Learning to learn HVAC failures: layering ML experiments in the absence of ground truth ⊙

Carlos E. Budde[1], Duncan Jansen[2], Inka Locht[3], and Mariëlle Stoelinga[2,4]

[1] University of Trento, Trento, Italy
[2] University of Twente, Enschede, the Netherlands
[3] Dutch Railways (NS), Utrecht, the Netherlands
[4] Radboud University, Nijmegen, the Netherlands

Abstract Passenger comfort systems such as Heating, Ventilation, and Air-Conditioning units (HVACs) usually lack the data monitoring quality enjoyed by mission-critical systems in trains. But climate change, in addition to the high ventilation standards enforced by authorities due to the COVID pandemic, have increased the importance of HVACs worldwide. We propose a machine learning (ML) approach to the challenge of failure detection from incomplete data, consisting of two steps: 1. human-annotation bootstrapping, on a fraction of temperature data, to detect ongoing functional loss and build an artificial ground truth (AGT); 2. failure prediction from digital-data, using the AGT to train an ML model based on failure diagnose codes to foretell functional loss. We exercise our approach in trains of Dutch Railways, showing its implementation, ML-predictive capabilities (the ML model for the AGT can detect HVAC malfunctions online), limitations (we could not foretell failures from our digital data), and discussing its application to other assets.

1 Introduction

Heating, Ventilation, and Air-Conditioning units (HVACs) are in charge of air circulation, filtering, heating, and cooling, not only in smart buildings and houses but also in every modern transport system, including trains [5]. At the same time, HVACs traditionally lack the data monitoring quality enjoyed by mission-critical systems, such as air-compressors for break release [8]. Moreover, climate change and higher ventilation standards—enforced by authorities to mitigate airborne diseases—have put HVACs in the foreground [5]. In this scenario where uninterrupted HVAC operation becomes essential, detecting their (even partial) malfunctions rapidly gains on importance. However, the need to automate failure detection is at odds with the reality of sub-optimal data coverage.

As a result, HVAC malfunctions can remain unnoticed for long time periods. This occurs even with sensorially noticeable functions, provided they are not used to perceptible levels. Consider a gas compressor used for cooling, damaged during autumn, whose failure is noticed in summer when it is needed the most.

⊙ This work was partially funded by EU grants 830929 (*H2020-CyberSec4Europe*), and 952647 (*H2020-AssureMOSS*), and NWO grant NWA.1160.18.238 (*PrimaVera*).

Silent failures are not rare: Dutch Railways (NS) handles hundreds of service requests for HVACs in trains, precisely in the first months of the year with temperatures above 25 °C. Periodic maintenance is the traditional mitigation technique for such cases, adjusted for the application domain via field-data studies [4, 11]. More modern approaches use physical, failure, and Machine Learning (ML) models to implement predictive- or condition-based maintenance [2, 3, 10].

Whichever the underlying model, an ideal failure-prevention system would deploy self-diagnosis fed by live data streams, that notify a control unit as soon as threats are detected. Unfortunately this is overall unrealistic because (a) the shelf life of HVACs can surpass 15 years, so many units used today were manufactured even before the Internet of Things began; (b) in any case it is not possible to detect, let alone diagnose every subcomponent and its possible failure causes.

In this scenario, *ML can provide smart solutions that build up to a business case with current assets* [1]. This involves processing large amounts of data, searching for patterns correlated to functional loss. However, *this is hindered by suboptimal data collection, specially if no ground truth is available*, namely when confirmed HVAC failures are not part of the input data.

In this work we propose a simple yet effective supervised ML solution to this pressing challenge, tested in industry and consisting of two steps:

1. human-annotation bootstrapping, to detect functional loss on a fraction of in-coach temperature readings, used to build an ML model capable of creating an artificial ground truth (AGT) for the entire dataset;
2. digital-data failure prediction, that uses the AGT to train an ML model based on failure diagnose codes, and can be used to foretell functional loss.

We define our approach for both steps, including ML features and data preprocessing, applied to generic train coaches with two HVACs each. The outcome of step 1 is a linear and lightweight ML model, pivotal in the construction of the AGT, that extrapolates a minimal human-annotated input. A by-product of such interface between steps 1 and 2 is a software program, that can detect HVAC functional failure online based on temperature readings. Then step 2 uses an independent (digital) dataset—of HVAC components diagnostics—to train another ML model on the AGT. We aim to find patterns in the digital data that can be correlated to later functional loss, thus revealing HVAC silent failures.

1.1 Scientific approach with practical applications

Setting and challenges. *The general goal is to detect and if possible foretell functional failures of HVACs in trains, when there is no ground truth to mark such failures.* By functional failure we mean that a specific function does not perform as needed, e.g. an HVAC should cool down the interior of a coach, but the temperature remains above the one desired. This goal is particularly challenging given our choice of a data-driven approach: detection and prediction must be solely based on a company's data stream[†]. Here, *the historical information*

[†]Besides being COVID-friendly, this is less cost- and time-consuming (although arguably less flexible) than mechanical experiments by technicians and engineers.

available for each HVAC are temperature readings and subcomponent diagnose codes. Both are time series: the former has continuous data; the latter digital data, i.e. a diagnose code in an HVAC (e.g. "high pressure in valve V_{A2}" or "no power in compressor") is either ON or OFF at each point in time.

We define our ML approach in Secs. 2 and 3, demonstrating its applicability in Secs. 2.3 and 3.3 by means of a study on the rolling stock of NS. Thus we showcase our work in a real-world scenario, and stress-tests the ML paradigm in a situation with high-volume but low-information data. Furthermore, the study with NS shows that our approach can be implemented in large-stock companies (with standard data streams) by a team of 2–4 computer-technical personnel.

