ISSN 1064-2269, Journal of Communications Technology and Electronics, 2019, Vol. 64, No. 12, pp. 1473–1484. © Pleiades Publishing, Inc., 2019.
Russian Text © The Author(s), 2019, published in Informatsionnye Protsessy, 2019, Vol. 19, No. 3, pp. 249–270.

On Construction of Early Warning Systems for Predictive Maintenance in Aerospace Industry

E. V. Burnaev[a, b, *]

[a] *Skolkovo Institute of Science and Technology, Moscow, Russia*
[b] *Institute for Information Transmission Problems, Moscow, Russia*
**e-mail: E.Burnaev@skoltech.ru*

Received July 1, 2019; revised July 12, 2019; accepted July 18, 2019

Abstract—The problem of constructing predictive models for early warning systems for diagnostic maintenance in the aerospace industry is considered. A new approach to predicting rare failures based on a new methodology that takes into account the properties of technical systems and specific requirements imposed by applications is proposed.

Keywords: monitoring system, anomaly detection, failure prediction, machine learning, predictive maintenance, signal processing

DOI: 10.1134/S1064226919120027

1. INTRODUCTION

The aviation industry is one of the most regulated industries—the quality, safety and efficiency of aviation systems operation has a direct impact on people's lives and health.

In recent years, major aircraft manufacturers and airlines have come to the conclusion that through the use of predictive analytics and engineering knowledge, data collected by aircraft telemetry systems during flights, as well as other data on the life cycle of the aircraft, can be used to maintain equipment based on automatic monitoring of its condition (so-called Predictive Maintenance, or Prognostic Health Management, PHM). PHM plays an increasingly important role in ensuring reliability and efficiency—PHM methods provide opportunities to predict and prevent possible failures, reduce maintenance costs and increase fleet utilization.

PHM services can assess the current state of the equipment, determine its operational status, detect abnormal conditions in a timely manner, prevent possible forced downtime [1–3, 20, 38, 43]. The difficulties encountered when constructing PHM models in aviation are that

—aircraft telemetry data have a very complex structure: time series with a large number of components (often more than a few hundred) and a high frequency of measurements (often on the order of tens of thousands of measurements per flight); large amounts of data (the typical sample size of training data is measured in terabytes); missing observations, non-homogeneous noise; a complex hierarchical structure of the types of failures that need to be predicted;

—usually, failures are rare events, for this reason the task of forecasting is imbalanced [13, 39], that is, typical data samples contain many examples of normal system functioning and only a small number of examples of a system in an abnormal state.

In [33, 42] the authors considered autoregression-type models for predicting telemetry time series with subsequent prediction correction based on kernel regression and applied the obtained model to detect anomalies in sensor readings. In [8, 10] they considered a PHM problem for predicting the time of replacement of the exhaust valve of the ventilation control system in the aircraft. To build the models, they used both features generated from data on certain events (for example, failures of certain types) at certain moments of time, and on the basis of autoregressive models of telemetry readings. In [9] the authors considered an approach to predicting the lifetime of critical components of the aircraft air intake system based on a linear support vector machine. In [44] the authors solved the problem of detecting failures in some chemical process on the basis of standard time series modeling methods. A similar approach based on vector linear autoregression was used in [22]. In [21] approaches to online analysis of telemetry from train axis bearings based on kernel regression were developed.

In fact, in the works listed above, the following approaches are used to solve the problem of PHM of a specific technical system:

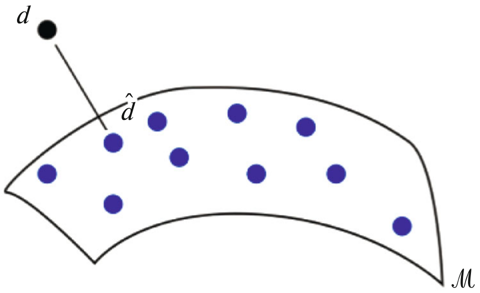


Fig. 1. The general idea of anomaly detection based on reconstruction with respect to the underlying lower dimensional data manifold.

—the problem of prediction of failures is actually "replaced" by the task of anomaly detection in a stream of telemetry data; in turn, the task of anomaly detection is solved by any standard method, often without taking into account the dependencies between the observed signals; in many cases to detect anomalies they build a linear model to forecast future time series values, and then by comparing real observations and their forecasts they make a decision about the presence/absence of anomaly;

—in case labels are known, that is, in the historical sample there is information about at what points in time failures of what type occurred, the authors usually use standard machine learning approach—they generate input features from the telemetry data and then apply supervised classification methods, e.g. random forests.

Note that both of the above "basic" approaches are generally not applicable when constructing early warning systems for diagnostic maintenance in the aerospace industry. Indeed, the number of examples of failures is usually very small, that is, the corresponding supervised classification problem is very imbalanced. At the same time, reformulating such problem as an anomaly detection problem does not always work either—the corresponding predictive model must have a low level of false alarms and at the same time be able to detect a sufficient number of failures in order to be practically useful; moreover, standard methods of anomaly detection applied to noisy high-dimensional telemetry data, will generate a large number of false alarms.

Thus, the development of an automated system for failure prediction and early warning of possible costly malfunctions is a very complex task, and requires the development of a specialized methodology that will take into account the above features of the data generated by technical systems in aviation.

In this paper we consider the problem of constrution of predictive models for early warning systems for diagnostic maintenance in the aerospace industry. The author proposes a new approach to the prediction of rare failures, based on a new methodology that allows to take into account the properties of technical systems and specific requirements imposed by applications. The application of the corresponding algorithms is illustrated on the problem of forecasting certain types of engine failures of a passenger aircraft.

2. GENERAL METHODOLOGY

There is the following obvious observation: equipment (say, aircraft engine) typically falls into pre-failure state starting with some minor flaws, e.g. cracks or leaks, that evolve in time and lead up to critical failure events such as complete engine destruction. Accordingly, emerging defects change the statistical properties of telemetry data, which can be detected by anomaly detection methods.

Thus, having this simple observation in mind, we propose the following general approach to construction of early warning systems for predictive maintenance, consisting of successive steps of telemetry data analysis of the technical system:

1. **Identification of Subsystems**: the features (report parameters) are partitioned into groups that correspond either to different subsystems within an object or into clusters, such that features are highly dependent within the clusters and almost independent in between clusters;

2. **Detection of Anomalies**: within each densely dependent group, or an identified subsystem, a method for anomaly detection is used to uncover either structural changes in dependence patterns, or simple extreme values, or any other abnormality (see Fig. 1);

3. **Alarm construction**: using simple rules of boolean logic individual simple anomaly binary series are combined into composite alarm signals;

4. **Event Matching**: every alarm is tested for predictiveness with statistical techniques designed to identify best signals which are precursory to failure or warning events, and tend not to happen anywhere but a short time prior to such events (see Section 3.2);

5. **Warning signal synthesis**: sufficiently predictive alarm signals are pooled into combined alarm, which fires when at least one constituent alarm is signalled;

6. **Leave-one-object-out validation**: the whole pipeline for predictive alarm construction is run over the dataset with all but one object and then tested against that left out object to estimate the performance with respect to the key early warning metrics (see Section 3.3).

