learning point of view, can be interpreted as classification or regression tasks. Classification and regression of time series have been widely approached in the literature. To mention a few relevant contributions, in Ref. [5] the authors proposed a methodology for classifying sets of data points in a multidimensional space based on the common regions through which only time series of one class pass; in Ref. [6] a new measure of distance between time series based on the normalized periodogram which estimates the spectral density of a signal has been presented; authors in

Manuscript received July 19, 2022; final manuscript received July 30, 2022; published online January 10, 2023. Editor: Jerzy T. Sawicki.

Ref. [7] proposed a highly-comparative method for learning feature-based classifiers for monovariate time series, using more than 9000 features; in Ref. [8], the authors presented a sequence auto-encoder based on a previous sequence-to-sequence model; finally, Längkvist et al. [9] gave an up-to-date taxonomy for neural network-based methods for time series classification and regression. In the recent literature, the potential benefits of applying functional learning techniques, such as the ones above, to gas turbine diagnostics have been studied to some extent [10,11]; more specifically, in Refs. [12–15], the authors focus on applying several functional models to gas turbine monitoring, from support vector machines to artificial neural networks, to nonlinear autoregressive models. Finally, in Ref. [16] the authors used a hidden Markov model applied to gas turbine sensors and actuators, and in Ref. [17] a particular kind of neural network model, known as extreme learning machine, was used to perform diagnostics of gas turbines by employing features extracted from vibration signals. A common element to virtually all attempts at applying learning techniques to gas turbine diagnostics and predictive maintenance tasks, up to and including trip events, is the *functional* nature of the applied methods. Functional models, from simple linear regression to neural networks, are generally considered a good approach in statistical terms, but they lack interpretability. In other words, while good models can be learned that are able to make reliable predictions, such predictions cannot be immediately explained in terms of rules, especially logical rules, that help to understand the nature of the predicted event. This is particularly important in many contexts, and even more so in trip prediction for gas turbines, which is, as we shall see, a very challenging problem. When data do not support the construction of reliable models in absolute terms, functional models become less useful, as they cannot be inspected, and the partial knowledge that may have been extracted is hidden behind not always satisfactory statistical results. *Symbolic* learning methods are an alternative to functional ones. They are well-known since the beginning of the machine learning era, but are often less considered as a potential solution for several reasons: *(i)* symbolic methods are generally considered less performing than functional ones; *(ii)* are less widespread in the engineering community; *(iii)* and do not have native capabilities to deal with temporal data. In the recent literature, however, *decision trees* and *random forests*, which are among the most typical and better-known symbolic learning methods, have been enhanced with *dimensional* (e.g., *temporal*) capabilities, making them able to deal, in a native way, with temporal data [18,19]. Combined with specific feature extraction methods such as [20,21], one is now able to devise a systematic, statistical treatment of pure temporal data such as those that emerge from recording operating values of gas turbines, and highlight which variables, if any, have a role in trip prediction, and how such a role can be described. The nature of the method inspires a systematic statistical pre-analysis of the data; as a matter of fact, by closely looking into how the different variables behave from the point of view of the variance of their statistical features, we can select the most informative ones in a novel way.

In this paper, we consider several measurable gas path variables recorded in a fleet of Siemens gas turbines located worldwide, and previously used by the same authors in other works [22–24]. Unlike previous attempts, we treat such data in the context of a regression problem, and we approach the question of predicting how close we can expect to be from the next trip event, given the current behavior of a turbine. We develop several temporal decision tree models, and we use them to extract *rules* that, to some extent, give us a *warning* situation. This is made possible by the symbolic nature of the method, which had not been applied before to predictive maintenance problems; symbolic learning allows the creation of *white-box* models, that can be further investigated, interpreted, and used for knowledge extraction. Our approach is also associated with a systematic exploration of the statistical properties of the variables, that allows us to identify the most informative ones and the best tests that should be applied to extract such an information; to the best of our knowledge, this is the first time that similar techniques are applied to the area of predictive maintenance in the industry. This paper is structured as follows. First, we analyze the data from a temporal series point of view in order to highlight their most important aspects, and we perform a series of systematic statistical tests that allow us to pinpoint the most informative variables and statistical measures. Then, we describe the learning methodology that we use in our simulations, and, subsequently, we describe both the simulation setting and the results. Finally, we discuss a general framework for the application of the presented methodology and we test a model on data taken from six different turbines located on a different continent. Finally, we draw some conclusions.

Statistical Analysis of Data

The case study considered in this paper consists of 52 recordings of trip events and the respective measured variables (22 in total) taken during twenty-four hours of operation before trip occurrence. Such data were taken from 4 different gas turbines that belong to Siemens' fleet, which in the following will be referred to as turbine 1–4. Each recording consists of the values of 25 variables per minute, therefore entailing 1440 tuples of 25 values per recording. Each recording ends with a trip event; we use this data to model the problem of establishing *how far away* a certain operational time point is from the trip event. The considered variables, which are all raw measured values during gas turbine operation, are described in Table 1.

We first model the problem as a regression problem, using the distance (in minutes) from the trip event as target variable. To this end, we exclude the last 10 min before the trip event. During this 10-minute time frame, the trip event is already occurring, and the values of all variables change abruptly, therefore undermining any learning step. While the time point of the likely onset of trip symptoms can vary, as demonstrated in Ref. [25], the actual development of the trip event takes place in a very short period of

Table 1 Description of the variables used in this work

Symbol	Variable
AMB_H	Ambient air humidity
AMB_T	Ambient air temperature
GAS_P	Gas fuel valve position
GAS_F	Gas fuel mass flow rate
IGV_P	IGV compressors position
CD_T	Compressor outlet temperature
CD_P	Compressor outlet pressure
SPEED	Rotational speed
POWER	Power output
EX_T1	Exhaust temperature thermocouple 1
EX_T2	Exhaust temperature thermocouple 2
EX_T3	Exhaust temperature thermocouple 3
EX_T4	Exhaust temperature thermocouple 4
EX_T5	Exhaust temperature thermocouple 5
EX_T6	Exhaust temperature thermocouple 6
EX_T7	Exhaust temperature thermocouple 7
EX_T8	Exhaust temperature thermocouple 8
EX_T9	Exhaust temperature thermocouple 9
EX_T10	Exhaust temperature thermocouple 10
EX_T11	Exhaust temperature thermocouple 11
EX_T12	Exhaust temperature thermocouple 12
EX_T13	Exhaust temperature thermocouple 13
EX_T14	Exhaust temperature thermocouple 14
EX_T15	Exhaust temperature thermocouple 15
EX_T16	Exhaust temperature thermocouple 16

