Refrigerant leak detection in industrial vapor compression refrigeration systems using machine learning

Amal Mtibaa, Valentina Sessa, Gilles Guerassimoff, Stéphane Alajarin

International Journal of Refrigeration 161 (2024) 51–61

Contents lists available at ScienceDirect

International Journal of Refrigeration

journal homepage: www.elsevier.com/locate/ijrefrig

Refrigerant leak detection in industrial vapor compression refrigeration systems using machine learning

Détection de fuite de liquide frigorigène dans les installations frigorifiques industrielles à compresseurs de vapeur utilisant l'apprentissage automatique

Amal Mtibaa [a,b,*], Valentina Sessa [a], Gilles Guerassimoff [a], Stéphane Alajarin [b]

[a] *Mines Paris-PSL, Centre de Mathématiques Appliquées, Sophia Antipolis, France*
[b] *Clauger, Clauger 3E Department, Brignais, France*

ARTICLE INFO

Keywords:
Refrigeration
Vapor compression system
Leakage detection
Machine learning

Mots clés:
Réfrigération
Système à compression de vapeur
Détection de fuite
Apprentissage automatique

ABSTRACT

Efficient detection of refrigerant leakage is of utmost importance for industrial refrigeration systems due to its potential to cause substantial impacts on system performance and the environment. Existing research on fault detection and diagnosis in refrigeration systems primarily revolves around solutions based on experimental or laboratory data. However, in the industrial use case, achieving accurate and early detection poses significant challenges. This paper reports on the development of a novel refrigerant leak detection method for industrial vapor compression refrigeration systems. Our method leverages real-world data obtained from operational installations, enabling us to assess its reliability and applicability. The proposed data-driven approach involves predicting the fault-free liquid level in the installation receiver and comparing the actual and predicted levels. In this work, we place emphasis on features and model selection. Dedicated metrics combined with a model comparison method are proposed to evaluate and compare the performance of commonly used regression models with two sets of features to determine the most effective one. Furthermore, we provide insights into the results obtained from the deployment of the proposed method in real-world industrial installations.

1. Introduction

Over the years, the industrial refrigeration market has developed significantly due to its widespread use in various applications, including food and beverage processing, cold storage, and others. Conditioned by different parameters such as production scale, temperature difference, and expected temperature, various customized installations can be used to provide industrial refrigeration, all of which have the potential to leak, as the internal pressure is usually many times higher than atmospheric (Tassou and Grace, 2005).

Refrigerant loss is a critical issue that can tremendously impact the refrigeration system, and it can go undetected in the case of gradual leaks until a significant amount of refrigerant has been lost. The impacts of this fault can be categorized as follows.

- **Environmental:** Leakage of fluorinated gases in refrigeration systems, considered in this work, affects the environment directly through their global warming potential. Moreover, when refrigerant leaks occur, the efficiency of the system is reduced, causing higher energy consumption (Francis et al., 2017; Grace et al., 2005).
- **Financial:** Refrigerant leakage can impose substantial financial implications, depending on the time it took to be discovered and repaired (Francis et al., 2017; ETSU, 1997; Koronaki et al., 2012). The financial impact stems from the cost of wasted fluid, expenses associated with maintenance and repairs, and potential production losses and catastrophic damages caused by mechanical failures.

In the last two decades, research has given considerable attention to maintaining Heating, Ventilation, Air Conditioning, and Refrigeration (HVAC&R) systems using fault detection and diagnosis (FDD) techniques for both component-level and system-level faults (Singh et al., 2022). While the component-level faults are relatively easy to discover, system-level faults, such as refrigerant leakage, are challenging to detect and diagnose (Gao et al., 2022).

* Correspondence to: 5 Rue du docteur Reure, Saint-genis-laval, 69230, France.
E-mail addresses: amal.mtibaa@minesparis.psl.eu, amtibaa@clauger.fr (A. Mtibaa), valentina.sessa@minesparis.psl.eu (V. Sessa), gilles.guerassimoff@minesparis.psl.eu (G. Guerassimoff), salajarin@clauger.fr (S. Alajarin).

https://doi.org/10.1016/j.ijrefrig.2024.02.016
Received 19 July 2023; Received in revised form 25 January 2024; Accepted 12 February 2024
Available online 16 February 2024

Nomenclature

°Csat	Degree Celsius saturated
Cp	Compressor
EP	Electric power
EXT	Extremely randomized trees
FDD	Fault detection and diagnosis
GB	Gradient boosting
HVAC&R	Heating, ventilation, air conditioning, and refrigeration
$\mathbb{I}$	Indicator function
LGB	Light gradient boosting
RF	Random forest
SVR	Support vector regression
XGBoost	Extreme gradient boosting

In the literature related to refrigerant leakage, Tassou and Grace (2005) proposed a data-driven method to differentiate normal (fault-free), undercharge, and overcharge conditions in vapor compression chillers using Artificial Neural Networks. In Navarro-Esbri et al. (2006), a dynamic black-box model-based method was proposed for detecting leakage in vapor compression chillers for both transient and steady-state operations. The recent paper Takeuchi and Saito (2018) proposed a fault diagnosis method for refrigerant leak detection based on soft sensor techniques, which considers the physical model and the control mechanism of commercial air conditioners. Similarly, Yoo et al. (2017) worked on air conditioners but for the residential use case by proposing a model-free knowledge-based method. Many papers investigating refrigerant leakage deal with air conditioning centrifugal chillers in buildings (Zhao et al., 2013a; Tran et al., 2015, 2016; Han et al., 2019; Yao et al., 2022; Gao et al., 2022). In these papers, data-driven FDD methods are proposed using the ASHRAE RP-1043 experimental data (Comstock and Braun, 1999). FDD was performed on a list of faults having four levels of severity each, including refrigerant leakage. In particular, Zhao et al. (2013a) proposed creating reference models that calculate the benchmarks of four fault indicators (referred to as performance indexes) in normal conditions, using Support Vector Regression (SVR) in combination with exponentially weighted Moving Average (EWMA) control charts for fault detection. Once detected, the fault can be diagnosed through the proposed rules set. In Tran et al. (2015), using an additional feature in the fault indicators allowed considering a further fault, and a Radial Basis Function (RBF) neural network was proposed to replace SVR to enhance prediction accuracy. The authors in Tran et al. (2016) improved the FDD performance by alternatively using the least squares SVR algorithm and enhancing its accuracy using differential evolution. In Han et al. (2019), support vectors were used as a classification tool trained on eight fault-indicative features. Furthermore, Yao et al. (2022) compared three tree-based ensemble learning algorithms with SVR and proposed using light gradient boosting with a multivariate EWMA control chart.