The following sections provide necessary details for these steps.

2.1. Subsystem Identification

In the general case, the feature space may have high dimensionality and not much information may be available about its structure. In order to overcome lack of domain knowledge it is possible to apply a clustering procedure to split the feature space into tightly related groups. The approach consists of three key steps:

(1) Calculation of interdependence between each two features;

(2) Pruning of weak dependency scores and construction of an undirected graph with features as vertices and edges between sufficiently strongly dependent ones;

(3) Partitioning of the feature space based on the resulting affinity graph with either hierarchical or spectral clustering methods for community detection [37], as well as using approaches to embedding of graphs and their nodes [24].

It should be noted that in rare cases, when there is access to a detailed description of the technical system and/or there is an opportunity to get advice from the relevant engineer, clustering based on the received description is superior to automatic, but, as the results of real experiments show, usually not by a very large margin. In this paper, when processing telemetry readings from an A380 passenger aircraft, clustering was obtained based on the description of the detailed structure and measured parameters presented in the ACMS reports, see Section 4. As a result, it was possible to ideally divide the feature space into engine-related groups of four parameters each and a group of parameters associated with the aircraft as a whole.

2.2. Anomaly Detection

The general case of anomalous observation detection, based on learning of a manifold, describing dependences between features in data, is depicted in Fig. 1, see also [29–32]. There d is a sample point, $\hat{d}$ is its reconstruction, obtained through projecting and embedding based on the learned data manifold $\mathcal{M}$. In general case a data point d may be ejected from the generating parent manifold $\mathcal{M}$ either due to noise or because of a structural change in manifold, or both.

Another basic method to detect anomalies is to estimate simple outliers, or extreme values.

In general, due to the fact that in practice it is necessary to process heterogeneous data in a variety of situations, it is necessary to use different methods of anomaly detection:

—some telemetry data is represented as multidimensional time series, so it is possible to detect anomalies in sensor data streams based on methods from [5–7, 18, 23, 36], in this case, to predict rare events and improve the reliability of the result, we can build ensembles of models, see [4, 26–28, 41], using the detected anomalies and their characteristics as precursors of the main types of failures;

—when building models, it is also possible to use privileged information about future events, which is available at the stage of model training. Similar approaches used in [14, 16, 40] to detect anomalies with simultaneous selection of hyperparameters of a model, allowed to significantly improve the accuracy of the forecast;

—the sensor data contained in the training sample often have something like a spatial dimension, since the different components of the time series correspond to different nodes of the engineering system; thus, it is possible to construct a graph of dependencies between data flows recorded by different sensors, and use modern methods of feature extraction based on graph data [24, 25], as well as methods for feature extraction based on panel time series data [34, 35], in order to enrich the set of input features used to construct a predictive model;

—Another typical method of anomaly detection approach is based on constructing a surrogate model [11, 19, 45] to approximate the dependencies between the observed parameters and then detect anomalies based on the prediction error of their values using nonparametric measures to estimate uncertainty [15, 17] as an indicator of confidence in the decision taken.

2.3. Alarms Signal Synthesis

Alarms signals (about possible future failures) are constructed by pooling relevant joint anomalies together. In practice in order to maintain interpretable prediction, usually we combine alarms signals of not more than 1–3 types of anomalies. For example, if x_{0t} and x_{1t} is a pair of distinct anomaly sources at some moment t, that were labelled as jointly relevant, then the resulting alarm signal would be

$$a_t = \begin{cases} 1, \text{ if } x_{0t} = 1 \text{ and } x_{1t} = 1, \\ 0, \text{ otherwise.} \end{cases}$$

2.4. Event Matching

There are two possible approaches to the problem of alarm signal construction. The first, manual employs domain and collateral knowledge to select and combine anomalies into potentially predictive signals. The second, is automatic algorithmic selection, which extracts sufficiently frequent patterns of alarm and target event co-occurrence to perform automated construction of the best alarm signals from individual anomaly series.

The “manual” method is warranted in rare cases when there is insufficient number of occurrences of target events to be able to apply automatic method, which is frequency based.

A more elaborate exposition of the event matching algorithm is in Section 3.

2.5. Validation

Model validation is a generic, yet crucial step in the analysis. The main goal is to test the stability of predictive patterns of anomalies extracted form the data. This is achieved by estimating the necessary parameters for the selected anomaly sources on the pooled sample of all but one object, and then test run the resulting early warning system on the left out object.

3. EVENT MATCHING

In order to determine which anomalies in the dataset are predictable with respect to one or another failure event, one can employ the so-called Event matching algorithm. Essentially, this is a feature selection procedure, which takes as input

—binary features, conveniently collected in a set $\mathscr{F}$; in our case these features encode whether an anomaly of a certain type appears at some moment of time,

—targets ($\{0,1\}$-signals), and filters out those features, which fail to occur statistically more frequent in a short time window before the onset of a target event.

3.1. Alarm Labeling

Let $(X_{ft})_{t=1}^{T} \in \{0,1\}$, $f \in \mathscr{F}$ be the alarms signals, or, more generally, binary features, and $(y_t)_{t=1}^{T} \in \{0,1\}$ is the target event signal. For any binary signal $(b_t)_{t=1}^{T} \in \{0,1\}$ we define firing times as

$$I_b = \{t = 1,\ldots,T \,:\, b_t = 1\} \subseteq \{1,\ldots,T\}\,.$$

If there are J events, then we define $I_b = (t^j)_{j=1}^{J}$, where $t^j < t^{j+1}$.

Let $w \geq 1$ be an integer size of the "predictive window", $h \geq 0$ be an integer size of the "predictive horizon", and $m \geq 0$ be an integer delay due to a "maintenance action" brought about by the fact that an event took place. The integers w and h determine the width and offset, respectively, of a window, within which an alarm is considered to have signaled this upcoming event correctly. In turn integers h and m determine the "maintenance period" during which alarms are ignored because the maintenance action is still in effect. The integer h determines for how many moments prior to the onset of an event an alarm is considered anticipatory: if $h = 0$, then any alarm prior to the onset of an event, even if it fired immediately before, is anticipatory.

Since consecutive streaks of 1 target event signals can be considered to pertain to the same underlying breakdown or cause, the occurrences which are too close to one another are clumped together and treated as one single "prolonged" event.

In the following, unless stated otherwise, intervals of the form $[a,b)$ are understood as left-closed right-open integer intervals, i.e. for $a < b$

$$[a,b) := \{n \in \mathbb{Z} : a \leq n < b\},$$

and $[a,b) = \emptyset$ for $b \leq a$.