time. Thus, a 10-minute truncation does not introduce any bias and contributes to data uniforming. From the remaining 1430 points of each recording, we extract 138 one-hour series by sliding a moving window backward from the last usable value, with a step of 10 min. We, therefore, obtain a dataset of 7176 one-hour multivariate time series of 60 points each; in the following, they will be referred to as *instances*. Then, we separate the numeric classes into two bins: more than 4 h (included) from the event (class *far from the event*, denoted *F*), and less than 4 h from the event (*close to the event*, *C*); such a separation has been performed by computing the length of the interval that lies between the last point of the considered one-hour series and the point 1430, and the limits have been decided after a preliminary study of the informative content of the possible ones and taking into account a minimal amount of operative time in case of upcoming trip prediction. As a consequence, we derive a binary classification problem of discerning whether the trip event is close (*C*) or not (*F*). Table 2 shows the number of instances for the two classes for each of the turbines. Canonical preliminary data analysis approaches must be generalized to the temporal component to be applied to this case. Such a generalization gives rise to an analysis protocol that starts by identifying the most informative measures that can be possibly applied to time series (see Ref. [20]). Consider the measures shown in Table 3; the purpose of this step is to identify: *(i)* the most informative variables, and *(ii)* the most informative measures. We proceed in two phases.

- First, we consider the whole dataset, and we evaluate the informative content of each variable and each statistical measure. To this end, we first compute every measure shown in Table 3 on every variable of every instance. Then, for each measure, we consider its variance in the dataset, and we aggregate the variables using the mean among all variances; to this end, variances are first normalized. The resulting set is sorted by average variance, and we select the top-five variables, which we interpret as the most informative variables. Second, limited to the chosen variables, we compute the variance of each of the measures, and we aggregate the results using their mean. Once again, this allows us to select the five measures that show more variance on the selected variables (and, as before, we interpret them as the most informative measures). Clearly, selecting a subset of variables and measures is a necessary step in order to get a response in a reasonable time.
- Second, we analyze the values of the chosen measures and the chosen variables across the two datasets that emerge from the two classes; we compute their statistical distribution by running a normality test; then, we run a comparison test in each group to establish whether there is a significative difference between the two classes.

Time series classification can be approached in several ways. Methods for classifying time series can be roughly separated into *symbolic* and *functional*. Symbolic methods aim to extract a logical characterization of the classes in terms of the behavior of the series, while functional ones approach the classification problem by extracting a mathematical function of the series. Time series classification methods can also be separated into *native* or *feature-based*. Native methods consider time series as they are, without performing any modification of the signals. Feature-based methods, on the other hand, focuses on extracting statistically interesting measures of the signals and using those for the classification phase. Feature-based methods are far more common, and they are both symbolic and functional; their major drawback is the lack of interpretability of the results in the functional case and the low predictive capabilities in the symbolic one. Native methods are more scarce, and most common ones among them, that is, distance-based methods, do not offer, in general, a real grasp of the underlying problem, despite their general good behavior in terms of performances.

Table 2 Number of instances for the four turbines

Turbine	# inst. for *C*	# inst. for *F*	# inst. (tot)
1	288	1368	1656
2	384	1824	2208
3	264	1254	1518
4	312	1482	1794

In Refs. [18,19], a new class of symbolic, native time series classification methods was proposed. Despite their short history, *temporal decision trees* showed a good compromise between interpretability and performance. The key points that define temporal decision trees are:

- They follow the general pattern and schema of conventional decision trees. Decisions are taken on a dataset in order to maximize the amount of *information gain* in a greedy fashion, starting from the original training dataset and obtaining, at each step, smaller, and more informative subsets. When the dataset associated with a node is too small, or too pure in terms of class, it is converted into a leaf, and labeled with the majority class (generating, as in the classical case, a certain amount of misclassifications). Classical techniques, up to and including *pre- and post-pruning*, can be applied, at least in a limited form, to propositional and temporal decision trees alike.
- Unlike conventional decision trees, decisions are relativized to intervals of the time series. So, while conventional decision trees treat time series by extracting features from them, and then taking decisions on such features, temporal decision trees take decisions directly on time series, in a native way. Consider, for example, the mean; while a conventional decision tree may separate the dataset using the fact that the

Table 3 Twenty-five statistical measures for time series (including 22 measures from Ref. [20])

Measure	Symbol
Mean	*M*
Max	*MAX*
Min	*MIN*
Mode of *z*-scored distribution (5-bin)	*Z5*
Mode of *z*-scored distribution (10-bin)	*Z10*
Longest period of cons. values above the mean	*C*
Time int. between success. extr. ev. above the mean	*A*
Time int. between success. extr. ev. below the mean	*B*
First $1/e$ crossing of autocorrelation function	*FC*
First minimum of autocorrelation function	*FM*
Tot. power in lowest 1/5 of freq. in the Fourier p.s.	*TP*
Centroid of the Fourier power spectrum	*CE*
Mean error from rolling 3-sample mean forec.	*ME*
Time-reversibility statistic	*TR*
Automutual information ($m=2$, $\tau=5$)	*AI*
First minimum of the automutual information fun.	*FMAI*
Proportion of successive differences ex. 0.04	*PD*
Longest period of successive incremental decreases	*LP*
Entropy of two successive letters	*EN*
Change in correlation length	*CC*
Exponential fit to successive distances	*EF*
Proportion of slower timescale fluct. with DFA	*FDFA*
Proportion of slower timescale fluct with lin. fits	*FLF*
Trace of covariance	*TC*
Periodicity measure	*PM*

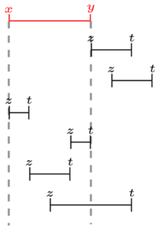
symbol	Allen's relation	graphical representation
$\langle A\rangle$	$[x,y]R_A[z,t] \Leftrightarrow y = z$	
$\langle L\rangle$	$[x,y]R_L[z,t] \Leftrightarrow y < z$	
$\langle B\rangle$	$[x,y]R_B[z,t] \Leftrightarrow x = z, t < y$	
$\langle E\rangle$	$[x,y]R_E[z,t] \Leftrightarrow y = t, x < z$	
$\langle D\rangle$	$[x,y]R_D[z,t] \Leftrightarrow x < z, t < y$	
$\langle O\rangle$	$[x,y]R_O[z,t] \Leftrightarrow x < z < y < t$	

Fig. 1 Allen's interval relations and their notation in temporal decision trees

mean of a specific variable on the whole time period exceeds a given threshold value (e.g. *if the mean value of a variable is more than that value, then…*), a temporal decision tree may do so using the existence of an interval in which the mean of a specific variable exceeds the same threshold value (e.g. *if the mean value of a variable is more than that value between the instants x and y, then…*).