In the previously discussed studies, the absence of a liquid receiver, which is a commonly used component in industrial settings, makes the examined systems unable to compensate for the lost fluid during a refrigerant leak. This results in more immediate variations in fault indicators, thereby increasing the sensitivity of the suggested methods for leak detection. Although the basic refrigeration cycles are similar, HVAC&R systems can vary in many aspects from one sector of applications to another (Behfar et al., 2017). The literature on FDD applied to industrial systems, particularly regarding refrigerant leakage is relatively limited. A possible reason could be the complexity and the challenges faced by FDD methods in industrial applications. For instance, one major issue is the quality of data captured by the sensors, which are often noisy, incomplete, and uncertain (Zhao et al., 2013b). In addition, multiple faults can occur simultaneously, and differentiating between each fault symptom can be challenging (Chen et al., 2022).

Similar complexities and challenges encountered in the industrial use case can also be observed in supermarket refrigeration systems (Behfar et al., 2017, 2019). In Srinivasan et al. (2015), an FDD method applied to real-world supermarket systems addresses four faults (including refrigerant leakage) by proposing a seasonal auto-regressive integrated moving average method based solely on energy signals for fault detection. In the scenario of refrigerant leakage, a statistical model is created to monitor the liquid level in the receiver. A leak is identified when a significant drop in the average liquid level occurs. However, this model neglects the impact of operational and external conditions on the fluctuations in the liquid level in the receiver.

In this work, we propose a data-driven refrigerant leak detection method for industrial vapor compression refrigeration systems. To the best of our knowledge, this is the first work focusing on leakage for this kind of system within the industrial use case. The main contribution of our work is outlined in what follows:

- We propose to work on direct expansion refrigeration systems equipped with a liquid receiver, presenting notable challenges in leak detection.
- A comparative analysis of commonly employed machine learning regression methods is proposed to predict the fault-free liquid level inside the receiver.
- The proposed system can detect gradual leaks regardless of the physical installation characteristics (e.g., system size, number of units, type of refrigeration fluid) and regardless of the operation condition (steady-state and transient).
- Differently from most proposed FDD approaches that are typically performed in limited or specific settings (e.g., in a laboratory) (Singh et al., 2022), we consider working on real-world installations.

The structure of the paper is as follows. First, the refrigeration system under consideration and data description are introduced in Section 2. Section 3 provides a concise overview of the machine learning models used in this work and presents the techniques employed for model evaluation and performance comparison. Afterward, we present the feature and model selection results in Section 4, together with the leak detection process in action on two real-world installations. The paper is concluded in Section 5.

2. Background

2.1. System description

This work considers an industrial direct expansion vapor compression refrigeration system. A basic vapor compression refrigeration system consists mainly of four components: a compressor, a condenser, an expansion valve, and an evaporator. It operates by compressing a refrigerant gas to raise its pressure and thus its saturated temperature, then condensing it into a liquid through a condenser which dissipates the refrigerant heat to the ambient environment. Afterward, the expansion valve decreases the pressure of the refrigerant. Finally, the low-pressure, low-temperature refrigerant enters the evaporator pipes to absorb the heat from the cold room environment. A receiver is integrated into the system to hold the excess refrigerant not in circulation. In a direct expansion system, refrigerant circulates through the installation pipes and interacts directly with the cold source for expansion and evaporation.

Given our focus on real-world cases, there is no universal installation schema. Each scenario presents unique refrigeration requirements, resulting in varied specifications and characteristics. As shown in Fig. 1, system components can be in the form of racks, with each

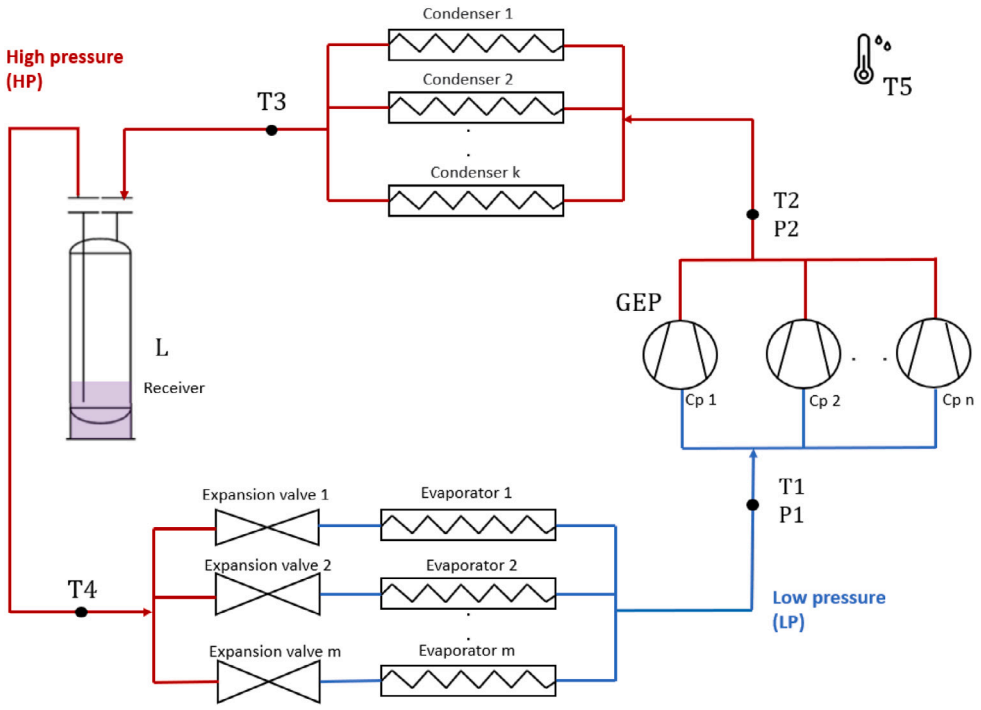


Fig. 1. Simplified schematic diagram of the considered refrigeration system.