Concrete objectives:

(O1) Build an ML model, whose input are the historical temperature readings of all HVACs in a train, that estimates the probability of cooling malfunction of each such HVAC currently in operation.

(O2) Build an ML model, whose input are the diagnose codes of an HVAC in operation, that estimates its probability of having a cooling malfunction.

(O3) Extrapolate objective O2 forward in the time series, to estimate the probability of malfunction before the next scheduled periodic maintenance.

We focus on cooling malfunctions as these are more critical for passenger comfort (the temperature in a coach raises with more people) and they are more abundant in our datasets than heating malfunctions. In objective O3 we use the diagnose codes for prediction, and not the temperature readings, because this dataset is richer (an HVAC has several components, each with possibly more than one code) and we expect it to be more likely to show patterns that can be correlated to silent failures. In contrast, patterns occurring in the temperature readings are susceptible to be sensed by train personnel, thus not leading to silent failures.

ML models. The proposed approach involves two supervised ML models, the second built on top of the results of the first, keeping explainability in mind.

- Supervision: Although default solutions in the absence of a ground truth point at unsupervised learning, our objectives involve classification and regression, which require labelled datasets for supervised learning. We label data manually to overcome this for objective O1: simple rules allow non-experts to detect too-high temperatures indicative of cooling failures. To make the process manageable we label small independent data subsets, then extrapolate to the whole time series, and cross-validate the results; full details are in Sec. 2.

- Layering: In contrast, non-experts cannot interpret how diagnose codes from subcomponents indicate functional HVAC failure. But these HVACs are the same units whose malfunctions were labelled for the temperature-based model. Thus we use the result of the process performed for the temperature-based model, as labelled dataset on which to train the diagnose-code-based model.

- Explainability: On top of suitability and performance, we select ML techniques to reach our objectives with the highest amount of transparency. White-box approaches like this are key for the acceptance of data-based solutions, specially in industrial sectors such as railways where processes are traditionally

expert-driven. Objective O1 uses temperature: this is linear data, which partly motivated our choice of a logistic regression (LR) solution. We use LR also for objectives O2 and O3, but this can only estimate the likelihood of malfunction of a set of diagnose codes. It is of additional interest to tell how each individual code contributes to that estimation: we use decision trees (DT) for this purpose.

Main results. On the one hand, • *our LR model for objective O1 can detect HVAC failures in real-time.* On the other hand, • *we could not find evidence of correlation between our HVAC diagnose codes for objectives O2 and O3, and HVAC cooling malfunctions.* As a further result, • *our data features (for all ML models) could be used to study other types of failures and systems.*

2 Learning HVAC failures from temperature readings

To reach objective O1 we use the readings of temperature sensors inside train coaches to detect malfunctioning HVACs. This section introduces our steps for data preprocessing, and for manual data labelling, to bootstrap the entire work. We also define the features used for LR, and the training and testing steps. The section ends showing our empirical studies done in the trains of NS.

2.1 Data preparation

Input. This step works on continuous-valued data, formatted as a time series of HVAC temperatures. So for each time point and for each HVAC, the input indicates the temperature inside of the coach that that HVAC is responsible for.

Preprocessing. Temperature values outside the range $[-20\,°\mathrm{C}, 60\,°\mathrm{C}]$ are considered outliers, and replaced by `NaN` in data imputation. Moreover, data streams may have interruptions that appear as missing values in the time series, which must also be imputed or discarded. If the missing data spans for less than 90 min we use linear interpolation to fill the gap—this was always the case for our NS studies—; else we impute by filling with `NaN`.

Those two steps remove or replace values that will later be used for LR. In addition, the desired temperatures must be computed, since the LR model of this step will be trained on human-annotated data, which must be created by comparing in-coach (actual) temperatures to set (desired) temperatures.

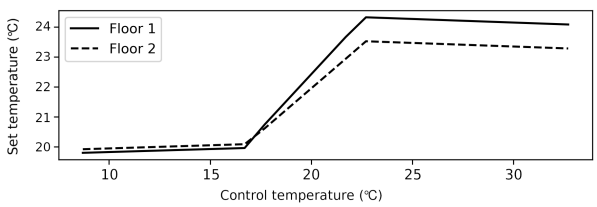


Figure 1: In-coach temperature set by control temp.

More in detail, the automatic operation of an HVAC is typically regulated by a thermostat and a set of rules, that use a control temperature—e.g. from outside the train—to dictate the temperature desired in-coach: see Fig. 1. These so-called *set temperature values* are computed from the (time series of) control temperature values, using functions defined by the temperature-regulation rules.

Data labelling. Supervised learning algorithms, such as LR, require labelled data. One contribution of this work is how to perform these studies on unlabelled data, i.e. when there is no ground truth indicating the moments in which an HVAC is malfunctioning. For that, we note that cooling failures can be spotted by (non-expert) human inspection, by comparing the desired and achieved temperatures inside of the train coaches. Thus we propose a bootstrapping process based on a manual labelling of 1–5 % of the available data as follows.