- Like conventional decision trees, a temporal decision tree has a clear logical interpretation but makes use of a more complex logic than propositional logic, which allows one to express properties over intervals and their relations. There are thirteen relations between two intervals, known as *Allen's* relations (see Fig. 1, in which we show only the six *direct* relations of the type $\langle X\rangle$; their inverses, denoted with $\langle\bar{X}\rangle$, can be obtained by switching the roles of each interval, and the thirteenth, *equals*, is denoted $\langle=\rangle$), and a temporal decision tree is able to learn interval patterns which we can formalize using suitable symbols to denote Allen's relations (see Fig. 1, first column).

In Refs. [18,19] it was shown that temporal decision trees perform better than their propositional counterparts, and, while retaining a very high level of interpretability, are able to extract classification models that are comparable with those extracted by noninterpretable approaches.

Results

The results of the preliminary statistical analysis, shown in Figs. 2 and 3, reveal that the ambient air humidity (*AMB_H*) and temperature (*AMB_T*), and the rotational speed (*SPEED*), followed by the exhaust temperature of thermocouples 5 and 4 (*EX_T5*, *EX_T4*) are the most informative of the 25 variables in consideration. Furthermore, the 5 most informative measures for these 5 variables are *MIN*, *FM*, *MAX*, *M*, and *C*. Recall that, as we have explained in the previous section, this situation emerges by comparing the level of variance of each variable and each measure against each other. It appears that some of the 5 measures are more discriminative when paired with specific variables, as opposed to others, as it can be seen in Fig. 3. However, during the learning phase, each of the selected 5 measures is paired with all 5 variables, giving rise to 25 different pairs, constituting 25 statistically relevant variable descriptors. Figure 4 shows a modellization of the distribution for the two classes, obtained with Julia's StatsPlots package [26]. From a qualitative point of view, two findings arise: *(i)* none of the samples seem to follow a normal distribution, and *(ii)* there is little statistical difference between the two classes, which suggests the difficulty of the considered classification problem. To quantitatively evaluate these results, statistical hypothesis testing is applied. Shapiro-Wilk tests support the non-normality hypothesis for all samples; furthermore, for each measure, a two-sided Mann-Whitney U-Test [27] is performed to test the hypothesis that samples from the two classes are drawn from the same distribution. The p-values for the latter test are reported in Table 4, and reveal that a statistically significant difference occurs in more than half of the pairs variable-measure (recall that, for the Mann Whitney U-Test, p-value lower than 0.05 is considered as a statistically significant difference).

Based on the findings of the statistical analysis and selection process of variables and measures, several simulations of temporal decision tree training and testing are run. In order to both reduce training time and achieve higher performance, each of the 60-points series is reduced via a moving average filter, which partitions the series into chunks of equal size s along the time axis, and aggregates each chunk by computing the average, ultimately producing a series of $\frac{60}{s}$ points. After a preliminary study in which different decision tree parametrizations are tested, two pruning limits were fixed, namely, a minimum number of 4 instances at the tree leaves, and a minimum entropy gain of 0.015 when selecting the best split condition at any internal node. The simulations are run using a variant of the typical cross-validation setting. In order to prevent data leakage and ensure a fair evaluation of performances, for each simulation, data from a single turbine is used for either training or testing, but not both. More specifically, for each parametrization the following protocol is adopted: 4 simulations are run, and each time the data from a single turbine is used for testing, while data from the other turbines, subsequent to a downsampling step ensures that the classes are balanced, is used for training. Table 5 displays some properties of the trained models, the training time required, and the performance obtained on the relative test data, for those simulations that achieved the most relevant results. The performance itself is measured in terms of κ coefficient (which relativizes the overall accuracy to the probability of a random correct answer), overall accuracy (OA), average accuracy (AA), sensitivity, specificity, precision (or positive predictive value, PPV), negative predictive value (NPV) and F1-score. It should be recalled that class C is considered as the positive class, while class F is considered as the negative class; as such, sensitivity and specificity represent the ability of the model to detect the presence or absence of trip events in the subsequent 4 h, respectively. Additionally, the size of each tree can be assessed via the number of nodes, leaves, and via the tree's height. Note that the test sets, always consisting of instances from a single turbine, display a high imbalance, with only 17% of the instances belonging to the positive class and 83% belonging to the negative class.

The results after the first round of simulations show that the overall accuracy, averaged over the four executions, is 73%. These results should be interpreted, however, taking into account the imbalance among classes; the average accuracy, which normalizes it in this sense, is about 64%–65%. The κ coefficient is a measure of "how well" the model has learned from the data (a value of 0 would mean that the model has the same predictive capacity of a random choice): on average, we reach 23%, which shows that we were able to extract, at least, *some* information from the data. Our ability to distinguish the class C is lower than the one to distinguish the class F, which means that the models are more likely to extract good rules to *exclude* an upcoming trip event than to *warn* against one. This first round of simulations varying s also shows how turbine 4 behaves differently from the others: when turbine 4 is used for testing, a drop in performance occurs. More specifically, sensitivity experiences a dramatic drop from an average of 58% to 32% and precision from an average of 35% to 24%. This is likely due to the trip event being caused by different factors in the recordings from turbine 4. Native temporal data mining systems such as the ones used in this paper search for temporal patterns; even if trip events in turbine 4 are similar to those in the other turbines, and could be detected with similar performances, they may show different patterns, thus worsening the accuracy of the entire system. When turbine 4 is ignored, the training phase is able to produce models with higher sensitivity, as well as models that are similar in performance, but more concise (i.e., with a lower number of nodes and leaves; observe that the smaller the tree, the more interpretable is). Table 5 shows the results for two representative parametrization, with $s = 15$ and $s = 12$, respectively. The first parametrization yields smaller trees