The parameters h, w and m determine the maximal time separation between consecutive occurrences for them to be clumped: if $t^k - t^j < h + m + w$, then events j and k are considered as the same event. Thus "clumping" transforms the firing times of target event series I_y into a collection of non-overlapping intervals $\left[t_0^{j_k}, t_1^{j_k}\right)$ with sufficient separation between them: $t_0^{j_{k+1}} - t_1^{j_k} \geq h + m + w$ for all $k = 1,\ldots,K$, where K is the number of "distinct" clumped events. Note that the intervals are left-closed right-open: $t_0^{j_k}$—the first moment of the current onset of an event, $t_1^{j_k}$—the first moment just after the event has stopped.

The "predictiveness" of an alarm $f \in \mathscr{F}$ with respect to target events y is determined by counting how often alarms $(X_{ft})_{t=1}^{T}$ fires too early (a false alarm) or timely (a true alarm) with respect to y. Alarms during maintenance action effects are ignored. The exact labeling of alarm signals is described below.

Let s be the time of firing of alarm f, i.e. $X_{fs} = 1$. If no event $y_t = 1$ takes place during $t = 1,\ldots,T$, $K = 0$, then the alarm is **too early** if $s \in [0, T - w)$, and **irrelevant**, if $s \in [T - w, T]$. If an event does occur, then let $[t_{K0}, t_{K1})$ be the interval during which the last Kth events takes place. This case is similar to the "no-event" case, in that the alarm is **too early**, if $s \in [t_{K1} + m, T - w)$, and **irrelevant**, if $s \in [T - w, T]$. In such cases it is assumed that the $K + 1$th event would take place in the "indefinite future", i.e. during the interval $[t_{K+1,0}, t_{K+1,1}) = [T + h + 1, +\infty)$.

Finally consider any $k < K$, and suppose that kth event takes place during the interval $[t_{k0}, t_{k1})$ (see Fig. 2). If the alarm at s is signaled before the event ends, $s < t_1$, then this alarm is pertinent to the upcoming event k and, depending on where s lies, the alarm firing is considered

1. **Timely** if $s \in [t_{k0} - h - w, t_{k0} - h)$;

2. **Irrelevant due to maintenance** when $s \in [t_{k-1,1}, t_{k-1,1} + m)$;

3. **Irrelevant due to event onset if** s lies within either $[t_{k0} - h, t_{k0})$, or $[t_{k0}, t_{k1})$;

4. To be **false** or **too early** whenever $s \in [t_{k-1,1} + m, t_{k0} - h - w)$.

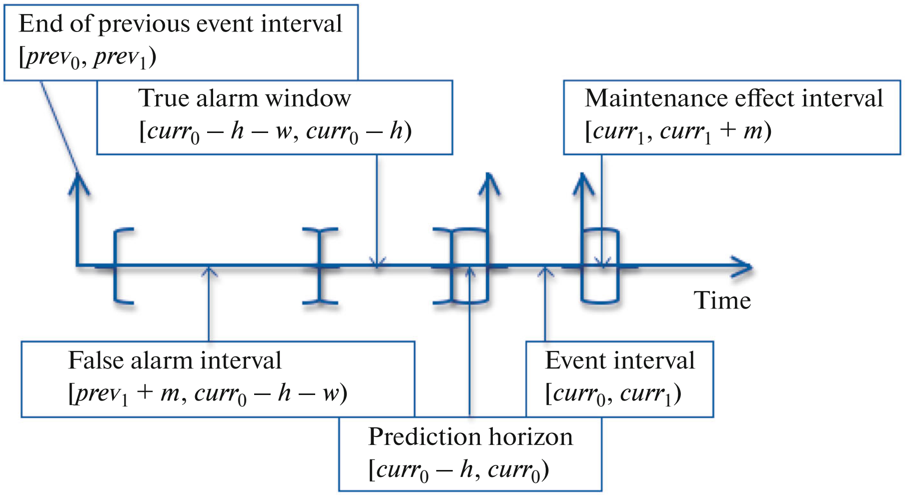


Fig. 2. Diagram of order and geometry of false positive and true positive periods, the predictive horizon, and event and "maintenance effect" intervals.

Here it is assumed that for $k = 1$ the previous event took place in the "infinite past", i.e. during $[t_{k-1,0}, t_{k-1,1}) = (-\infty, -m)$.

Let K^+ be the number of true alarm periods (coincides with K), K^- be the number of false alarm periods (either K, or $K+1$). The number of unique true alarm hits is computed by $U_f^+ = \left|\{k = 1,\dots,K^+ : \exists t \in [t_{k0} - h - w, t_{k0} - h) \text{ with } x_{ft} = 1\}\right|$, and total amount of true alarm hits is given by $S_f^+ = \left|\{t = 1,\dots,T : x_{ft} = 1 \text{ and } \exists k = 1,\dots,K^+ \text{ with } t \in [t_{k0} - h - w, t_{k0} - h)\}\right|$. The counter of unique false alarms is $U_f^- = \left|\{k = 1,\dots,K^- : \exists t \text{ with } x_{ft} = 1 \text{ and } t \text{ is too early}\}\right|$, and total true alarm hits by $S_f^- = \left|\{t = 1,\dots,T : x_{ft} = 1 \text{ and } t \text{ is too early}\}\right|$.

3.2. *Matching Algorithm*

If $(X_{ft})_{t=1}^{T} \in \{0,1\}$ is the feature binary signals (alarms), $f \in \mathcal{F}$ and $(y_t)_{t=1}^{T} \in \{0,1\}$ is the target event signal, then the algorithm for matching events with alarms is as follows:

1. Produce distinct event intervals for $(y_t)_{t=1}^{T}$ of the form $[t_{k0}, t_{k1})$, for $k = 1,\dots,K$ (K might be zero);

2. Let K^+ be the number of true alarm periods (coincides with K), K^- be the number of false alarm periods (either K, or $K+1$);

3. For each $f \in \mathcal{F}$ compute the base metrics:

(a) Count true alarms total and unique hits S_f^+ and U_f^+, respectively;

(b) Count S_f^- and U_f^-, total and unique hits of false alarms, respectively;

(c) Compute the following derived metrics:

- the false and true alarm probability estimates

$$p_f^{\varepsilon} = \frac{S_f^{\varepsilon}}{K^{\varepsilon}}$$

for $\varepsilon \in \{+,-\}$;

- the false alarm ratio: $\mathrm{fa}/\mathrm{cf}_f = \dfrac{S_f^-}{U_f^+}$;

- p-value PV_f of the exact test for statistical equality between p_f^+ and p_f^-, when $p_f^+ > p_f^-$;

4. Apply at least one of the following strategies to get $\mathcal{F}_{\text{predictive}} \subseteq \mathcal{F}$:

rule based on t-test: keep $f \in \mathcal{F}_{\text{predictive}}$, if $U_f^+ > 1$ and $\mathrm{PV}_f \geq \alpha$, where α is some fixed significance level $[0,1]$; usually $\alpha = 0.05$;

hard rule: keep $f \in \mathcal{F}_{\text{predictive}}$, when $U_f^+ \geq \theta$ and $U_f^- = 0$; here θ is a hard threshold, which controls the support of the hypothesis that f implies the occurrence of the event y;

soft rule: keep $f \in \mathcal{F}_{\text{predictive}}$, if $\mathrm{fa}/\mathrm{cf}_f \leq \theta$, where θ is some "soft" threshold, controlling the ratio of false to true alarms.