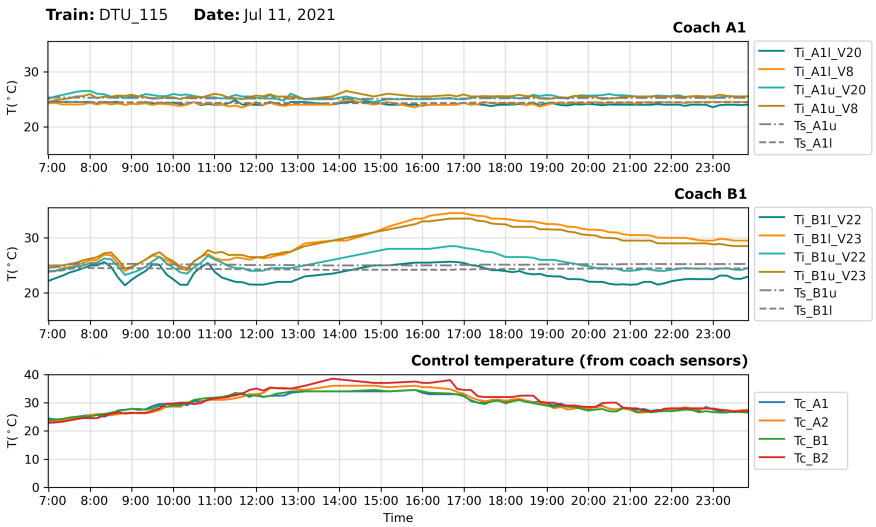


Figure 2: Temperature plots used in the manual-labelling bootstrap process

Humans interpret images better and faster than numeric values, so the temperature time series should be plotted as in Fig. 2. In particular our plots show two values per HVAC because each coach has two decks, and an HVAC controls the temperature of both decks in one side of a coach. *For instance in coach A1, HVAC V20 controls the temperature of the lower and upper deck of the back side.*

These plots of internal temperatures must be time-aligned to the control temperatures, so it is visually straightforward to match high internal values with control values. Observing such high values suggests a cooling failure in the HVAC. *For instance in coach B1, the HVAC V23 shows such positive deviation.*

However, temperature deviations might also be explainable as data anomalies. If false positives are undesired (bringing trains to maintenance is costly) a conservative approach is suggested, where labels are applied iff other HVACs show correct performance during the same time period than the offending unit. *For instance in Fig. 2, this is the case for V23 in coach B1 after 15:00. But from 7:00 to 12:00 both V23 and V22 exhibit oscillatory values that are less easy to interpret as a malfunction, and thus we omit labelling that time period.*

Also periods of HVAC correct operation must be labelled, so the model learns to tell them apart from potential silent failures. *This is the case of both HVACs in coach A1, that keep the temperature at their set values despite the high control*

temperatures. Thus in our case, a **true positive** is when the ML model tells an HVAC is failing, and this coincides with a human hot label. Instead, a **true negative** is when the ML model tells an HVAC is operating correctly, and this coincides with a human healthy label.

The result of such process are human-generated entries as in Table 1, which indicate the time periods of functional correct- and incorrect-operation of an HVAC in a train. Note that we label HVAC V22 in coach B1 as healthy in the period when V23 is labelled as hot. This is due to the low temperatures observed for V22 in that time period, indicating that it was over-cooling its side of coach B1, in an attempt to compensate for the high temperatures on the other side.

train	coach	HVAC	date	period	symptom
DTU_115	A1	V20	11.07.2021	07:00-23:50	healthy
DTU_115	A1	V8	11.07.2021	07:00-23:50	healthy
DTU_115	B1	V22	11.07.2021	15:00-23:50	healthy
DTU_115	B1	V23	11.07.2021	15:00-23:50	hot

Table 1: Manual labels for Fig. 2

Practicality vs. correctness. Human labels can be biased, specially when coming from images susceptible to interpretation. Our guidelines on when to label an HVAC as (un-)healthy reduce this bias, but cannot suppress it. Moreover, plots inspection is a time-consuming process that we suggest to perform on less than 5 % of the data. To quantify the resulting subjectivity, $N > 1$ persons should label independent data sets following the same rules. Then active learning can be used to compute the inter-rater reliability: create training data from the sets labelled by e.g. $\lfloor \frac{N}{2} \rfloor$ persons, and then validate the data of the rest [9]. This allows using metrics such as the area under the receiver operating characteristics curve, to validate whether the different and independent manual labellings are consistent. A concrete example of such procedure is shown in Sec. 2.3.

2.2 Machine learning experiment

Our feature engineering process defines four data features to perform LR for objective O1. If data volume is too large because the frequency of the time series is high, e.g. a data point per second, the averages of a rolling window can be used. This involves defining the window size and the step used to discretise the time series. We do this to define four features over a rolling window of N steps.

Set Point (SP). This feature compares the coach temperature to the set temperature. More specifically, $\mathrm{SP}_{c,v,d}$ is the average of the differences between the in-coach temperature on deck d of coach c in the side corresponding to HVAC v $\left(TI_i^{c,d,v}\right)$, and the desired (set) temperature on that deck of the coach $\left(TS_i^{c,d}\right)$, for each time step i in a rolling window of N steps. We use c to indicate both a coach and its train. We also define a feature $\overline{\mathrm{SP}}_{c,d,v}$ that compares $TS_i^{c,d}$ to the temperature kept by the HVAC $\overline{v}$ that is on the opposite side of v in coach c (this feature is omitted for coaches with a single HVAC):

$$\mathrm{SP}_{c,d,v} = \frac{1}{N}\sum_{i=1}^{N} TI_i^{c,d,v} - TS_i^{c,d} \qquad \overline{\mathrm{SP}}_{c,d,v} = \frac{1}{N}\sum_{i=1}^{N} TI_i^{c,d,\overline{v}} - TS_i^{c,d}.$$

The sign of SP is informative: for cooling malfunctions only positive values are relevant. The magnitude of the difference should be positively correlated to the (cooling) malfunction probability. Similarly, the magnitude and sign of $\overline{\mathrm{SP}}$ are related to possible compensations of HVAC v for failures in $\overline{v}$.