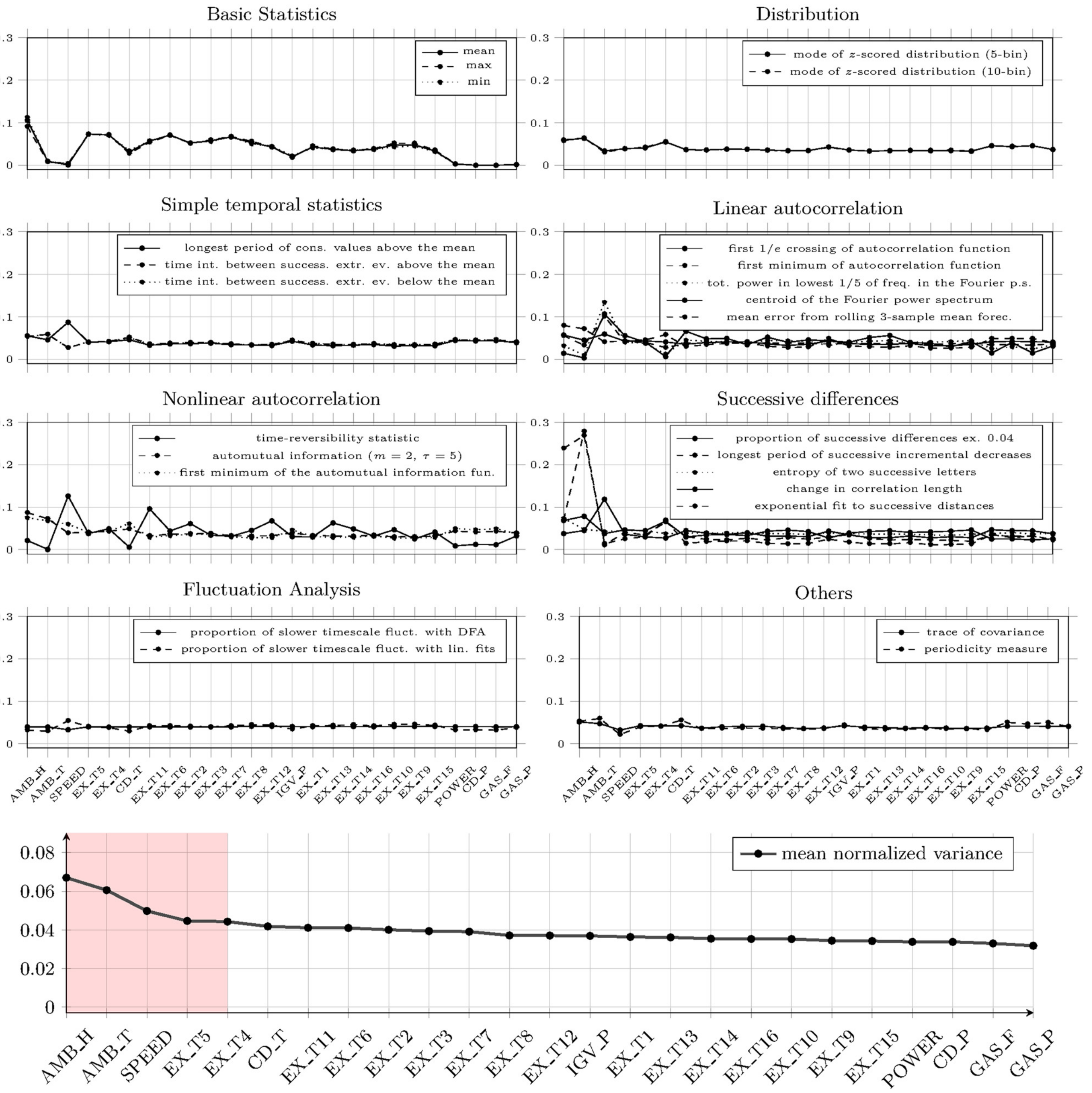


Fig. 2 Normalized variance of each measure by variable, for each group of measures (top 8 plots), shown in descending order by their mean value (bottom plot)

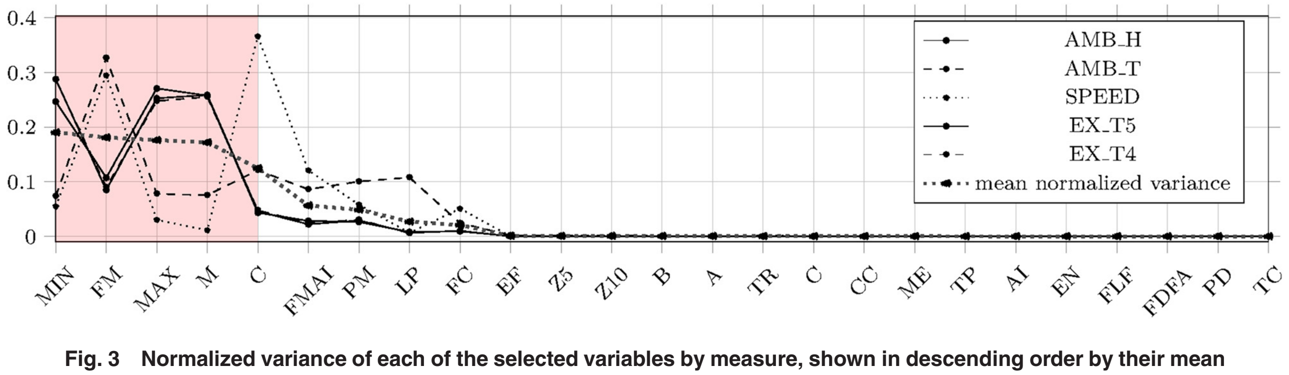


Fig. 3 Normalized variance of each of the selected variables by measure, shown in descending order by their mean