It is possible that no feature in $\mathcal{F}$ is sufficiently predictive according to specified filtering strategies and their parameters. Of course, in this case

Table 1. ACMS-reports information

Report	Columns	Records	Aircrafts (out of 31)	Phases
R1	322	58917	31	8, 9
R2	406	62919	31	8, 9
R3	318	33831	31	7.1
R4	339	33467	31	7.1
R5	1452	22	12	1, 2, 8
R6	Differ	80	17	1, 2, 8, 11
R7	315	122	4	8
R8	Differ	5701	25	1.1, 2
R9	Differ	30823	31	3
R10	Differ	355	28	1, 1.1, 8
R12	Differ	27432	31	11
R13	Differ	2574	31	9
R14	83	34490	31	11
R15	Differ	359	30	9
R17	225	16964	31	1, 2
R21	Differ	24016	31	8
R23	Differ	6426	31	12
R25	Differ	33533	31	11
R26	Differ	264	19	2, 8, 8.1
R30	831	33557	31	8.1

$\mathcal{F}_{\text{predictive}} = \emptyset$. Furthermore it is quite likely, that different target events y^1 and y^2 have different sets of predictive features $\mathcal{F}_{\text{predictive}}\left(y^1\right) \neq \mathcal{F}_{\text{predictive}}\left(y^2\right)$.

3.3. Predictive Metrics

The key metrics, which characterize efficiency of any early warning system with fixed prediction horizon and window width, are:

- number of false alarms (false positives), or false alarm rate;
- ratio of covered events (true positives) to total number of event occurrences correctly detected.

The precision of the early warning system is inversely proportional to the false alarm rate (or number). The more precise a system is, the more confidence its users may put to the fact that alarms indeed anticipate events, rather than just being a spurious signal.

The sensitivity of a system measures the share of events that a system can catch, and it is directly proportional to the number of covered events (the coverage rate).

These metrics are at the core of event matching algorithm, and are directly used for feature selection. For a particular target event and alarm signals $(y_t)_{t=1}^T \in \{0,1\}$ and $(x_t)_{t=1}^T \in \{0,1\}$, respectively, these metrics are calculated with the following steps:

1. Compute the false-, true- and event intervals for y_t with parameters $h \geq 0$, $w \geq 1$ and $m \geq 0$, as in Section 3.1;

2. For the alarm signal x_t get K^+, U^+, and S^- counters over this interval structure;

3. Get the metrics:

$$\textbf{false alarm rate } \mathrm{fa}_x = \frac{S_f^-}{K^+} \in [0, +\infty);$$

$$\textbf{coverage rate } \mathrm{cf}_x = \frac{U_f^+}{K^+} \in [0, 1].$$

Note, that false alarms are counted as total number of hits of false positive intervals, whereas the true alarms are computed as unique hits only.

4. RAW DATA SUMMARY

Passenger aircraft telemetry data is presented in the Central Management System (CMS) reports and in the Aircraft Condition Monitoring System (ACMS) reports. CMS reports contain binary (yes/no) data about certain types of events (for example, failures of certain types) at certain points in time. CMS reports contain information about one or more snapshots of the values of a set of specified parameters, as well as messages with descriptions of certain failures collected from different subsystems of the aircraft.

CMS event occurrence data and raw CMS report data are stored in plain text files with a tabular structure. This paper presents the results obtained from historical data that covered the period from April 2011 to November 2013. The data is available for 31 aircrafts.

Each record in CMS files contains at least time of occurrence, aircraft number and flight phase. Extra fields are not used in dataset preparation and further analysis.

Each ACMS report has a type (a number ranging from 1 to 30), contains time of acquisition, flight phase, values of measurements of parameters and extra data such as codes of departure and arrival points, type of aircraft, etc. ACMS reports conform to Engine Alliance specification and the following types are available: 1 through 15, 17, 21, 23, 25, 26, and 30.

Sets of measured parameters contained in a report vary with its type. Time range covered by available reports also depends on their type, e.g., all available reports of type 1 may cover different time range than the ones of type 2. Table 1 provides a summary of available ACMS reports, including information on a number of aircrafts out of 31 aircrafts, for which ACMS reports of the corresponding type are available.

For analysis of some types of failures in A380 we used ACMS reports available for all aircrafts and hav-

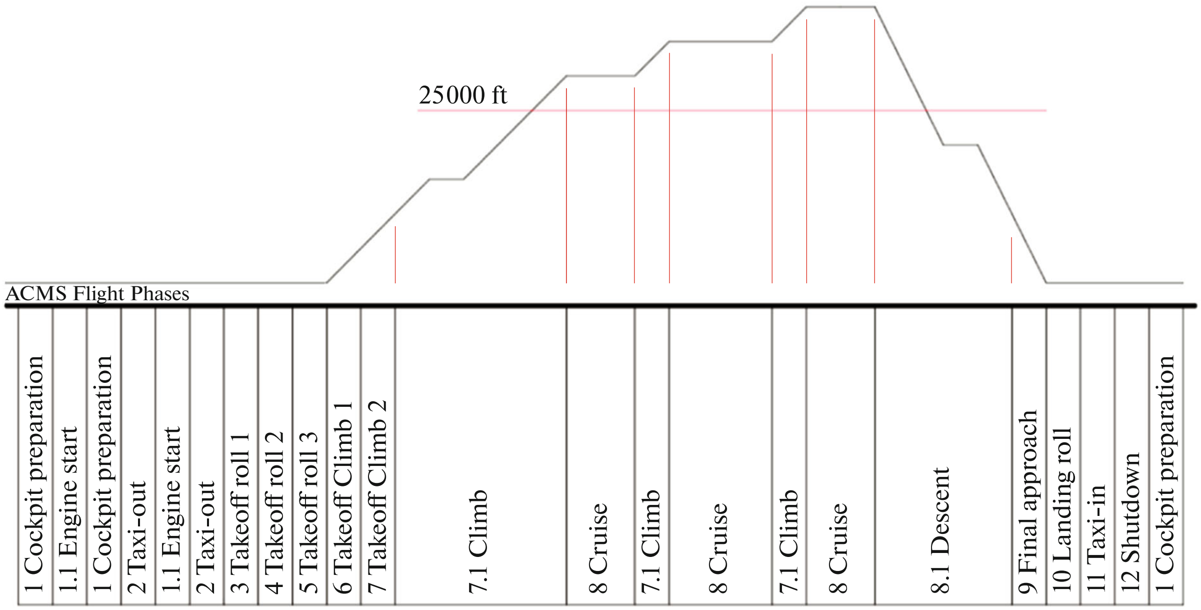


Fig. 3. Flight phases.

ing the same set of columns. According to Table 1, these reports are 1, 2, 3, and 4.