Compensation Behaviour (CB). This feature is a specialisation of $\overline{\mathrm{SP}}$, that compares directly the temperature on opposite sides of a coach. $\mathrm{CB}_{c,d,v}$ is the difference between the in-coach temperature of HVAC v and of $\overline{v}$, telling the degree to which v must compensate for the possible lack of cooling of $\overline{v}$:

$$\mathrm{CB}_{c,d,v} = \frac{1}{N}\sum_{i=1}^{N} TI_i^{c,d,v} - TI_i^{c,d,\overline{v}}.$$

Comparison to Other Coaches (COC). The set-point temperatures among all coaches in a train should be almost equal. $\mathrm{COC}_{c,v}$ compares the average temperature on both decks of a coach $TI_i^{c,v} = \frac{1}{2}\big(TI_i^{c,1,v} + TI_i^{c,2,v}\big)$ against the corresponding median over all other coaches:

$$\mathrm{COC}_{c,v} = \frac{1}{N}\sum_{i=1}^{N} TI_i^{c,v} - \operatorname*{median}_{c'\neq c}\big(TI_i^{c',v}\big).$$

This helps to spot inactivity: if the train is in standby, the HVACs could be inactive and let temperatures raise, but this should happen in all coaches similarly.

Defective Control Sensor (DCS). This feature tries to determine whether the control sensors in a coach are defective, which would result in an incorrect set temperature value. DCS_c compares the temperature measured by the sensor TC^c of coach c against the median over all other coaches:

$$\mathrm{DCS}_c = \frac{1}{N}\sum_{i=1}^{N} TC_i^c - \operatorname*{median}_{c'\neq c}\big(TC_i^{c'}\big).$$

ML model. These features can be used to train a Logistic Regression classifier. We choose LR over other ML solutions because temperature data is unidimensional, so a linear classifier should suffice to divide HVAC malfunctions from their normal operation. Indeed, linear models can be high-accuracy detectors of HVAC availability [3, 10]. Furthermore LR is known to be computationally fast, resistant against overfitting, and it can produce probabilistic values (indicating failure likelihood), as opposed to a binary output (defective/healthy).

After training and assessing this ML model, the features and permutation importance can be extracted, to determine which features contribute the most to the generalisation power of the model. As minimal-detection boundary a *random binary feature* can be introduced, which contains no information and hence no detection power. Computing the importance of this random feature helps to understand the relative importance of the other features.

2.3 Study on rolling stock of NS

We used the approach defined above to detect HVAC failures in the historic data for 2 months of operation (in summer) of 176 double-deck trains of NS.

Data. A sample in this experiment is a 1 h window corresponding to one HVAC, for which all features were computed. The time step was 30 min. The resulting class distribution had an imbalance of ratio 5:1 in favour of the healthy-label class, which we balanced via class weights as is common practice [7].

Assessment. The performance of the resulting LR classifier was measured via the Receiver Operating Curve (ROC), summarised with the area under the ROC (AUC); and also with the Precision Recall Curve (PRC), summarised with the average precision score (PRS). For this we used a 4-repeated stratified 4-folded cross validator [6]; we also grouped samples per HVAC, to avoid comparisons of temperature readings (human-labelled vs. ML-predicted) across different units. For stratification we added an indicator of whether the groups contain samples of the hot or healthy class, with the resulting distribution 17 (hot) vs. 71.

Main result. Fig. 3 shows how the data samples are automatically labelled by the resulting LR model. The probability of cooling failure of either of the HVACs in a coach is indicated by the value of the colour on a bar below the temperature lines: darker red indicates higher malfunction probability.

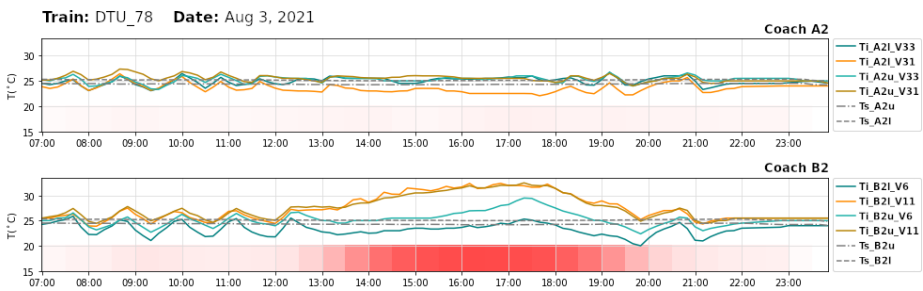
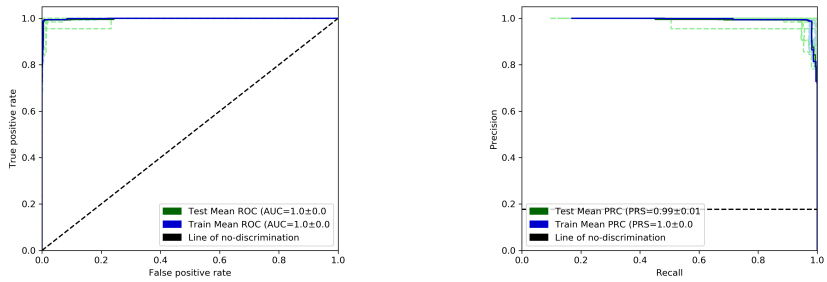


Figure 3: Detection of HVAC malfunctions by LR model from temperature readings

Model performance. Figs. 4a and 4b show respectively the ROC and PRC for this model. The dashed green lines are the ROC (and PRC) for the splits on either the test or train data; the solid blue line is their mean. The AUC of the ROC is 1.00 on average for both test and train sets, *denoting an approximately perfect score of the LR model, to detect both healthy and malfunctioning HVACs.* The PRC points in the same direction, with PRS values of 1.00 and 0.99 on average resp. for the train and test sets, which further indicates that *the model is not overfitting on the training data.*

Inter-rater reliability performance. We also computed the ROC and PRC to validate the human-labelled sets as indicated in Sec. 2.1. There were 191 samples (1 % of the data available) distributed randomly and labelled independently by two authors of this work. The resulting AUC and PRS values were