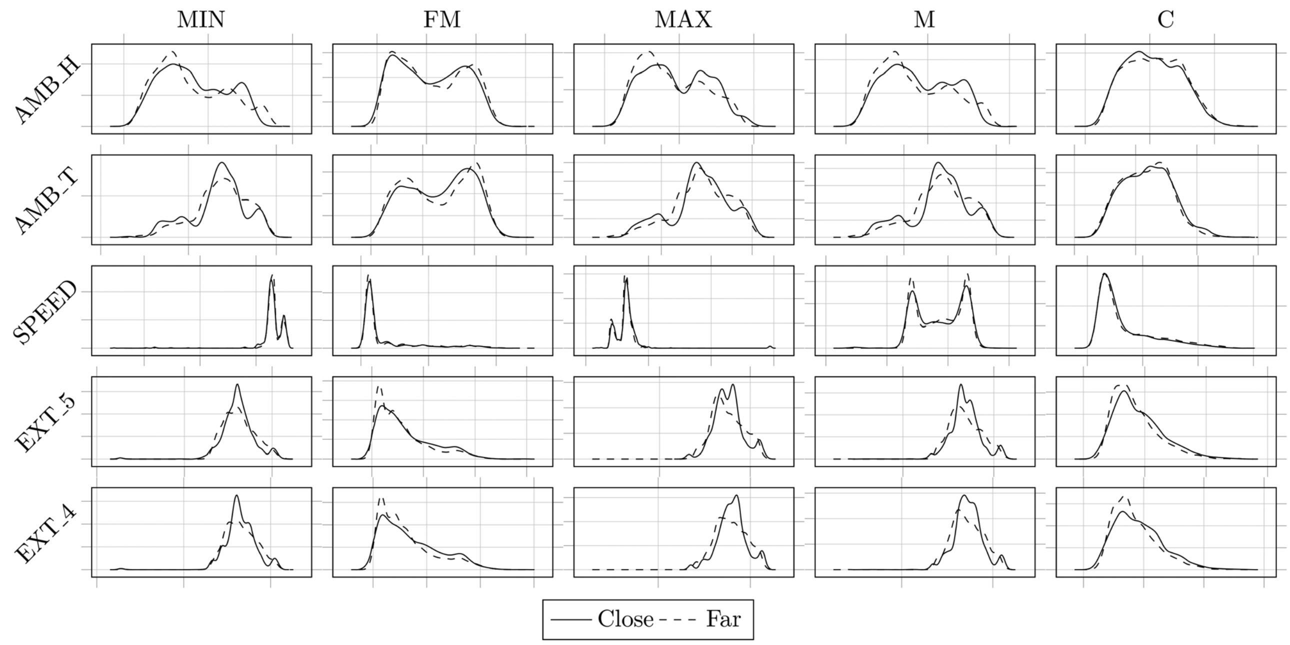


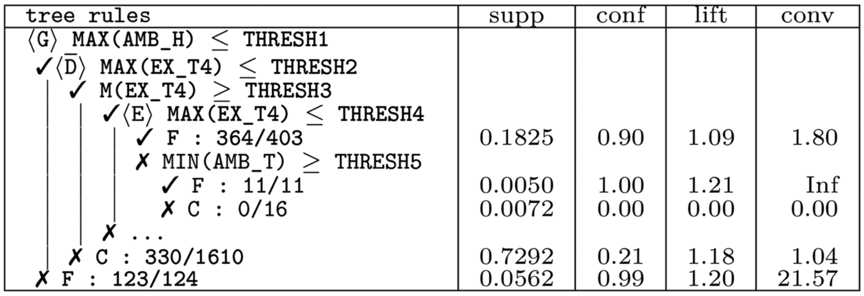
Fig. 4 Distribution of the selected variables by measure

Table 4 *p*-values for the Mann Whitney U-Test (*p*-values lower than 0.05 show a statistically significant difference)

	Min	FM	Max	M	C
AMB_H	$\mathbf{1.07 \cdot 10^{-2}}$	0.13	$\mathbf{2.24 \cdot 10^{-6}}$	$\mathbf{4.12 \cdot 10^{-4}}$	$\mathbf{1.69 \cdot 10^{-2}}$
AMB_T	$\mathbf{2.75 \cdot 10^{-3}}$	0.89	0.58	0.21	$\mathbf{2.56 \cdot 10^{-2}}$
SPEED	$\mathbf{1.23 \cdot 10^{-2}}$	$8.9 \cdot 10^{-2}$	$\mathbf{1.87 \cdot 10^{-7}}$	0.91	0.15
EXT_5	0.71	$\mathbf{5.6 \cdot 10^{-8}}$	$\mathbf{8.05 \cdot 10^{-3}}$	0.11	$\mathbf{7.47 \cdot 10^{-8}}$
EXT_4	0.81	$\mathbf{2.7 \cdot 10^{-6}}$	$\mathbf{4.55 \cdot 10^{-2}}$	$6.53 \cdot 10^{-2}$	$\mathbf{1.69 \cdot 10^{-8}}$

Table 5 Test results obtained by the trained classification models

	Test turbine	κ (%)	OA (%)	AA (%)	sens (%).	spec (%).	PPV (%)	NPV (%)	F1 (%)	# nodes	# leaves	height	time (s)
$s = 4$, 4 turbines	1	36	79	71	60	83	42	91	49	275	138	21	821
	2	20	71	62	48	77	30	87	37	203	102	17	686
	3	28	71	69	66	73	33	91	44	251	126	18	801
	4	9	71	55	32	79	24	85	27	213	107	16	823
	avg.	23.38	73.09	64.5	51.31	77.68	32.43	88.44	39.54	235.5	118.25	18	782.61
$s = 15$, 3 turbines	1	11	51	61	76	45	23	90	35	69	35	13	202
	2	**6**	**38**	**57**	**87**	**28**	**20**	**91**	**33**	**15**	**8**	**5**	**86**
	3	21	68	65	60	70	29	89	39	185	93	21	186
	avg.	12.78	52.18	60.94	74.37	47.51	24.05	90.09	35.7	89.66	45.33	13	157.80
$s = 12$, 3 turbines	1	21	70	64	54	73	30	88	38	209	105	19	249
	2	**28**	**75**	**67**	**56**	**79**	**35**	**89**	**43**	**177**	**89**	**15**	**141**
	3	15	65	61	56	66	26	88	36	145	73	15	170
	avg.	21.32	69.67	64	55.32	72.69	30.38	88.5	39.08	177	89	16.33	186.66



```
tree rules                                supp     conf    lift    conv
⟨G⟩ MAX(AMB_H) ≤ THRESH1
 ✓⟨D̄⟩ MAX(EX_T4) ≤ THRESH2
 │  ✓ M(EX_T4) ≥ THRESH3
 │  │  ✓⟨E⟩ MAX(EX_T4) ≤ THRESH4
 │  │  │  ✓ F : 364/403               0.1825   0.90    1.09    1.80
 │  │  │  ✗ MIN(AMB_T) ≥ THRESH5
 │  │  │     ✓ F : 11/11              0.0050   1.00    1.21    Inf
 │  │  │     ✗ C : 0/16               0.0072   0.00    0.00    0.00
 │  │  ✗ ...
 │  ✗ C : 330/1610                    0.7292   0.21    1.18    1.04
 ✗ F : 123/124                        0.0562   0.99    1.20    21.57
```

Fig. 5 Decision tree τ_1. Three rules and paths with high confidence and/or support are highlighted.