5. DATA PREPARATION

A “feature”, or a “measurement” is a parameter measurement contained in an ACMS report generated at a particular phase. Parameters from different reports, but with the same name are treated as measurements of the same phenomenon. For example:

- parameter MN_D10 (Mach Number) from R1 measured in phase 8 is one feature;
- MN_D10 from R2 measured in phase 8 is the same feature, although measured for a report of another type;
- MN_D11 from R1 measured in phase 8 is another feature, since, as its name indicates, it draws on a different source for its value;
- MN_D10 from R1 measured in phase 9 is also another feature, since flight phase of acquisition is different (see the list of phases in Fig. 3).

For each aircraft, matrices X and Y with following properties were formed:

- Matrix X is a *flight-feature matrix*, i.e., X_{ij} is a value of jth feature, measured during ith flight;

—Matrix X may contain missing values due to one of the following reasons:

- A missing value in initial ACMS data, a value labelled as erroneous, as determined by the prefix in the raw report dump;
- Not every report is generated during a flight;

—Matrix Y is a *flight-failure* binary matrix, i.e. $Y_{ij} = 1$ if a CMS message with the jth failure code occurs in the ith flight, otherwise $Y_{ij} = 0$.

Each row X and Y represents a single flight, whereas a column of X represents a single ACMS feature and a column of Y represents an indicator of a CMS failure code appearance. Rows of both matrices are synchronized: $X_{i\cdot}$ and $Y_{i\cdot}$ correspond to the same flight, and ordered chronologically.

The raw ACMS and CMS data neither directly identifies a flight during which a reports was generated, nor provides information on the time and duration of a flight. This information is crucial to proper synchronization of the dataset, for otherwise these artifacts may be introduced into it:

- either grouping of report instances that actually corresponding to different flight;
- reports from different flights might get pooled into same flight;
- flight, during which no reports were produced at all, which means that some consecutive rows in the dataset could represent non-consecutive flights;
- mis-assignment of CMS fault event records, would result in incorrect results.

5.1. Details of Data Preparation

These problems, indicated above, were resolved by pooling the row data from all reports and ordering it with respect to the timestamp of report acquisition and its phase. This solution rests on the following assumptions:

—During a flight, an aircraft passes flight phases in their natural order: consecutively from 1 to 7, then 7.1, 8, 8.1 in arbitrary order, and at last from 9 to 12 in succession;

—During each flight, at least one ACMS report is generated during phases 1–7, at least one during phases $\{7.1, 8, 8.1\}$, and at least one during phases 9–12;

Algorithm of construction of matrices X and Y for each aircraft:

1. Recognize flights and assign reports to them

(a) Sort all ACMS reports available for the aircraft by their acquisition timestamp;

(b) Find intersection of time ranges covered by ACMS reports R1–R4, and drop all reports acquired outside of this intersection;

(c) Denote remaining reports as $r_1,\dots r_n$ (in chronological order). Denote value of parameter param of report r_i as $r_i[\text{param}]$.

(d) Find all $i > 1$, such that r_i has earlier flight phase than r_{i-1}, excluding the cases when phases of r_i, and r_{i-1} from the subject domain point of view should not contain information about those types of failures which we would like to predict.

(e) Denote i from the previous step as $s_2,\dots,s_m$, also denote $s_1 = 1$, $s_{m+1} = n + 1$.

(f) Under the assumption, reports acquired in flight j, are $r_{s_j}, r_{s_j+1},\dots,r_{s_{j+1}-1}$, and m is total number of flights in data. Assign reports to flights using this rule.

(g) Estimate time of start and time of end for each flight as $t_j^{\text{start}} = r_{s_j}[\text{TIMESTAMP INDEX}]$, $t_j^{\text{end}} = r_{s_{j+1}-1}[\text{TIMESTAMP INDEX}]$. In this manner flight j starts no later than t_j^{start}, and ends no sooner than t_j^{end}.

6. PREDICTION OF CERTAIN TYPES OF FAILURES FOR A380

The data used in this case is ACMS reports 1 through 4, which provide value in the areas of Indication Faults, Scheduling faults, Gas Path Monitoring, and Engine Performance Monitoring.

These report provide data which is a major element in Engine Removal Scheduling, with potential for extending engine Time on Wing, and for Engine Workscope planning.

Given below are basic descriptions of reports 1–4:

Report $\langle 001\rangle$:

The "Engine Cruise Report" provides data on such Indication Faults are sensor errors (offset, drift, noise) and sensor processing errors (signal conditioning circuitry, wiring). Examples of Scheduling faults are misrigging, broken or sticky linkages for Bleeds and Vanes, Bleed Leaks, TCC faults (valves), and Air Oil Cooler faults (valves). Detection of Scheduling faults is often aided by information from "Take Off" and/or "Climb Reports". Gas Path Monitoring provides data for trending of Module performance degradation.

Report $\langle 002\rangle$:

The "Aircraft Cruise Performance Report" captures data required for aircraft performance monitoring in "cruise" phase. It extends the "Engine Cruise Report" by recording sufficient parameters for aircraft, engine and internal component performance.

Report $\langle 003\rangle$ and $\langle 004\rangle$:

"Engine Climb Report" and "Engine Takeoff Report", respectively, provide data over Indication and Scheduling faults, which are detailed in the Engine Stable Cruise Report; Engine Performance Monitoring measurements include Maximum Continuous EGT Margin, Climb Reduced Thrust Derate during Climb phase, and Take Off EGT Margin and Take Off Reduced Thrust Derate during the Take off phase. These reports complement Report $\langle 001\rangle$ as subtle Scheduling faults require observation at two or more operating points.

The goal was to find failures related to engines' operation (ATA codes 60-80; ATA codes—Air Transport Association numbering system), that could be foreseen by cleverly constructed early warning signals. Physically irrelevant positional, angular and control parameters were excluded. The resulting early warning alarms rely mostly on engine related data, but sometimes include measurements related to the aircraft as a whole.

The dataset was prepared with the technique, described in Section 5.

6.1. Feature Partitioning

The original format of reports 1–4 as presented in specifications for Engine Alliance engines is the following: a header block followed by the data block. Each block is a set of groups consisting of a header line and up to four lines of data in a column. Each line is numbered according to the following pattern: **block name‖within-block line number**, where the "‖" symbol stands for concatenation. Block name has format '$[A - Z][0 - 9]$', i.e. a letter and a decimal digit. Line number is a one-digit number. The block header defines the measurement reported in its column, whereas the number of lines varies, depending on the source of the data.