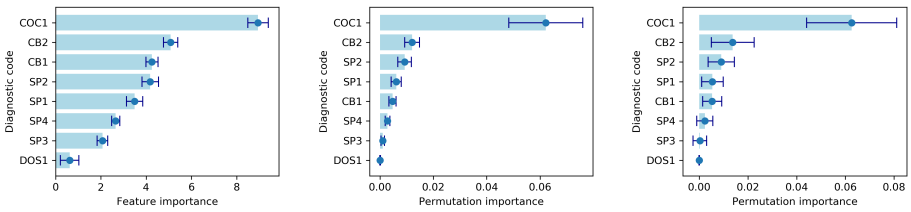


(a) Receiver Operating Curves (ROC) **(b)** Precision Recall Curves (PRC)

Figure 4: Performance of the LR classifier from temperature readings

0.99 on average for both cases, again denoting an excellent classification and a *negligible bias in the bootstrapping used as artificial ground truth.*

Feature importance. Fig. 5 shows the features and permutation importance of the LR model. The average feature importance (across all splits) is visualized with horizontal light-blue bars; the blue whiskers are their 95 % confidence intervals. Fig. 5 shows that features COC1 (Comparison to Other Coaches) and CB2 (Compensation Behaviour for the upper deck) are consistently ranked as the most predictive features, in that order. In contrast, DOS1 (Defect Control Sensor), SP3, and SP4 (Set Point for the complementary sides of the coaches), are the least useful features. Furthermore, the ranks for train and test sets in the permutation tests are equal, indicating that the model is not overfitting; and only features with very similar importance change among ranks.



(a) Absolute LR coefficients **(b)** Based on test set **(c)** Based on training set

Figure 5: Importance of features used for the LR classifier from temperature readings

3 Learning HVAC failures from diagnose codes

To reach objectives O2 and O3 we use the diagnose codes automatically sent by the HVACs to the data hub of the company. The intention is to detect malfunctioning HVACs online: with present data only, and also based on past readings. This section introduces our steps for data preprocessing and automatic labelling, defines the features used for classification, and explains the training and testing steps. The section ends with our studies in the trains of NS.

3.1 Data preparation

Input. This is digital data: a diagnose code (e.g. "high pressure in valve V_{A2}") identifies an event triggered by an HVAC component, including its activation and deactivation time, e.g. $x = (id : 333, start : 658210639, end : 658216644)$. We format this input data as a time series of codes corresponding to HVAC components. So for each time point and for each HVAC, we are able to know whether diagnose code x is ON or OFF.

Preprocessing. Code IDs must be chosen or computed s.t. an ID matches a type of symptom, regardless of the unit where it occurred. For example if the label of a code includes the HVAC or train numbers where it happened, this must be stripped from the label used as unique ID, e.g. by grouping. Diagnose codes without an ID or activation time must be discarded. Instead, missing deactivation times can be imputed if there is a standard deactivation time of all HVACs, e.g. when the trains are shut down for the night. We observed missing deactivation times on less than 1 % of the data analysed from NS, for which we inserted the deactivation time corresponding to the end of the working day.

Data labelling. We apply the proposed ML layering: use the LR model from the previous step, to build an artificial ground truth of HVAC malfunctions used in this step for training and testing. Therefore, unlike the previous step where the main input was human-generated and covered 1–5 % of the data, here the AGT is automatically generated (via the previous LR model) and covers the whole dataset. We highlight that this scheme hinges on obtaining high-quality extrapolations in the previous step, such as those presented in Sec. 2.3.

Technically, this requires to match the discretisation used for the temperature time series, to that of the diagnose code time series. The work-day window of a train is too coarse: although an HVAC might be broken, the functional failure is only noticeable when cooling is needed, e.g. in the afternoon when the outside temperature rises. Moreover the windows and labels chosen must permit feature extraction: for the digital input in this step, all features are based on the presence of codes during the time periods when HVACs fail (or are deemed healthy).

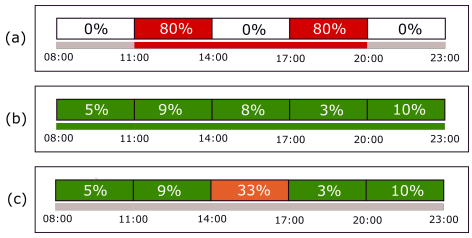


Figure 6: An example of windows that divide the work day in 5 sections, for which we show the aggregated probability of HVAC malfunctions coming from the AGT. Periods that are labelled healthy (green), defect (red), or none (gray), are indicated by colours in the narrow bar under the windows.

A generic solution is to split the work day in windows, labelled as healthy or defect (or none) using the AGT, i.e. the aggregated failure probabilities. This can be parameterised with thresholds $0 < T_\ell < T_h < 1$ such that the window corresponding to an HVAC is healthy if its probability of failure is below T_ℓ; if it is above T_h the window is labelled defect; and if the probability falls in $[T_\ell, T_h]$

no label is applied. Further logical conditions include e.g. no healthy labels for HVACs whose failure probability is above T_ℓ at any point of the day.

3.2 Machine learning experiment

For objective O2 we define one feature that indicates whether a code is present during a healthy- or defect-labelled period. For O3 we define three features that quantify the occurrence of the code in the time before a labelled period. Thus all features need to match codes to labelled HVAC periods, which we do as follows.

The i-th occurrence of diagnose code x is given by a time interval $[x_i^{\text{ON}}, x_i^{\text{OFF}}]$. Similarly, for an HVAC period p labelled healthy or defect we have $[p_{start}, p_{end}]$ (an HVAC period p is implicitly linked to a specific train and coach). We say that *code x is present during period p* if there is an occurrence of x in which these time intervals overlap: $\exists\, i\,.\, [x_i^{\text{ON}}, x_i^{\text{OFF}}] \cap [p_{start}, p_{end}] \neq \varnothing$, as illustrated by the red segments in Fig. 7.