Table 1
Installation measured sensors values.

Temperatures [°C]	**Pressures [°Csat, bar]**	**Power [kW]**
T1 – Compressor suction temperature	**P1** – Compressor suction pressure	**GEP** – General electric power
T2 – Discharge outlet temperature	**P2** – Compressor discharge pressure	$= \sum_{i=1}^{n} EP(Cp_i)$
T3 – Refrigerant temperature at condenser outlet		
T4 – Liquid temperature in high pressure	**Liquid level [mm]**	
T5 – Ambient temperature	**L** – Refrigerant liquid level in the receiver	

rack housing one or multiple units. Temperatures before and after each major component are measured. Additionally, two pressure transducers are employed to measure refrigerant pressure respectively at compression suction and discharge. Furthermore, an ambient temperature sensor is utilized, along with the general electric power (GEP) calculated as the aggregate of compressors electrical powers. Finally, the liquid level is determined by assessing the pressure differential between the upper and lower sections of the receiver. Components units may have varying capacities but share a common inlet and outlet. Consequently, the collected sensor data are independent of the number of present units.

Although the use of experimental and laboratory data could be interesting for preliminary FDD analysis, experiments performed under real-world conditions are necessary to take into account its complexity (Momeni et al., 2021). In particular, we consider seven real-world industrial installations deployed in different sectors of activities, as well as a specifically designed one to acquire more real-world data. Table 2 summarizes the main characteristics of these installations, each identified by an alphabetical 'id', the refrigerant fluid name, the number of compressors, and the quantity of the fluid initially charged in the receiver.

Table 2
Characteristics of the considered installations.

id	Fluid	Compressors	Fluid initial quantity [kg]
Installation A	R404A	4	300
Installation B	R449A	3	135
Installation C	R407F	5	430
Installation D	R449A	4	432
Installation E	R404A	4	350
Installation F	R404A	5	2000
Installation G	R449A	2	70
Installation H	R404A	1	7

2.2. Data description

Working with data obtained from functioning real-world systems, careful consideration is required when selecting data to create a fault-free dataset. As discussed above, this is due to the limited information available regarding potential faults, some of which may not be considered urgent and can remain unrepaired for a long time. Hence, to ensure the collection of an appropriate fault-free dataset, we adhered to the following set of rules:

- for each installation, we ensure that the data are collected when no maintenance operation takes place;
- we collect a reasonable amount of data for training the learning models, taking into consideration the constraints set by the available computational resource;
- we collect data over diverse time slots along a calendar year and varying time gaps between training and test sets;
- to ensure the absence of any known sensor faults, we employ rule-based data quality verification methods.

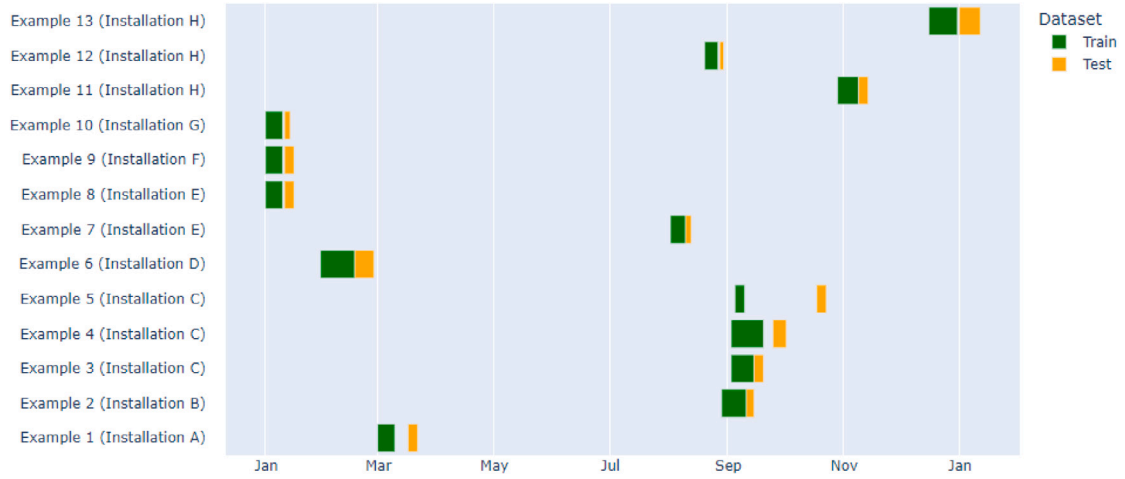


Fig. 2. Fault-free database.

Following these rules, we created a fault-free database with thirteen data examples. Fig. 2 displays the time ranges of each example corresponding to the indicated installation and distinguishing between training and test data.

Although the majority of existing FDD literature focuses on steady-state operation, we propose keeping startup transient data despite the increased uncertainty associated with it for the following reasons:

- Removing transient data can lead to a delay in detecting refrigerant leaks, as leaks can indeed occur during these periods. This delay can result in significant liquid losses, especially when encountering rapid leaks that require immediate attention.
- Excluding transient startup data in scenarios involving successive startups and shutdowns, which can result from compressor failures, could lead to removing a significant portion of the analyzed data. In such cases, omitting transient startup data would make it impracticable to detect potential leaks.
- Detecting the steady-state operation in systems with multiple compressors can be computationally expensive and complex.

To mitigate the impact of transient startup data on the construction of accurate models, we first reduce noise in each training set by resampling values at 5-min intervals. Second, we apply an anomaly detection method based on the operational regime (i.e., the ON/OFF state of the installation) that identifies data points that do not fall into the usual patterns. This approach identifies and excludes anomalous data points that may affect the modeling process adversely.