For these reports, in a parameter column with 4 reported rows, each measurement has 1 of 4 ACMS channels as its source. Each channel can be identified with a particular engine mount point on the aircraft, which is assume to be hardwired through the fuselage of the airplane and never rewired throughout the whole lifespan of the aircraft, until it is decommissioned entirely.

In the raw data of these reports parameter columns are collapsed into a single row with parameter (measurement) identifiers having the following format: **parameter name** ‖ "_" ‖ **block name** ‖ **within-block line number**.

It is important to note that parameter names are alphanumeric sequences separated from the column identifiers by an underscore. For these reports a rule which determines whether a given parameter is related to the aircraft or is a measurement of some engine-related characteristic is very simple:

—**engine-related,** if four rows within a block are reported;

—**aircraft-related,** otherwise.

For example "EGTRM_E10, EGTRM_E11, EGTRM_E12, and EGTRM_E13" are considered engine-related, whereas "CAS_A10, CAS_A11, and CAS_A12", or "GWT_B10" are different aircraft-related measurements. Thus engine-related features are partitioned into groups of 4 measurements—one per each engine (ACMS channel).

The measurement groups used in the analysis are formed from parameter and block name, while the elements of a group are values in the the rows of the corresponding column. It is important to note that data is grouped only within each report and never between the reports. This produces the following groups: "R" ‖ **report number** ‖ "::" **parameter name** ‖ "_" ‖ **block name** ‖ "x".

For instance, the measurements mentioned before get this name: "R1::EGTRM _ E1x" . Aircraft-related data is not grouped at all and each measurement is used as a single feature.

6.2. Anomaly Extraction

Anomalies within each engine-related parameter group are extracted by a detector based on the Principal Component analysis [12], which consists of the following steps:

1. The training data is used to learn a 1-dimensional linear manifold, which explains the most of joint variance of elements within the group; see Fig. 1 with $\mathcal{M}$, being a linear 1 subspace, constructed from the principal directions of the estimated covariance matrix of the measurements in a group;

2. The test data point d is projected onto the manifold to a point $\hat{d}$, with respect to which the linear reconstruction error $Q = \|d - \hat{d}\|^2$ is inferred;

3. The abnormality score is computed by taking the logarithm of Q;

4. The threshold, which categorizes the scores into "normal" and "anomalous", is set to the $(1-\alpha)$-quantile of the empirical distribution of the score of the train dataset, where $\alpha \in \{95\%, 99\%\}$.

In general, we can use any other anomaly detection method instead of this particular anomaly detection approach based on PCA. In the considered case it is the detector based on PCA, which provided the best possible failure prediction accuracy.

Anomaly extraction process for aircraft-related measurements is simpler: the threshold for abnormality is set to $(1-\alpha)$-quantile of the empirical distribution of the measurement in the train dataset, with $\alpha \in \{95\%, 99\%\}$.

6.3. Generation of Alarms Signal

For each target fault event signal $(y_t)_{t=1}^T$ (occurrence of an CMS fault code) construct a subset of pairs of individual anomalies $\mathcal{P} \subseteq \{\{a_1, a_2\} : a_1, a_2 \in \mathcal{F}\}$, such that $A \in \mathcal{P}$ if and only if y_t is considered to be sufficiently predictable by the signal $A_t = \wedge_{a \in A} a_t$. Intuitively, if anomalies happen exactly at during the same flight, see Section 2.3, and significantly more frequently before a fault event, then they are in $\mathcal{P}$.

Alarms signals for event y are synthesized using: $S_t = \vee_{A_t \in \mathcal{P}} A_t$, i.e. an early warning signal is fired if at least one joint anomaly was detected. The advantage of this "at least one" rule, is that the chances that pooling predictive anomalies increases the chances of successfully anticipating an event, whereas a drawback is that such ensemble has elevated number of false alarms, since not always anomalies are perfectly predictive (i.e. occur never but before an event).

6.4. Estimation of Forecast Accuracy

The quality parameters of the early warning system (see Section 3.3) were set to $h = 0$ in case of horizon, $w = 30$ in case of flight window width, and $m = 0$ in case of duration of maintenance effects.

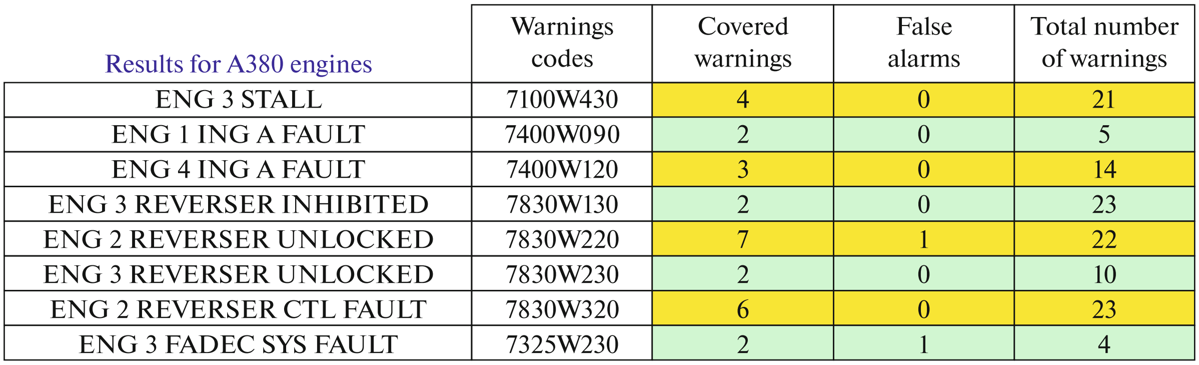
In Fig. 4 we present the results of predicting various types of failures associated with the engine:

—the first column of the table lists the names of different types of engine-related failures;

—the second column of the table lists the ATA codes for these types of failures;

—the next three columns of the table for each type of failure contain information about the number of failures that were correctly predicted, the number of false alarms, and finally the total number of failures of this type, respectively.

Thus, from the results presented in Fig. 4, it follows that using the algorithm proposed in this work, it was possible to obtain the accuracy of the forecast, which provides the detection of ~28% of important failures, and the number of false alarms is less than ~10% (that



Results for A380 engines	Warnings codes	Covered warnings	False alarms	Total number of warnings
ENG 3 STALL	7100W430	4	0	21
ENG 1 ING A FAULT	7400W090	2	0	5
ENG 4 ING A FAULT	7400W120	3	0	14
ENG 3 REVERSER INHIBITED	7830W130	2	0	23
ENG 2 REVERSER UNLOCKED	7830W220	7	1	22
ENG 3 REVERSER UNLOCKED	7830W230	2	0	10
ENG 2 REVERSER CTL FAULT	7830W320	6	0	23
ENG 3 FADEC SYS FAULT	7325W230	2	1	4

Fig. 4. Results of predicting various types of engine-related failures (see description in Section 6.4).

is, on average, nine correctly predicted failures for one false alarm). These accuracy indicators meet the requirements for such models in practice, and therefore the models have been integrated into the customer's system for early warning of failures.