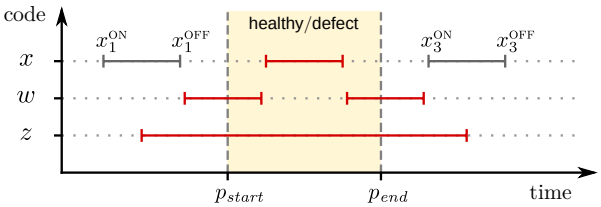


Figure 7: Occurrence of diagnose codes during healthy- or defect-labelled periods of an HVAC. In practice, p is divided into time windows as in Fig. 6.

Code During Period (CDP). This is the Boolean feature used for objective O2. It determines, for each (time window of each) healthy/defect period p, whether diagnose code x was present during p:

$$\text{CDP}_{p,x} \;\equiv\; \exists\, i\,.\, (x_i^{\text{ON}} \leqslant p_{end}) \wedge (x_i^{\text{OFF}} \geqslant p_{start}).$$

Code: Number of Days (CND). This integral feature is used for objective O3. For a time window equal to the periodic maintenance of the HVACs, $\mathcal{T}$, and backwards from the end of a period p, it counts the number of days since code x was last observed, introducing NaN if it was not observed:

$$\text{CND}_{p,x} = \text{CDP}^{\mathcal{T}}_{p,x} \;\mathbf{?}\; \text{days}\big(p_{end} - \max_i\{x_i^{\text{OFF}} \mid x_i^{\text{OFF}} \geqslant p_{end} - \mathcal{T}\}\big) \;\mathbf{:}\; \texttt{NaN},$$

where $(\mathit{cc} \;\mathbf{?}\; \mathit{tt} \;\mathbf{:}\; \mathit{ff})$ is the ternary operator on condition cc, true case tt, and false case ff, and $\text{CDP}^{t}_{p,x}$ is the CDP feature bounded from below in time by $p_{end} - t$.

Code: Number of Occurrences (CNO). This feature is similar to CND, but counts the number of occurrences of code x in the interval $[p_{end} - \mathcal{T}, p_{end}]$:

$$\text{CNO}_{p,x} = \text{CDP}^{\mathcal{T}}_{p,x} \;\mathbf{?}\; \#\{x_i \mid x_i^{\text{OFF}} \geqslant p_{end} - \mathcal{T}\} \;\mathbf{:}\; \texttt{NaN}.$$

Code: Cumulative Time (CCT). This floating-point feature counts the total time that code x was active in the time window $[p_{end} - \mathcal{T}, p_{end}]$:

$$\text{CCT}_{p,x} = \text{CDP}^{\mathcal{T}}_{p,x} \;\mathbf{?}\; \sum\{x_i^{\text{OFF}} - x_i^{\text{ON}} \mid x_i^{\text{OFF}} \geqslant p_{end} - \mathcal{T}\} \;\mathbf{:}\; \texttt{NaN}.$$

ML models. These features can be used to train different kinds of classifiers. For objective O2 we use logistic regression (as with O1), to check whether the full set of diagnose codes has good detection capabilities of HVAC malfunctions.

However, the components of an HVAC produce different codes, which could have different importance to predict a general failure. It would be useful to learn how each code contributes to the failure probability of the whole HVAC. This is in the best interest of companies, whose data streams already contain these codes, and thus could implement simple predictive maintenance rules such as "if codes x and w are seen together, send the HVAC to maintenance within n days".

For this purpose we also train a Decision Tree model (DT) for both objectives O2 and O3. Besides learning to estimate malfunctions, DTs can unfold the estimate to indicate how much each feature contributes to the total probability.

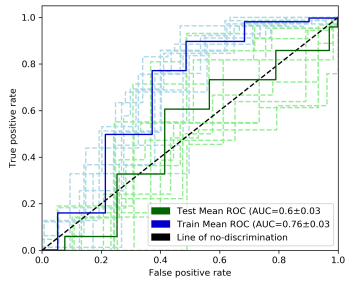
3.3 Study on rolling stock of NS

We apply this approach to the same trains used in Sec. 2.3. However, the features for objectives O2 and O3 are computed from two inputs: the diagnose codes, and the AGT computed from temperature readings in the previous step.

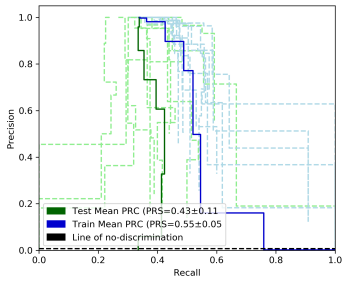
Data and assessment for objective O2. Only periods that contained a maximum ("outside") control temperature above 24 °C were used, to ensure that the HVACs had to cool down the train. On top of that, periods without active diagnose codes were discarded, since the source of information for CDP (the feature used for objective O2) is the intersection between diagnose codes and healthy/defect periods. The resulting data set contained 294 points, distributed with an imbalance ratio of 2:1 in favour of the healthy class. This data was used to train an LR model, and also a DT model. These were assessed via the ROC and PRC, equivalently to what was done for the LR model of the temperature readings. Data was stratified by grouping samples by HVAC, and adding an indicator on whether these groups included samples of the healthy class (66 % of the total), defect (20 %), or both (14 %). Also here we added a random binary feature with no detection power, to draw the line of useful codes when we build the ranking based on feature importance.