3. Methodology

To determine the most suitable method for predicting the liquid level in the receiver, we propose comparing multiple regression models. A preliminary selection process considering various regression and forecasting techniques such as tree-based methods, support vector algorithms, neural networks, statistical models, and the Facebook forecasting tool Prophet (Taylor and Letham, 2018), was initially conducted. This selection was based on several factors, including the adaptability of the models to resource limitations, training time requirements, performance, and sensitivity to data drifts. Based on these factors, we have chosen four variants of ensemble tree-based models and Support Vector Regression (SVR) for further analysis. This section briefly explains these methods, along with the used techniques for model evaluation and comparison.

3.1. Brief theoretical background

In learning regression problems, a training set of samples $\{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$ is used to obtain an estimate $\hat{f}(\mathbf{x})$ of a function $f(\mathbf{x})$ that maps $\mathbf{x}$ to y. In our case, we consider a set of features $\mathbf{x} \in \mathbb{R}^m$ and $y \in \mathbb{R}$. The approximated map can be obtained by implementing different methods. We present here the ones considered in this paper.

Tree-based models are popular machine learning techniques for regression and classification tasks. These models build a decision tree by recursive binary splitting the predictor space into simple regions. A prediction for a given observation is then computed using the mean or the mode of the training observations in the region it belongs to James et al. (2017). Each split involves a decision about selecting one of the m features and the computation of a splitting point.

The most significant advantage of a decision tree is its interpretability. Conversely, the prediction accuracy is generally low, and they suffer from serious variance-bias balance issues. An ensemble of trees can be built parallel or sequentially to overcome these drawbacks. Common ensemble techniques include:

- Bootstrap AGregation, or bagging, is a technique commonly used to reduce the variance of statistical methods. The idea is to create many training datasets and build multiple prediction models using each training set. Then, the final prediction is the average of each model output. Multiple training datasets are created by bootstrap, that is, by sampling from the original dataset with replacement. Each tree is deep and built without pruning, so it has high variance and low bias. However, averaging the output of multiple trees reduces the variance and improves the prediction accuracy. Random Forest (RF) (Breiman, 2001) models take this concept one step further. Indeed, to create decorrelated trees, each split in the tree is only allowed over a random subset of features. In the Extremely Randomized Trees (EXT) (Geurts et al., 2006), another degree of randomness is added in the choice of the splitting point. This makes it more efficient than RF in reducing the variance and also computationally faster. Moreover, EXT was shown to be less affected by the presence of noise in the features.
- Boosting is another technique that could be applied to general methods. In the case of decision trees, small trees are built sequentially, with each tree focused on correcting the error of the previous one (James et al., 2017). The idea behind this technique is that instead of building one large decision tree, which would cause overfitting, it creates multiple small decision trees so that it learns slowly. In Friedman (2001), the residuals for a given model are shown to be proportional to the negative gradients of the mean squared error loss function with respect to the function

space $f(\mathbf{x})$. Hence, the boosting algorithm could be regarded as a gradient descent algorithm, and it is known as Gradient Boosting (GB) (Friedman, 2001). GB becomes computationally inefficient when the number of features and data points in the training set increases, basically because all the features and data have to be considered when deciding on the feature and the splitting point. The Light GB (LGB) (Ke et al., 2017) overcomes these issues by reducing data instances based on gradient evaluation and reducing the number of effective features, exploiting the sparsity of the feature space. Differently, XGBoost (Chen and Guestrin, 2016) implements several improvements to GB including an additional regularization term to avoid overfitting, a tree learning algorithm that better handles sparse features, and a distributed weighted sketch algorithm to propose candidate splitting points.

The next method to be presented extends the well-known support vector machine to regression problems (Bishop, 2006). SVR searches for the regression curve fitting the data points inside a fixed-width tube using a so-called ϵ- insensitive error function. The model parameters $\mathbf{w}$ can be computed by solving the dual of the following regularized optimization problem:

$$\min_{w,\xi,\hat{\xi}} \quad C\sum_{i=1}^{N}(\xi_i+\hat{\xi}_i)+\frac{1}{2}\|\mathbf{w}\|^2 \tag{1a}$$

$$\text{s.t.} \quad \xi_i \geq 0, \hat{\xi}_i \geq 0,\ i=1,\dots,N \tag{1b}$$

$$\hat{f}(x_i) \leq y_i+\epsilon+\xi_i,\ i=1,\dots,N \tag{1c}$$

$$\hat{f}(x_i) \geq y_i-\epsilon-\xi_i,\ i=1,\dots,N, \tag{1d}$$

where ξ and $\hat{\xi}$ are slack variables, and C is the inverse of the regularization parameter. As in the classical SVM, the dual formulation allows the introduction of a kernel function that generalizes the concept of the scalar product in a higher dimensional case. In our case, preliminary experiments showed that by using the Gaussian Radial Basis Function (RBF) kernel, accurate results can be achieved.

3.2. Model evaluation metrics

Model evaluation is an important step for assessing its performance and determining its effectiveness in solving a particular problem. Several evaluation metrics can be used to quantify the quality of a model prediction, fine-tune it, and compare it with other methods. The choice of evaluation metrics depends on the task of the model (classification, regression, clustering), its implementation, and the specific application at hand. Relying on a single evaluation metric can be misleading, as it may not capture the different aspects of a model performance. Therefore, we suggest considering multiple evaluation metrics to comprehensively evaluate the model strengths and weaknesses.

For regression problems, evaluation metrics consist of errors that assess how close the predicted values are to the real ones. Mean Absolute Error (MAE) and Mean Square Error (MSE) are commonly used error metrics, defined as follows,

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i-\hat{y}_i|, \tag{2}$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2, \tag{3}$$

where $\hat{y}$ is the predicted value, y is the actual value, and N is the total number of samples.