Note that the constructed models of early warnings about failures are based on a combination of several "simple" models. Each of these "simple" models either detects anomalies in the behavior of parameters from the group of parameters associated with the engine, or by a threshold rule detects anomalies in the values of parameters related to the aircraft (see Section 6.2). Thus, in each case of early warning about a failure, it is possible to find out due to change in the values of which parameters the alarm announcement has occurred. This in turn makes these types of models interpretable and helps engineers servicing the technical system to find the causes of failure faster if it occurs.

If we apply standard methods of machine learning for the construction of predictive models based on random forests, gradient boosting over decision trees, neural networks, etc., it is not possible to achieve comparable accuracy indicators, which are presented in Fig. 4. This conclusion applies both when the same input features that were used in the construction of the proposed models of early warnings about failures are used, and when any other available telemetry inputs and combinations thereof are used. Perhaps the reason is that to achieve an effective relationship between such indicators as "rate of false alarms" and "rate of undetected targets" it is necessary to use those models that effectively take into account the structure of the problem. The point is that in practice, before a failure occurs, small changes in the behavior of the system usually begin, caused by emerging defects that can be detected by analysing properties of the data flow coming from the sensors. These defects worsen over time and lead to critical failures, up to complete equipment failure. Accordingly, the approach described in Section 3 allows to build models that simulate this observation, and these models have a fairly "simple" structure—and therefore they are more robust. In turn, predictive models constructed using standard machine learning methods (construction of features based on available telemetry data and subsequent application of methods such as gradient boosting over decision trees) are excessively complex and have an inflated level of false alarms.

7. CONCLUSIONS

In this paper, we consider the problem of constructing a model to predict rare events in a situation where there are hundreds of different indicators of the state of the technical system and it is necessary to predict the failure, examples of which in the available sample of historical data is quite small (the sample is imbalanced). Thanks to the developed approach, it was possible to predict some possible failures of an aircraft. Due to the special structure of the model, we provided a low level of false alarms with a significant proportion of detected failures. Experiments using standard machine learning methods to construct predictive models have not allowed to obtain predictions with comparable accuracy.

E. Burnaev expresses gratitude to I. Nazarov and P. Erofeev for assistance in data processing, to N. Klyuchnikov for technical assistance in the preparation of the text of the article, and to Datadvance LLC for providing data and setting the task under the contract with the external company.

FUNDING

The study was supported in part by a grant Russian Foundation for Basic Research 16-29-09649 ofi_m.

REFERENCES

1. *Maintenance Optimization. Airplane Health Management* (2015).
2. S. Alestra, C. Bordry, C. Brand, E. Burnaev, P. Erofeev, A. Papanov, and C. Silveira-Freixo, "Application of rare event anticipation techniques to aircraft health

management," Adv. Mater. Res. (N.Y.) **1016**, 413–417 (2014).

3. S. Alestra, E. Burnaev, et al., "Rare event anticipation and degradation trending for aircraft predictive maintenance," in *Proc. Joint WCCM-ECCM-ECFD Congress, Barcelona, July 20–25, 2014* (Int. Center Numer. Methods Eng., 2014), pp. 1–12.
4. A. Artemov and E. Burnaev, "Ensembles of detectors for online detection of transient changes," in *Proc. SPIE* **9875**, 9875–9875-5 (2015).
5. A. Artemov and E. Burnaev, "Detecting performance degradation of software-intensive systems in the presence of trends and long-range dependence," in *Proc. IEEE 16th Int. Conf. on Data Mining Workshops (ICDMW), Barcelona, Spain, 2016* (IEEE, New York, 2016), pp. 29–36.
6. A. Artemov and E. Burnaev, "Optimal estimation of a signal perturbed by a fractional brownian noise," Theor. Probab. Appl. **60**, 126–134 (2016).
7. A. Artemov, E. Burnaev, and A. Lokot, "Nonparametric decomposition of quasi-periodic time series for change-point detection," Proc. SPIE **9875**, 9875–9875 (2015).
8. M. Baptista, I. P. de Medeiros, J. P. Malere, C. Nascimento, H. Prendinger, and E. M. P. Henriques, "Comparative case study of life usage and data-driven prognostics techniques using aircraft fault messages," Comput. in Industry, **86**, 1–14 (2017).
9. M. Baptista, I. P. de Medeiros, J. P. Malere, H. Prendinger, C. L. Jr. Nascimento, and E. Henriques, "Improved time-based maintenance in aeronautics with regressive support vector machines," in *Proc. Ann. Conf. Prognostics & Health Management Soc.*, Denver, Colorado, Oct. 3–6, *2016* (*Prognostics & Health Management Soc,* 2016), pp. 1–10.
10. M. Baptista, Sh. Sankararaman, I. P. de Medeiros, C. Nascimento, H. Prendinger, and E. M. P. Henriques, "Forecasting fault events for predictive maintenance using data-driven techniques and arma modeling," Comput. & Indust. Eng. **115**, 41–53 (2018).
11. M. Belyaev, E. Burnaev, E. Kapushev, M. Panov, P. Prikhodko, D. Vetrov, and D. Yarotsky, "Gtapprox: Surrogate modeling for industrial design," Adv. Eng. Softw. **102**, 29–39 (2016).
12. E. Burnaev and S. Chernova, "On an iterative algorithm for calculating weighted principal components," J. Commun. Technol. Electron. **60**, 619–624 (2015).
13. E. Burnaev, P. Erofeev, and A. Papanov, "Influence of resampling on accuracy of imbalanced classification," Proc. SPIE **9875**, 9875–9875-5 (2015).
14. E. Burnaev, P. Erofeev, and D. Smolyakov, "Model selection for anomaly detection," Proc. SPIE **9875**, 9875–9875-6 (2015).
15. E. Burnaev and I. Nazarov, "Conformalized kernel ridge regression," in *Proc. 15th IEEE Int. Conf. on Machine Learning and Applications (ICMLA), Anaheim, California, USA, December 18–20, 2016* (IEEE, New York, 2016), pp. 45–52.
16. E. Burnaev and D. Smolyakov, "One-class svm with privileged information and its application to malware detection," in *Proc. IEEE 16th ICDMW, Barcelona, Spain, Dec. 12–15, 2016* (IEEE. New York, 2016), pp. 273–280.
17. E. Burnaev and V. Vovk, "Efficiency of conformalized ridge regression," in *Proc. 27th Conf. Learning Theory, Barcelona, Spain, June 13–15, 2014* (Proc. Machine Learning Res. (PMLR), 2014), Vol. 35, pp. 605–622.
18. E. V. Burnaev and G. K. Golubev, "On one problem in multichannel signal detection," Problems Inform. Transmis. **53**, 368–380 (2017).
19. "On a method for constructing ensembles of regression models." Autom. Remote Control **74**, 1630–1644 (2013).
20. J. Dai and H. Wang, "Evolution of aircraft maintenance and logistics based on prognostic and health management technology," in *Lecture Notes in Electrical Engineering: Proc. First Symp. on Aviation Maintenance and Management-Volume II* **297**, 665–672 (2014).
21. E. Fumeo, L. Oneto, and D. Anguita, "Condition based maintenance in railway transportation systems based on big data streaming analysis," Proc. Comput. Sci. **53**, 437–446 (2015). (INNS Conf. on Big Data 2015 Program San Francisco, CA, USA, 8–10 August 2015).
22. F. P. García, D. J. Pedregal, and C. Roberts, "Time series methods applied to failure prediction and detection," Reliability Engin. & System Safety **95**, 698–703 (2010).
23. V. Ishimtsev, A. Bernstein, E. Burnaev, and I. Nazarov, "Conformal k-nn anomaly detector for univariate data streams," in *Proc. 6th Conformal and Probabilistic Prediction and Applications (COPA 2017) Workshop, June 13–16, 2017* (Proc. Machine Learning Res. (PMLR), 2017), Vol. 60, pp. 213–227.
24. S. Ivanov and E. Burnaev, "Anonymous walk embeddings," in *Proc. 35th Int. Conf. on Machine Learning (ICML), Stockholm, Sweden, 2018* (Proc. Machine Learning Res. (PMLR) **80**, 2186–2195 (2018)).
25. S. Ivanov, N. Durasov, and E. Burnaev, "Learning node embeddings for influence set completion," in *Proc. IEEE ICDMW, 2018* (IEEE, New York, 2018), pp. 1034–1037.
26. A. Korotin, V. V'yugin, and E. Burnaev, "Long-term online smoothing prediction using expert advice," ArXiv e-prints, abs/1711.03194 (2017).
27. A. Korotin, V. V'yugin, and E. Burnaev, "Aggregating strategies for long-term forecasting," in *Proc. 7th Workshop on Conformal and Probabilistic Prediction and Applications (PMLR), 2018*, Vol. 91, pp. 63–82.
28. A. Korotin, V. V'yugin, and E. Burnaev, "Adaptive hedging under delayed feedback," ArXiv e-prints, abs/1902.10433 (2019).
29. A. Kuleshov, A. Bernstein, and E. Burnaev, "Conformal prediction in manifold learning," in *7th Symp. on Conformal and Probabilistic Prediction and Applications, (COPA 2018), Maastricht, The Netherlands, June 11–13, 2018* (Proc. Machine Learning Res. (PMLR), 2018), Vol. 91, pp. 234–253.
30. A. Kuleshov, A. Bernstein, and E. Burnaev, "Kernel regression on manifold valued data," in *Proc. IEEE 5th Conf. on Data Science and Advanced Analytics (DSAA), Turin, Italy, 2018*, (IEEE, New York, 2018), pp. 120–129.
31. A. Kuleshov, A. Bernstein, and E. Burnaev, "Manifold learning regression with non-stationary kernels," in *Ar-*