Data and assessment for objective O3. The scheduled maintenance for the fleet under consideration occurs approximately every 3 months. Therefore we chose $\mathcal{T} = 12$ weeks, which results in a large dataset on which to compute the features CND, CNO, CCT defined for this objective. To alleviate computations we randomly selected 1000 healthy periods, but kept all (625) defect ones: this reduced the data imbalance without hindering our intention to foretell members of the defect class. From this set we also filtered out periods with max temperature below 24 °C, and which did not coincide with any diagnose code (in their $[p_{end} - \mathcal{T}, p_{end}]$ time window). This resulted in a balanced data set, with 499 members of the healthy class and 496 of the defect class. We used this to train a DT classifier, but omitted the LR classifier since the data was too sparse and we expected no further gain w.r.t. DT. Assessment was performed as for the previous objective: in this case the groups including samples of the healthy class added up to 61 % of the total, and defect was 24 %.

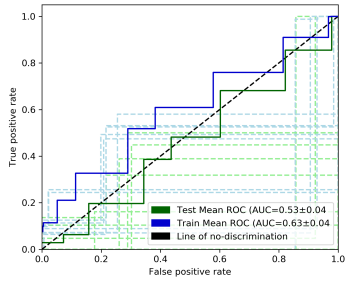
Performance of the LR and DT models for objective O2. Figs. 8a to 8d show the ROC and PRC curves obtained for the two models trained on the CDP feature for objective O2. As before, the blue lines are the average of different splits between train and test sets. The intention was to use the diagnose codes to detect HVAC failures, i.e. determine which codes occur during functional loss of these units. The corresponding curves in Fig. 8 show that the correlation between the features and the defect (or healthy) periods of the HVACs is very low. The AUC for the ROC of the LR model is 0.6 on average, and the PRS for the PRC is 0.43. These low values suggest that the codes in our dataset, or the CDP feature used here, cannot detect HVAC failures in real time. However, the low values can also be linked to issues with the previous step, e.g. the AGT built with the LR classifier could be inaccurate. We discuss this further in Sec. 4.2.



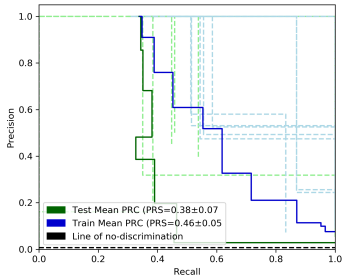
(a) ROC of LR model for objective O2



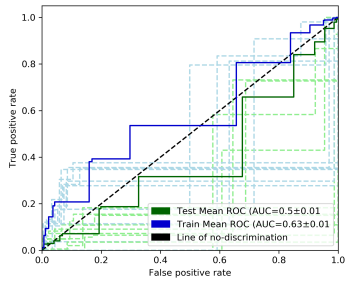
(b) PRC of LR model for objective O2



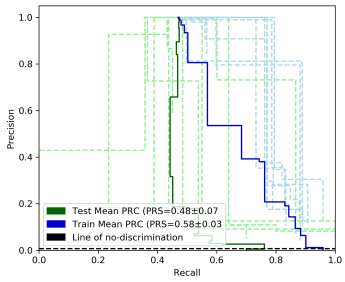
(c) ROC of DT model for objective O2



(d) PRC of DT model for objective O2



(e) ROC of DT model for objective O3



(f) PRC of DT model for objective O3

Figure 8: Performance of ML models from different features of diagnose codes

Performance of the DT models for objective O3. Figs. 8e and 8f show resp. the ROC and PRC curves for the decision tree model built on features CND, CNO, and CCT. These use past information, which was expected to help the DT classifier. However Fig. 8 shows differently, as the resulting ROC and PRC are 0.5 and 0.48 resp. As above, this suggests that the diagnose codes, or all features used to interpret them, cannot be faithfully used to detect HVAC failures *for the given dataset used*. In that sense it is important to highlight that our data covered only two (summer) months, which might well be insufficient to discover the correlations sought. We touch upon this point again in Sec. 4.2.

Feature importance. The rankings of feature importance (not included here) are another indicator of the low detection/prediction capacity of the dataset and features used for objectives O2 and O3. For the LR model, the permutation tests always place the random feature among the 10 most important codes. For the DT models of both feature sets (CDP on one side, and $\{\mathrm{CND}, \mathrm{CNO}, \mathrm{CCT}\}$ on the other) this does not happen, but the best-ranked code is an HVAC self-test, which is not expected to have true predictive capability of HVAC failures.

4 Final discussion and perspectives

4.1 Temperature readings to detect HVAC failures

The results of the LR classifier in Sec. 2.3 make it clear that **it is possible to identify HVACs with cooling failures based on temperature readings.**

We note that the performance of our model could be experiencing a positive bias, related to the conservative rules applied to manually label hot or healthy periods. This could have resulted in relatively easy predictions for the NS input, as we intentionally ignored anomalies, and focused on data with high probability of coming from a defective (or healthy) HVAC.

To further improve the capabilities and scope of application of this model, one could use ground-truth labels of HVACs malfunctions (if available), to reduce any potential bias introduced by our manual annotation process. If that is not feasible, then manually labelling more data and employing multiple human annotators—as we show in Sec. 2.3—should also reduce any bias.

Regarding feature rankings, the COC feature (Compare to Other Coaches) is clearly the most relevant to determine HVAC malfunctions. Its importance seems nearly high enough to base all predictions on it alone. However there are scenarios where the SP (Set Point) and DCS (Defect Control Sensor) features are required, e.g. if the control sensors of the coaches have a manufacturing failure. Moreover, defects in the outside temperature sensor are interesting on their own, since they require different maintenance than HVAC systems, and may cause misbehaviour of any train component that depends on their data.

Other features for this dataset appear less important and could be removed without deteriorating the quality of the resulting LR model. We note however that another practical solution to remove features is to count with information that indicates when the HVAC is actively heating/cooling. In that case it should be possible to work with the SP features alone.