On top of these errors, we present customized metrics that showed a more significant impact on our application. First, we introduce two metrics inspired by MAE, where we propose to split our data into small chunks, each containing p samples. Hence, instead of computing the difference between individual values point by point, we calculate the difference between the mean (respectively the median) of these data chunks. The Mean Absolute of the Chunk Mean Error (MA-CME) and the Mean Absolute of the Chunk Median Error (MA-CMdE) are defined as follows,

$$\text{MA-CME} = \frac{p}{N}\sum_{i=0}^{\frac{N-p}{p}}|mean(\{y_{ip+j}\}_{j=1}^{p})-mean(\{\hat{y}_{ip+j}\}_{j=1}^{p})| \tag{4}$$

and

$$\text{MA-CMdE} = \frac{p}{N}\sum_{i=0}^{\frac{N-p}{p}}|median(\{y_{ip+j}\}_{j=1}^{p})-median(\{\hat{y}_{ip+j}\}_{j=1}^{p})| \tag{5}$$

where N/p is the number of chunks.

Furthermore, taking into account the inherent uncertainty in liquid sensors, we suggest also evaluating the model performance by calculating the percentage of data points with a residue that falls within an acceptable error range. This metric, denoted as AE-AR$_{\%}$ (percentage of Absolute Error within Acceptance Range), is computed as follows:

$$\text{AE-AR}_{\%} = \frac{100}{N}\sum_{i=1}^{N}\mathbb{I}(|y_i-\hat{y}_i|\leq\epsilon_1), \tag{6}$$

where $\mathbb{I}$ is the indicator function and ϵ_1 is a small positive value.

Building upon the aforementioned metrics, we also introduce the percentage of Absolute Chunk Mean Error within an Acceptable Range (ACME-AR$_{\%}$), and the percentage of Absolute Chunk Median Error within an Acceptable Range (ACMdE-AR$_{\%}$), defined as

$$\text{ACME-AR}_{\%} = \frac{100p}{N}\sum_{i=0}^{\frac{N-p}{p}}\mathbb{I}(|mean(\{y_{ip+j}\}_{j=1}^{p})-mean(\{\hat{y}_{ip+j}\}_{j=1}^{p})|\leq\epsilon_2) \tag{7}$$

and

$$\text{ACMdE-AR}_{\%} = \frac{100p}{N}\sum_{i=0}^{\frac{N-p}{p}}\mathbb{I}(|median(\{y_{ip+j}\}_{j=1}^{p}) - median(\{\hat{y}_{ip+j}\}_{j=1}^{p})|\leq\epsilon_3), \tag{8}$$

where ϵ_2 and ϵ_3 are small positive values.

3.3. Model performance comparison

With multiple evaluation metrics, the model performance comparison could be challenging. While some metrics may favor one model over another, others may provide different results altogether. Due to the inherent differences across installations, a distinct model for each installation should be created to accurately predict the liquid level.

To compare model performances considering different evaluation metrics, we propose calculating the percentage of times the model under consideration produces the best evaluation metric values above the others (or closely approaches the best model value within a specified tolerance). For instance, suppose we want to compare the performance of m different regression models (M_1, M_2, …, M_m). For each Example i, we train them on the same training set and evaluate each model by calculating the evaluation metrics presented in Section 3.2 on the corresponding test set. Afterward, we identify, for each metric j, the model providing the best value. Note that since the best value can be obtained by multiple models (within a certain tolerance), we indicate with S_i^j the set of top-performing models according to metric j for Example i. We illustrate this procedure in Fig. 3.

Finally, we define $P(M_m)$ as the sum of the total times the model M_m has given the best values, divided by the number of examples n (13 for the fault-free database) and the number of evaluation metrics k (7 when considering all previous metrics). Hence, we have

$$P(M_m) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{I}(M_m\in S_i^j)}{k*n}. \tag{9}$$

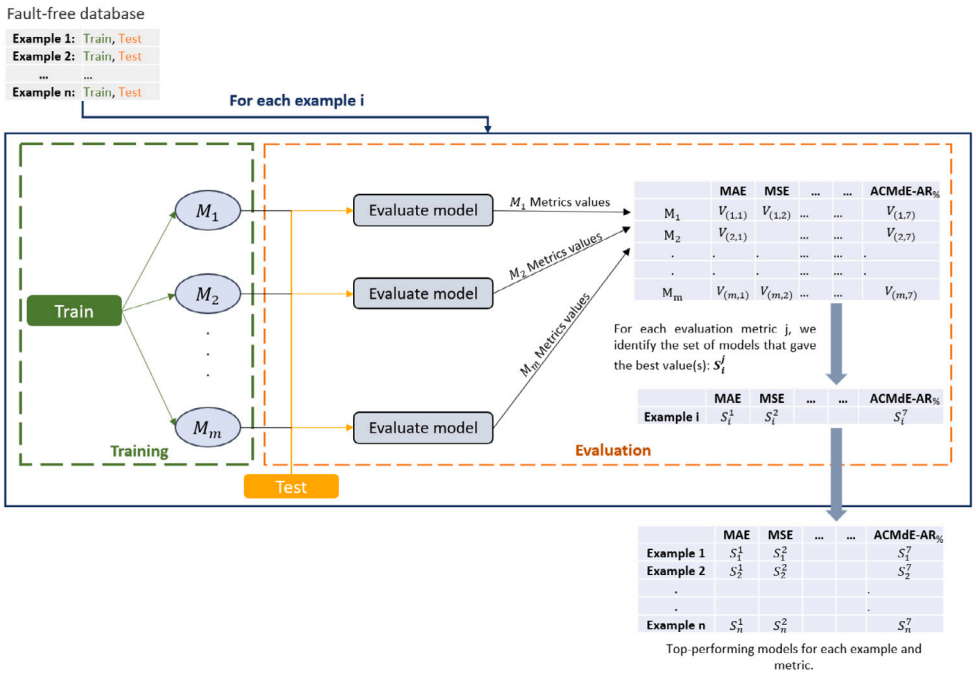


Fig. 3. Procedure for determining the most performing models on the fault-free database.

Table 3
List of considered hyper-parameters.