(average of 45 leaves, compared with 118 in the previous round) with lower precision and F1-score, but a much higher sensitivity (average of 74%, compared with 51% in the previous round). The second parametrization yields trees with higher specificity and precision, namely, that perform better at detecting the absence of a trip event. The different behavior obtained with different but similar choices of s should be further investigated. From the first and second parametrization, we select and consider the two trees with highest sensitivity and precision, respectively, and denote them as τ_1 and τ_2. The rows relative to the two selected trees are

```
 tree rules                                      supp    conf   lift    conv
⟨G⟩ MAX(AMB_H) ≤ THRESH6
 ✓⟨D̄⟩ MAX(EX_T4) ≤ THRESH7
   ✓⟨Ē⟩ MIN(EX_T4) ≥ THRESH8
     ✓ F : 362/398                               0.1803  0.91   1.10    1.92
     ✗ ...
   ✗⟨Ō⟩ C(AMB_T) ≥ THRESH9
    ✓ MAX(AMB_T) ≤ THRESH10
      ✓⟨A⟩ MIN(AMB_H) ≥ THRESH11
         ✓⟨L̄⟩ MAX(EX_T5) ≤ THRESH12
           ✓⟨B̄⟩ M(SPEED) ≤ THRESH13
              ✓ F : 137/158                      0.0716  0.87   1.05    1.31
              ✗ ...
           ✗ ...
         ✗ MIN(EX_T5) ≥ THRESH14
          ✓ ...
          ✗ F : 25/25                            0.0113  1.00   1.21     Inf
      ✗⟨A⟩ MIN(AMB_H) ≥ THRESH15
       ✓⟨L̄⟩ MIN(SPEED) ≥ THRESH16
         ✓⟨D̄⟩ MIN(SPEED) ≥ THRESH17
            ✓ C : 6/14                           0.0063  0.43   2.46    1.45
            ✗ ...
         ✗ C : 12/20                             0.0091  0.60   3.45    2.07
       ✗ MIN(AMB_H) ≥ THRESH18
       ✓⟨E⟩ MIN(AMB_H) ≥ THRESH19
         ✓⟨B̄⟩ MAX(AMB_T) ≤ THRESH20
            ✓ ...
         ✗ F : 39/39                             0.0177  1.00   1.21     Inf
       ✗⟨O⟩ MIN(SPEED) ≥ THRESH23
       ✓ F : 28/29                               0.0131  0.97   1.17    5.04
       ✗ ...
       ✗⟨B⟩ MIN(EX_T5) ≥ THRESH27
       ✓⟨A⟩ MAX(SPEED) ≤ THRESH28
         ✓⟨L̄⟩ MAX(SPEED) ≤ THRESH29
            ✓ ...
            ✗ C : 10/36                          0.0163  0.28   1.60    1.14
         ✗ F : 59/61                             0.0276  0.97   1.17    5.30
       ✗ F : 58/69                               0.0312  0.84   1.02    1.09
    ✗⟨A⟩ MIN(EX_T5) ≥ THRESH30
     ✓ MAX(AMB_H) ≤ THRESH31
       ✓ C : 22/22                               0.0100  1.00   5.75     Inf
       ✗ F : 16/16                               0.0072  1.00   1.21     Inf
     ✗ MAX(EX_T4) ≤ THRESH32
      ✓⟨Ā⟩ MIN(AMB_H) ≥ THRESH33
        ✓⟨Ē⟩ MIN(AMB_T) ≥ THRESH34
           ✓ C : 17/23                           0.0104  0.74   4.25    3.17
           ✗⟨Ō⟩ MIN(AMB_H) ≥ THRESH35
            ✓ C : 14/15                          0.0068  0.93   5.37   12.39
            ✗⟨B̄⟩ ...
        ✗ MIN(EX_T4) ≥ THRESH40
         ✓ MIN(EX_T4) ≥ THRESH41
           ✓ MAX(AMB_T) ≤ THRESH42
              ✓ MIN(AMB_T) ≥ THRESH43
                ✓ MIN(AMB_H) ≥ THRESH44
                   ✓ C : 17/36                   0.0163  0.47   2.72    1.57
                   ✗ ...
                ✗ ...
              ✗⟨L̄⟩ MAX(AMB_T) ≤ THRESH45
               ✓ ...
               ✗ MIN(EX_T4) ≥ THRESH46
                ✓ C : 10/30                      0.0136  0.33   1.92    1.24
                ✗ ...
           ✗ F : 46/51                           0.0231  0.90   1.09    1.77
         ✗ ...
      ✗ FM(AMB_T) ≤ THRESH51
       ✓⟨E⟩ MAX(AMB_H) ≤ THRESH52
         ✓ ...
         ✗⟨B⟩ MAX(AMB_T) ≤ THRESH53
          ✓ MIN(AMB_T) ≥ THRESH54
            ✓ F : 48/50                          0.0226  0.96   1.16    4.35
            ✗ ...
          ✗ ...
       ✗ F : 62/62                               0.0281  1.00   1.21     Inf
 ✗ F : 119/119                                   0.0539  1.00   1.21     Inf
```

Fig. 6 Decision tree τ_2. Three rules and paths with high confidence and/or support are highlighted.

highlighted in bold in Table 5. Incidentally, both trees are obtained by using turbine 2 for testing and turbine 1 and 3 for training, but they cover different aspects of providing *good* classifications. The two trees are displayed in textual and graphical form in Figs. 5 and 6, respectively. Rows in the figures either correspond to an inner decisional node, or to a leaf, where the test instances are routed to and finally classified. Note that thresholds have been made anonymous and denoted as *THRESH*1, *THRESH*2, etc., and that some uninteresting parts of the trees have been ellipsed for a clearer presentation. Any tree leaf corresponds to a classification rule for one of the two classes and can be evaluated considering the number of instances routed to the leaf n, the number of instances that the leaf correctly classifies c, the total number of instances in the test set t, and the number of instances that belong to that class, denoted here as t_{class}, where *class* is either C or F. Recall from Table 2 that $t = 2208$, with $t_C = 384$ and $t_F = 1824$. Typical performance metrics for rule extraction are support (Eq. (1)), confidence (Eq. (2)), lift (Eq. (3)), and conviction (Eq. (4))

$$\text{support} = \frac{n}{t} \tag{1}$$

$$\text{confidence} = \frac{c}{n} \tag{2}$$

$$\text{lift} = \frac{c}{n \cdot t_{\text{class}}} \tag{3}$$

$$\text{conviction} = \frac{1 - t_{\text{class}}}{1 - \frac{n}{t}} \tag{4}$$

Focusing on τ_1 (Fig. 5), a few considerations can be made. First, there exists a simple rule (r_1) for excluding the case of class C, that applies to about 5%–6% of the cases, which can be paraphrased as *if there is not an interval where the ambient air humidity is lower than THRESH1, then the trip event is farther than 4 h with a 99% confidence*. Another interesting rule (r_2) states that *if there exists an interval where the ambient air humidity is lower than THRESH1, as well as another larger interval that contains the first, where the exhaust temperature of thermocouple 4 is (i) on average higher than THRESH3, but (ii) always less than THRESH2, and (iii) less than THRESH4 in the final part of the interval, then the trip event is farther than 4 h with a 90% confidence*. This second rule is more articulated, and has a lower confidence than the first. However, it covers 18% of the test instances, and 364 out of 1824 test instances are correctly classified by this rule, which means that this rule is responsible for a 20 of the 28 percentage points of specificity (note that another 7 points are covered by r_1). Rules similar to r_1 and r_2, but with higher confidence and slightly lower support were extracted when using a different moving average filter; in fact, τ_2, shown in Fig. 6, provides a variant of r_1 that uses a stricter threshold (*THRESH*6) for the ambient air humidity and achieves 100% confidence with a nearly equal support. Additionally, it provides a variant r_2 that states that *if there exists an interval where the ambient air humidity is lower than THRESH6, as well as another larger interval that contains the first, where the exhaust temperature of thermocouple 4 is (i) always less than THRESH7, (ii) but higher than THRESH8 just before, and throughout, the whole interval, then the trip event is farther than 4 h with a 91% confidence*. As for class C, τ_1 provides the following rule: *if there exists an interval where the ambient air humidity is lower than THRESH1, but there does not exist an interval satisfying the properties presented in r_2, then the trip event is closer than 4 h with a 21% confidence*. The peculiarity of this rule lies in the fact that 330 out of 384 instances for C are correctly classified; this entails that if a trip event is about to occur, it can be correctly predicted 86% of the time by solely applying this rule. While it is true that further investigation on rules, their semantics, and their significance is necessary, it is worth observing that, in this approach, rules can work together even if they are extracted with different processes and in different moments; each rule adds new information, effectively improving the scalability of the whole system.