4.2 Diagnose codes to detect and predict HVAC failures

The results of Sec. 3.3 show that **we could not find strong evidence of correlation between diagnose codes and HVAC failures**, given our limited dataset and choice of LR and DT classifiers.

Further analysis revealed that no codes are present in 97 % of the periods labelled as healthy by our LR model from the previous step. This could indicate good data quality, but also no codes are present in 80% of the periods labelled as defect. In fact, the tables built for the features (of intersections between healthy/defect periods and diagnose codes) were very sparse, e.g. 85 % of the total data was purely zeroes for the CDP feature computed for objective O2.

This shows few intersections between HVAC malfunctions and codes in general, despite the high accuracy of the LR model that built the AGT. Such lack of intersections could be explained by a relative absence of codes, that in turn might have two explanations: the HVACs may be sending more events than the ones received, with messages being lost in the data stream; or the diagnose system of the HVACs could have limitations, thus sending less codes than required.

Nevertheless, we highlight that the diagnose codes are designed to identify specific errors in subcomponents of the HVAC. It may be that, contrary to our initial hypothesis, the codes in our dataset do not cover failures in components that can be related to cooling failures in the HVACs. Overfitting is yet another explanation for this result, since a few samples (i.e. independent cases of broken HVACs) are trying to be classified with many dimensions (i.e. diagnose codes). Moreover, the AGT used for this step comes from ML extrapolation of human-labelled data. However good the accuracy of the resulting LR model, it remains to be proved that the defect-labelled periods truly correspond to defective HVACs.

4.3 Perspectives

In Sec. 4.1 we indicate two ways to improve the usability of the LR model based on temperature readings: add ground-truth data regarding HVAC malfunctions; and add activation data regarding HVAC cooling/heating.

Regarding features and their importance, **the outcome of our feature engineering process—see Secs. 2.2 and 3.2—could be generalised to other rolling stocks or systems**. These features can be used on any train, or vehicle, or building, whose architecture is similar to those of this study. That is: having a main unit (train/building), divided in connected adjacent compartments (coaches/rooms), each with an HVAC and one or more temperature sensors. In particular, even though our studies are focused on cooling, the same can be done for heating, and also for ventilation e.g. using CO_2 sensor readings.

Another promising extension would be to use the temperature readings to foretell a malfunction. For instance, in our rules for manual labelling we disregard zigzag patterns as those observed in Fig. 3. It could be possible to build an ML model capable of finding correlations between these (or other) temperature patterns in time, that serve to estimate a probability of the HVAC failing before the next scheduled maintenance. For this we foresee the use of neural networks

and frequency analysis, which are more complex (and more opaque, but also more powerful) than the simple and explainable LR approach followed here.

Regarding the diagnose codes, a first attempt to improve our results should use a larger dataset, e.g. covering several years. Also, the same features could be used in other sets of codes, e.g. from other train types, to determine whether it is the feature engineering process (and not the dataset) that requires revision.

Beyond HVAC cooling failures, diagnose codes are promising to guide the first steps of maintenance. For this, maintenance data serves as ground-truth to indicate the cause behind an HVAC failure. Further data analyses would reveal the preceding diagnose codes, which could be used in future failures to indicate to technicians where to look first, as a new broken HVAC arrives for repair.

Acknowledgements. The authors thank Nick Oosterhof, who contributed with invaluable discussion and feedback that helped to carry out and shape this work.

References

1. Aslansefat, K., Kabir, S., Gheraibia, Y., Papadopoulos, Y.: Dynamic Fault Tree Analysis: State-of-the-Art in Modelling, Analysis and Tools, pp. 73–112. Taylor & Francis (2020). https://doi.org/10.1201/9780429268922-4
2. Catelani, M., Ciani, L., Guidi, G., Patrizi, G., Galar, D.: Estimate the useful life for a heating, ventilation, and air conditioning system on a high-speed train using failure models. ACTA IMEKO **10**(3), 100–107 (2021)
3. Daniel, R., Russell, R., Billing, J., Schonewill, P., Burns, C., Shimskey, R., Peterson, R.: Filtration understanding: FY10 testing results and filtration model update. Tech. rep., Pacific Northwest National Laboratory (2011)
4. Hale, P., Arno, R.: Survey of reliability and availability information for power distribution, power generation, and HVAC components for commercial, industrial, and utility installations. In: IEEE Ind. Commer. Power Syst. Tech. Conf. (Cat. No.00CH37053). pp. 31–54 (2000). https://doi.org/10.1109/ICPS.2000.854354
5. Lin, N., Du, W., Wang, J., Yun, X., Chen, L.: The effect of COVID-19 restrictions on particulate matter on different modes of transport in China. Environmental Research (2021). https://doi.org/10.1016/j.envres.2021.112205
6. Ojala, M., Garriga, G.C.: Permutation tests for studying classifier performance. J. Mach. Learn. Res. **11**, 1833–1863 (2010)
7. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
8. Ruijters, E., Guck, D., Drolenga, P., Peters, M., Stoelinga, M.: Maintenance analysis and optimization via statistical model checking: Evaluation of a train's pneumatic compressor. In: QEST. LNCS, vol. 9826, pp. 331–347. Springer (2016)
9. Settles, B.: Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning (2012). https://doi.org/10.2200/S00429ED1V01Y201207AIM018
10. Tehrani, M.M., Beauregard, Y., Rioux, M., Kenne, J.P., Ouellet, R.: A predictive preference model for maintenance of a heating ventilating and air conditioning system. IFAC **48**(3), 130–135 (2015). https://doi.org/10.1016/j.ifacol.2015.06.070
11. Wong, D.: A knowledge-based decision support system in reliability-centered maintenance of HVAC systems. Ph.D. thesis, University of Newfoundland (2000)