Tree-based models hyper-parameters	Model(s)
n_estimators: number of trees in the forest.	All
max_depth: max number of levels in each decision tree.	All
min_samples_split: min number of data points placed in a node before the node is split.	EXT, GB
min_samples_leaf: min number of data points allowed in a leaf node.	EXT, GB
learning_rate: controls the impact of each tree prediction on the overall model.	GB, LGB, XGBoost
reg_alpha: controls the L1 regularization strength on leaf weights.	LGB, XGBoost
reg_lambda: controls the L2 regularization strength on leaf weights.	LGB, XGBoost
boosting_type: used strategy to train the boosting model.	LGB, XGBoost
gamma: controls the minimum loss reduction required to split a node.	XGBoost
SVR hyper-parameters	
gamma: determines the influence of each training sample on the model decision boundary.	
C: controls the trade-off between achieving a low training error and a small margin.	

The higher $P(M_m)$, the better M_m is compared to others. The proposed method enables evaluating the robustness of a model performance across multiple data examples while considering multiple metrics. To ensure a valid comparison, a substantial number of data examples need to be used. Additionally, this method can be employed to compare how different data preparation features and methods affect the performance of a single regression model.

3.4. Hyper-parameter tuning and model selection

The hyper-parameters of a machine learning model represent an external configuration that cannot be estimated from the data. Tuning them is an essential part of creating a robust predictive model. Cross-Validation (CV) (James et al., 2017) is a widely used technique for hyper-parameter tuning. In particular, K-Fold CV is commonly utilized for regression tasks. In K-Fold CV, the training set is randomly shuffled and divided into K subsets of equal size. Then, $K-1$ subsets are used to train the model, and the remaining one is used as a validation set. The training and validation are repeated K times, each time choosing a different validation set and training a model on the remaining ones. Note that since we work with time series, we do not randomly shuffle data points when splitting the folds (Snijders, 1988).

To identify the most suitable hyper-parameter settings, we suggest combining the K-Fold CV with the Bayesian optimization search technique (Bishop, 2006) . This approach uses Bayes theorem to direct the search for an extreme point of a given objective function. With this aim, it builds a probabilistic model of the objective function, called the surrogate function, that can be efficiently sampled. Along with being fast, it does not treat hyper-parameter sets independently, as it keeps track of past evaluation results and focuses on those areas of the parameter space that could bring the most promising validation scores.

Sequential model-based optimization (SMBO) (Horn et al., 2015; Hutter et al., 2011) methods represent a formalization of Bayesian optimization with trial points being run one after another, attempting better hyper-parameters each time by applying Bayesian reasoning and updating the surrogate model.

The Python library Hyperopt (Bergstra et al., 2015), which we propose to use, employs SMBO as its underlying optimization algorithm. Hyperopt is employed to hyper-tune and select between multiple models (i.e., EXT, GB, LGB, XGboost, SVR with RBF kernel (SVR_Rbf)). The defined objective function calculates the mean of 5-fold CV scores, with MAE as the scoring method. The list of considered hyper-parameters and their corresponding models are presented in Table 3.

4. Experiments

The proposed refrigerant leak detection method is based on a data-driven approach that models the fault-free liquid level in the receiver.

Table 4
Considered system performance indicators.

Feature	Formula	Indication
Suction overheating	$T1 - P1_{(°CSat)}$	• Operating conditions of the evaporator. • Level of liquid retention in the evaporator.
Discharge overheating	$T2 - P2_{(°CSat)}$	• Compression efficiency.[a]
Condenser outlet subcooling	$P2_{(°CSat)} - T3$	• Condensation efficiency. • Level of liquid retention in the condenser.
Liquid subcooling	$P2_{(°CSat)} - T4$	• System liquid line temperature difference.[b] • Condensation efficiency.
Condenser approaching temperature	$P2_{(°CSat)} - T5$	• Condensation efficiency. • Heat transfer optimality. • Level of liquid retention in the condenser.[c]
Compression ratio	$\frac{P2_{(bar)}+1}{P1_{(bar)}+1}$	• Compression performance and efficiency.

[a] When sufficient liquid charge is present.
[b] The system liquid line is the pipe connecting the condenser coil to the expansion valve. The temperature difference depends on the thermal exchange with the outside of the liquid line, which in turn depends on refrigerant flow, liquid line route, insulation, external conditions, etc.
[c] In combination with the condenser outlet subcooling.

A leak is identified when the measured liquid level drops in comparison to the predicted one. This section begins by discussing feature engineering and selection. It then presents the results obtained from hyper-parameter tuning and model selection. Finally, an overview of the leak detection process using the chosen model is provided, along with preliminary results demonstrating its effectiveness in detecting leaks in two real-world installations.

4.1. Feature engineering

The measured features, displayed in Fig. 1, consist of the installation temperatures (T1, T2, T3, T4), the compression pressures (P1, P2), the ambient temperature (T5), and the general electrical power (GEP). A regression model that takes these features as input can be used to predict the liquid level in the receiver (L). Moreover, changes in the refrigeration system can happen frequently. Whether it is because of seasonal weather fluctuations, maintenance activities, or even the occurrence of potential faults, these events can significantly vary the relationships between features, as well as their patterns and ranges of values. Since most regression models can only obtain accurate predictions with inputs similar to those used for training, they need to be re-trained to cope with a new system behavior. In addition to being computationally expensive, it is challenging to identify accurately the events that necessitate a model re-training. As a practical way to minimize the need for frequent model training, we investigated system performance indicators commonly used in the industrial environment to understand and monitor the operating conditions of a refrigeration circuit across its various phases. The proposed features, along with their formulas and system indications, are presented in Table 4.

4.2. Feature selection

Each refrigeration installation possesses unique characteristics, implying that the optimal combination of input features for a given model may differ. However, our practical goal is to identify a common set of features capable of effectively predicting fault-free liquid levels across these systems, while minimizing the necessity for frequent model re-training. We use the wrapper feature selection method (Jović et al., 2015) to assess the quality of each model given a selected ensemble of input features. Indeed, testing all possible feature combinations on each example of the fault-free database and for each model is computationally demanding and time-consuming. As a result, we propose to compare models performances using the following two ensembles of features:

– **Raw data:** representing the sensor measured data (see Table 1).

Fig. 4. $P(M_m)$ on the fault-free database for each modeling method respectively when using raw data and calculated data.