Applicability and Robustness

In order to highlight the ability of the presented method to deal with new operational data, Fig. 7 presents a generalized workflow for applying temporal decision trees to trip prediction tasks on new gas turbines. The process starts with sensor data collection, and proceeds to the synthesis of time series instances (each associated with a class label), that are then reduced by a statistical analysis and feature selection step. The dataset is then used for training temporal decision trees with different parametrizations. Finally, a tree model can be synthesized by manual or automatic selection of good rules, and it can be deployed as a trip event warning system. As new data is collected, newer models can be trained and

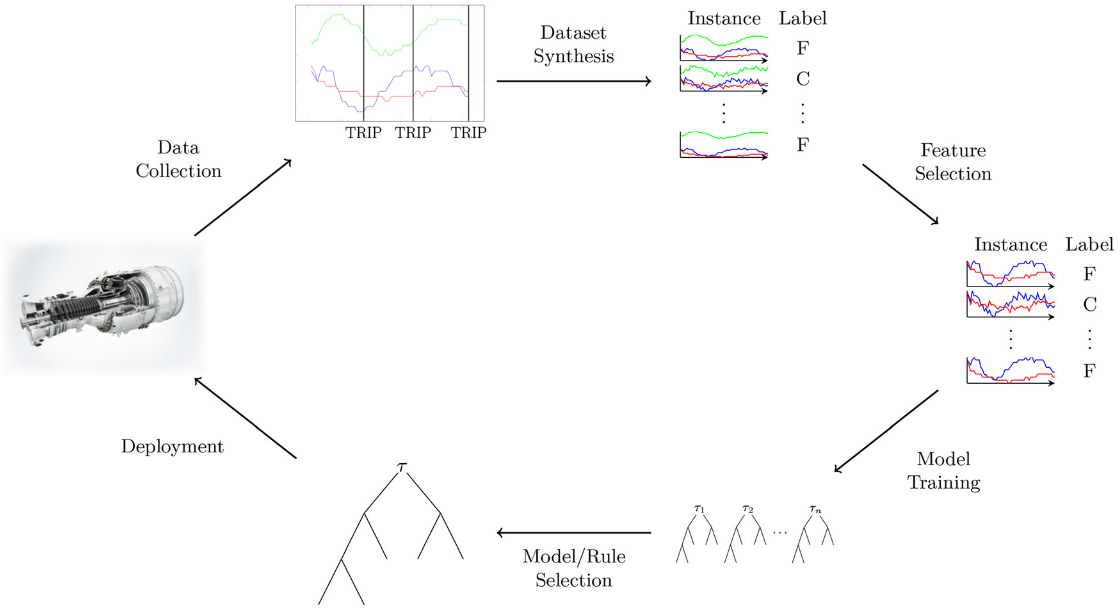


Fig. 7 A workflow of the proposed method

synthesized, making the process robust to changes in time (e.g., deterioration of a turbine's inner component).

A different question concerns the robustness of the obtained models; while a model deployed on a given turbine is expected to yield higher performances when trained on data from the same turbine, testing a model on data taken from an entirely different environment allows one to estimate its generalization abilities. To this end, we performed a final external test by applying both decision trees τ_1 and τ_2 on data from six different turbines located on a different continent. The new dataset includes the recordings of 42 trip events, each with length of twenty-four hours. The observed change in overall accuracy was from 38% to 43% and from 75% to 65%, respectively. Interestingly, τ_1 decreases its sensitivity from 87% to 75%, but the gained test accuracy is due to a higher specificity, which increases from 28% to 36%, suggesting that the rules for class F may be more compelling than expected. As for τ_2, both sensitivity and specificity experience a decrease, respectively from 56% to 36% and from 79% to 71%, but, at least in the latter case, such a degradation of performance can be considered relatively contained. While more specific research must be performed in order to assess the performance change for the single rules, we can conclude that our results are a solid starting point for further research toward AI-driven trip prediction.

Conclusions

In this paper, we considered several measurable gas path variable data recorded in a fleet of four Siemens gas turbines located worldwide, and previously used by the same authors in other works, and approached the question of predicting a trip event. We treated data as multivariate time sequences, where each sequence is the recording of the selected variables during one hour of activity of a turbine, and we labeled each sequence with the amount of time until the next trip event, to predict if a particular sequence, corresponding to one hour of recording, or, more precisely, a particular time point, is far (more than 4 h) or close (less than 4 h) to a possible trip event. We applied a new strategy and technique to obtain our results; first, we designed a complex, but reproducible, statistical analysis to identify the most informative variables and the most informative statistical measures on such variables; then, we asked the question of whether there is, in fact, a statistically significant difference (in terms of the selected variables and measures) between the two cases (far or close to trip); finally, we applied a novel machine learning technique, called temporal decision trees, to the resulting dataset. As a result, we identified five variables that play an important role in trip prediction (air humidity and temperature, rotational speed, plus the exhaust temperature of two particular thermocouples), and we were able to extract a classifier with an overall accuracy of 73% (which becomes approximately 65% when the numerosity of the classes are taken into account). We also identified one turbine that behaves, apparently, in a clearly different way from the others, and, by excluding it, we extracted more precise classifiers in which we identified interesting rules. We analyzed such rules (which have small support but very high confidence, over 90% in some cases) and we concluded that they may provide interesting insights into the turbines' behavior, useful to design, at the very least, a *warning system* that may supervise the everyday operations. Finally, the applicability of our method to different environments has been evaluated using data from a completely different fleet of turbines, with promising results. As a continuation of this work, we plan on applying the same methodology to data from sensors that measure vibrations; in fact, it is known that excessive vibration levels can lead to trip events, therefore we expect that this kind of data can help improve the accuracy of the models.

Acknowledgment

The authors gratefully acknowledge Siemens for the permission to publish the results. The authors also wish to express sincere gratitude to Dr. Giuseppe Fabio Ceschini for inspiring and supporting the research activity about gas turbine trip prediction.

Data Availability Statement

Data provided by a third party listed in Acknowledgements.