tificial Neural Networks in Pattern Recognition, (Workshop, ANNPR 2018, Siena, Italy, Sept. 19–21, 2018) (Springer-Verlag, 2018), pp. 152–164.

32. A. Kuleshov, A. Bernstein, E. Burnaev, and Y. Yanovich, "Machine learning in appearancebased robot self-localization," in *Proc. 16th IEEE ICMLA, Cancun, Mexico, Dec. 18–21, 2017* (IEEE, New York, 2017), pp. 106–112.

33. Liu Datong, Peng Yu, and Peng Xiyuan, "Fault prediction based on time series with online combined kernel svr methods," in *Proc. IEEE Instrumentation and Measurement Technology Conf.,* Singapore, *May 5–7,2009* (IEEE, New York, 2009), pp. 1163–1166.

34. R. Rivera, I. Nazarov, and E. Burnaev, "Towards forecast techniques for business analysts of large commercial data sets using matrix factorization methods," J. Phys.: Conf. Ser. **1117**, 012010 (2018).

35. R. Rivera-Castro, I. Nazarov, Yu. Xiang, A. Pletneev, I. Maksimov, and E. Burnaev, "Demand forecasting techniques for build-to-order lean manufacturing supply chains," ArXiv e-prints abs/1905.07902, 2019.

36. A. Safin and E. Burnaev, "Conformal kernel expected similarity for anomaly detection in time-series data," Adv. Systems Sci. Appl. **17** (3), 22–33 (2017).

37. B. Saha, A. Mandal, S. B. Tripathy, and D. Mukherjee, "Complex networks, communities and clustering: A survey," Computing Research Repository (CoRR) abs/1503.06277 (2015).

38. T. Shen, F. Wan, W. Cui, and B. Son, "Application of prognostic and health management technology on aircraft fuel system," in *Proc. IEEE Prognostics and System Health Management Conf., Macao, Jan. 12–14, 2010* (IEEE, 2010), pp. 1–7.

39. D. Smolyakov, A. Korotin, P. Erofeev, A. Papanov, and E. Burnaev, "Meta-learning for resampling recommendation systems," Proc. SPIE (11th ICMV) **11041**, 110411S, 2019.

40. D. Smolyakov, N. Sviridenko, E. Burikov, and E. Burnaev, "Anomaly Pattern Recognition with Privileged Information for Sensor Fault Detection," in *Articial Neural Networks in Pattern Recognition,* Ed. by L. Pancioni, F. Schwenker, and E. Trentin, (Springer-Verlag, Int., Cham, 2018), pp. 320–332.

41. D. Smolyakov, N. Sviridenko, V. Ishimtsev, E. Burikov, and E. Burnaev, "Learning Ensembles of Anomaly Detectors on Synthetic Data," arXiv e-prints, abs/1905.07892 (2019).

42. S. Su, W. Zhang, and S. Zhao, "Online fault prediction for nonlinear system based on sliding arma combined with online ls-svr," in *Proc. 33rd Chinese Control Conf. (CCC), Nanjing, China, July 28–30, 2014*, (CCC, 2014), pp. 3287–3291.

43. L. Tegtmeier, "Math. and Maintenance," Aviation Week & Space Technol. **174** (39) (2012).

44. Tian Wende, Hu Minggang, Li Chuankun, "Fault prediction based on dynamic model and grey time series model in chemical processes," Chinese J. Chem. Eng. **22**, 643–650 (2014).

45. A. Zaytsev and E. Burnaev, "Large scale variable fidelity surrogate modeling," Ann. Math. Artificial Intellig. **81** (1), 167–186 (2017).