– **Calculated data:** representing the calculated performance indicators as in Table 4 + GEP.

Using the model comparison approach described in Section 3.3, we evaluate the performance of each modeling method when trained using each ensemble of features separately. At this stage, default hyper-parameters are used in training. In Fig. 4, we display the value of $P(M_m)$ for each model when trained on raw and calculated data. The observed percentages reveal a slightly better performance of tree-based models when using calculated data for training. However, these results are significantly more evident in the case of SVR models.

Seasonal changes (such as variations in ambient temperature, humidity levels, sunlight, and wind) are frequent events that have a notable impact on the refrigeration system (Stoecker, 1998; Whitman et al., 2012). Specifically, as the ambient temperature rises, the pressure and temperature of the refrigerant in the receiver increase, leading to expansion and a rise in the liquid level. Conversely, colder ambient temperatures under similar operating conditions can cause the refrigerant to contract, resulting in a lower liquid level. Further analysis revealed that the accuracy of predicting the liquid level is significantly influenced by the selection of feature ensemble when dealing with examples that exhibit substantial variations in input feature values due to seasonal changes. To delve deeper into this behavior, we consider an additional example from installation F, say Example 14, where we analyze a training set collected during winter and a test set collected in the midst of spring. During this gap, substantial changes in the ambient temperature, plus potentially unknown events and faults, have occurred. These events caused significant changes and fluctuations

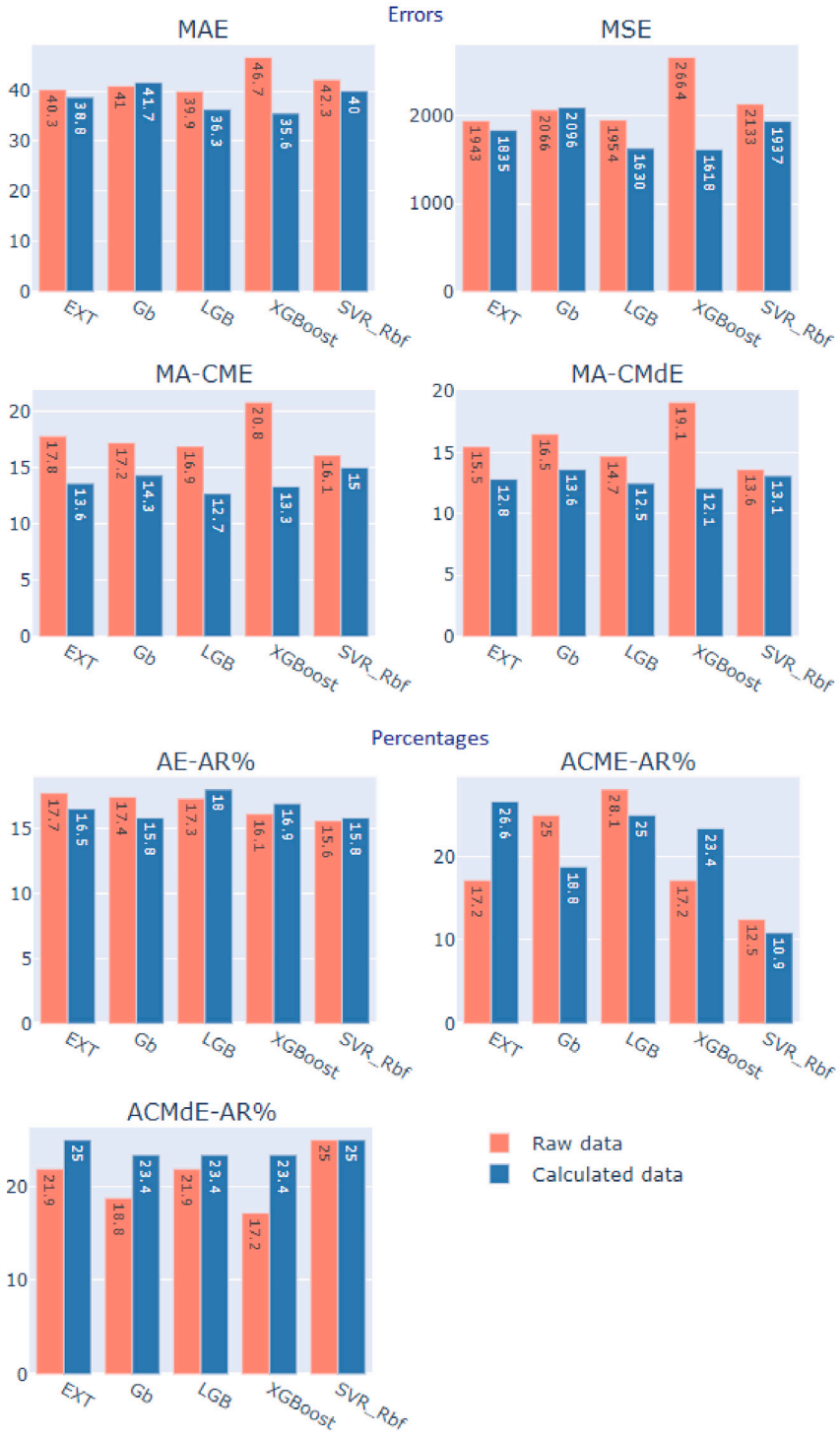


Fig. 5. Performance metrics on Example 14, for each model when trained using raw and calculated data.

in the measured features, including the liquid level. The prediction accuracy of models using raw data proved to be lower than those using calculated data, as shown by the performance metrics in Fig. 5.

Motivated by these results, we consider solely examples that exhibit significant temperature variations, namely Examples 1, 4, 5, and 14, and we compute $P(M_m)$ for each model when trained using raw and calculated data. Fig. 6 clearly demonstrates that using calculated data yields superior performances across all models, highlighting the significant role of feature selection in such scenarios. Consequently, calculated data is the selected ensemble of features that we propose to use in the remainder of this paper.