Nomenclature

AA	=	average accuracy
C	=	class *close from the event*
c	=	number of instances that a leaf correctly classifies
F	=	class *far from the event*
n	=	number of instances routed to a leaf
NPV	=	negative predictive value
OA	=	overall accuracy
PPV	=	positive predictive value
r	=	a rule
s	=	window size for the moving average filter
t	=	number of instances in the test data set
$\langle X \rangle$	=	a binary relation
τ	=	a temporal decision tree

References

[1] Bhargava, R., 2017, *Technical Dictionary on the Gas Turbine Technology*, Innovative Turbomachinery Technologies Corp, TX.

[2] Morini, M., Pinelli, M., and Venturini, M., 2007, "Development of a One-Dimensional Modular Dynamic Model for the Simulation of Surge in Compression Systems," ASME J. Turbomach., **129**(3), pp. 437–447.

[3] Morini, M., Pinelli, M., and Venturini, M., 2009, "Analysis of Biogas Compression System Dynamics," Appl. Energy, **86**(11), pp. 2466–2475.

[4] Pezzini, P., Tucker, D., and Traverso, A., 2013, "Avoiding Compressor Surge During Emergency Shut-Down Hybrid Turbine Systems," ASME J. Eng. Gas Turbines Power, **135**(10), p. 102602.

[5] Kudo, M., Toyama, J., and Shimbo, M., 1999, "Multidimensional Curve Classification Using Passing–Through Regions," Pattern Recognit. Lett., **20**(11–13), pp. 1103–1111.

[6] Caiado, J., Crato, N., and Peña, D., 2006, "A Periodogram-Based Metric for Time Series Classification," Comput. Stat. Data Anal., **50**(10), pp. 2668–2684.

[7] Fulcher, B. D., and Jones, N. S., 2014, "Highly Comparative Feature-Based Time-Series Classification," IEEE Trans. Knowl. Data Eng., **26**(12), pp. 3026–3037.

[8] Malhotra, P., Tv, V., Vig, L., Agarwal, P., and Shroff, G. M., 2017, "Timenet: Pre-Trained Deep Recurrent Neural Network for Time Series Classification," Proc. of the 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, Apr. 26–28, pp. 607–612.

[9] Längkvist, M., Karlsson, L., and Loutfi, A., 2014, "A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling," Pattern Recognit. Lett., **42**, pp. 11–24.

[10] Lu, F., Ju, H., and Huang, J., 2016, "An Improved Extended Kalman Filter With Inequality Constraints for Gas Turbine Engine Health Monitoring," Aerosp. Sci. Technol., **58**, pp. 36–47.

[11] Qin, S., and Chiang, L., 2019, "Advances and Opportunities in Machine Learning for Process Data Analytics," Comput. Chem. Eng., **126**, pp. 465–473.

[12] De Giorgi, M., Campilongo, S., and Ficarella, A., 2018, "A Diagnostics Tool for Aero-Engines Health Monitoring Using Machine Learning Technique," Energy Procedia, **148**, pp. 860–867.

[13] Zhong, S., Fu, S., and Lin, L., 2019, "A Novel Gas Turbine Fault Diagnosis Method Based on Transfer Learning With CNN," Measurement, **137**, pp. 435–453.

[14] Amozegar, M., and Khorasani, K., 2016, "An Ensemble of Dynamic Neural Network Identifiers for Fault Detection and Isolation of Gas Turbine Engines," Neural Networks, **76**(01), pp. 106–121.

[15] Bai, M., Liu, J., Chai, J., Zhao, X., and Yu, D., 2020, "Anomaly Detection of Gas Turbines Based on Normal Pattern Extraction," Appl. Therm. Eng., **166**, p. 114664.

[16] Naderi, E., and Khorasani, K., 2018, "Data-Driven Fault Detection, Isolation and Estimation of Aircraft Gas Turbine Engine Actuator and Sensors," Mech. Syst. Signal Process., **100**, pp. 415–438.

[17] Wong, P., Yang, Z., Vong, C., and Zhong, J., 2014, "Real-Time Fault Diagnosis for Gas Turbine Generator Systems Using Extreme Learning Machine," Neurocomputing, **128**, pp. 249–257.

[18] Sciavicco, G., and Stan, I., 2020, "Knowledge Extraction With Interval Temporal Logic Decision Trees," Proc. of the 27th International Symposium on Temporal Representation and Reasoning (TIME), Vol. 178 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 9:1–9:16.

[19] Manzella, F., Pagliarini, G., Sciavicco, G., and Stan, I., 2021, "Interval Temporal Random Forests With an Application to COVID-19 Diagnosis," Proc. of the 28th International Symposium on Temporal Representation and Reasoning (TIME), Vol. 206 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 7:1–7:18.

[20] Lubba, C., Sethi, S., Knaute, P., Schultz, S., Fulcher, B., and Jones, N., 2019, “Catch22: Canonical Time-Series Characteristics - Selected Through Highly Comparative Time-Series Analysis,” Data Min. Knowl. Discov., **33**(6), pp. 1821–1852.

[21] Christ, M., Braun, N., Neuffer, J., and Kempa-Liehr, A., 2018, “Time Series Feature Extraction on Basis of Scalable Hypothesis Tests (Tsfresh – a Python Package),” Neurocomputing, **307**, pp. 72–77.

[22] Losi, E., Venturini, M., Manservigi, L., Ceschini, G., Bechini, G., Cota, G., and Riguzzi, F., 2021, “Structured Methodology for clustering gas turbine Transients by Means of Multivariate Time Series,” ASME J. Eng. Gas Turbines Power, **143**(3), pp. 1–13.

[23] Losi, E., Venturini, M., Manservigi, L., Ceschini, G., Bechini, G., Cota, G., and Riguzzi, F., 2021, “Data Selection and Feature Engineering for the Application of Machine Learning to the Prediction of Gas Turbine Trip,” ASME Paper No. GT2021-58914.

[24] Losi, E., Venturini, M., Manservigi, L., Ceschini, G. F., Bechini, G., Cota, G., and Riguzzi, F., 2022, “Prediction of Gas Turbine Trip: A Novel Methodology Based on Random Forest Models,” ASME J. Eng. Gas Turbines Power, **144**(3), p. 031025.

[25] Losi, E., Venturini, M., Manservigi, L., and Bechini, G., 2022, “Detection of the Onset of Trip Symptoms Embedded in Gas Turbine Operating Data,” ASME Paper No. GT2022-80666.

[26] Breloff, T., and Community, J., 2021, “Statsplots.jl, v0.14.29,” accessed Nov. 24, 2022, https://github.com/JuliaPlots/StatsPlots.jl

[27] Mann, H. B., and Whitney, D. R., 1947, “On a Test of Whether One of Two Random Variables is Stochastically Larger Than the Other,” Ann. Math. Stat., **18**(1), pp. 50–60.