4.3. Model selection

As explained in Section 3.4, to identify the model that exhibits optimal performance for our specific case, we perform hyper-parameter tuning and model selection using the Python library HyperOpt. The output of this optimization process includes the selected best model and the hyper-tuned parameters. This information, along with the MAE calculated on the test set of each example, are presented in Table 5.

HyperOpt selects the EXT model for almost all the examples but two, while LGB and XGBoost are selected for only one example each. Since LGB is slightly faster than XGBoost, and both follow a similar

Table 5
Model selection results on the fault-free database.

Example id	Model	Tuned hyper-parameters	MAE
Example 1	EXT	n_estimators: 1222, min_samples_split: 6, min_samples_leaf: 1, max_depth: 10	4.1
Example 2	LGB	n_estimators: 302, max_depth: 5, learning_rate: 0.016, reg_alpha= 72, reg_lambda: 0.671	9.2
Example 3	EXT	n_estimators: 1928, min_samples_split: 7, min_samples_leaf: 1, max_depth: 100	18.6
Example 4	EXT	n_estimators: 1541, min_samples_split: 2, min_samples_leaf: 3, max_depth: 20	16.8
Example 5	EXT	n_estimators: 749, min_samples_split: 2, min_samples_leaf: 1, max_depth: 70	12.3
Example 6	EXT	n_estimators: 1389, min_samples_split: 5, min_samples_leaf: 1, max_depth: 1110	37.8
Example 7	EXT	n_estimators: 490, min_samples_split: 5, min_samples_leaf: 1, max_depth: 20	13.9
Example 8	EXT	n_estimators: 412, min_samples_split: 3, min_samples_leaf: 1, max_depth: 100	32.6
Example 9	EXT	n_estimators: 928, min_samples_split: 8, min_samples_leaf: 3, max_depth: 10	1.6
Example 10	EXT	n_estimators: 1909, min_samples_split: 6, min_samples_leaf: 1, max_depth: 20	13.1
Example 11	EXT	n_estimators: 589, min_samples_split: 9, min_samples_leaf: 1, max_depth: 90	0.5
Example 12	XGBoost	booster: gbtree, n_estimators: 1613, gamma: 0.083, max_depth: 14, reg_alpha: 93, reg_lambda: 0.906	0.7
Example 13	EXT	n_estimators: 1001, min_samples_split: 6, min_samples_leaf: 1, max_depth: 10	1.5

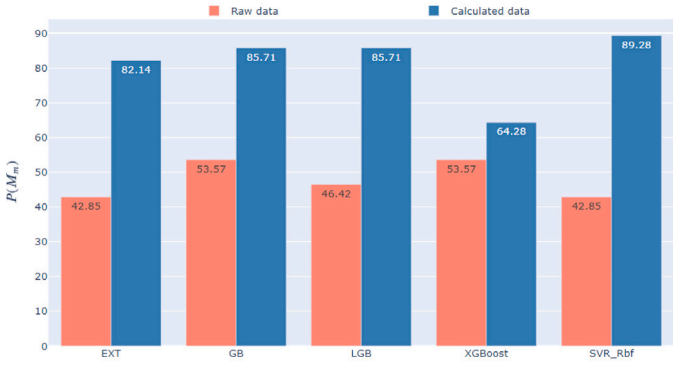


Fig. 6. $P(M_m)$ for Examples 1, 4, 5, and 14 for each model trained using raw data and calculated data.

approach, we propose to further compare EXT and LGB. HyperOpt applied exclusively on EXT and LGB chooses EXT as the top-performing model for all examples but Example 2.

Furthermore, we individually fine-tune each model by running the optimization procedures over a larger number of iterations. Once the optimal hyper-parameters are set, we test the created models over all the examples. Calculating the percentages $P(M_m)$, EXT achieves 80.2%, while LGB achieves 68.1%. Based on these results, it is evident that EXT consistently outperforms the other models. Therefore, it is the method we select for predicting the liquid level in the receiver.

4.4. Leak detection at work

A trained, hyper-tuned EXT model is used to predict the fault-free liquid level in each installation receiver. A scheduled task is set up for each installation to compare the actual and predicted values. This comparison is conducted through an adapted version of the dynamic threshold method proposed by Chakraborty and Elzarka (2019), which adapts to transient behaviors and has exhibited superior effectiveness compared to the conventional application of fixed thresholds.

Due to the impracticability of leak simulation in real-world scenarios, initial experiments were performed on the specifically designed installation H. Afterward, the proposed approach was progressively employed in real-world industrial refrigeration systems. Up to the present moment, our methodology has been deployed in 19 installations, resulting in the detection of 14 leaks with different severity levels.

In Fig. 7, we display two examples of identified leaks. In Fig. 7(a), a gradual leak is illustrated, with a fluid loss of 34.7% occurring over more than 13 days. On the other hand, Fig. 7(b) shows a rapid leak, where 57.4% of the fluid was lost within 8 h. For the gradual leak depicted in Fig. 7(a), the system initiated generating alerts sporadically starting on July 19 and eventually became consistent from July 25. While the system could detect the leak at an early stage, technicians in charge noticed it only on August 2 through manual detectors. In the case of the rapid leak, alerts were triggered 2 h after the beginning of the leak. A failure in the evaporation process led to a complete loss of the refrigerant and resulted in a system shutdown.

Observations in real-world refrigeration systems have revealed that gradual refrigerant leakage can persist for extended periods without significantly impacting system performance. Changes in the system become evident only when a substantial amount of refrigerant has been lost. The proposed approach has been proven effective in detecting both gradual, as well as sudden, abrupt leaks. Upon confirmation of a leak, technicians promptly intervene to initiate the necessary repairs to the system. Depending on the situation, they can choose to add fluid to the receiver. To ensure continuous adaptation to evolving system conditions, an automated mechanism is implemented to detect system changes that could arise from these interventions. Once such changes are detected in an installation, the model is automatically trained.

5. Conclusions

This paper proposes a refrigerant leak detection method for industrial vapor compression refrigeration systems. Experiments were conducted on real-world data, and a fault-free database was specifically created to perform features and model selection. We compared the performance of four ensemble tree-based models and the Support Vector Regression method in predicting the fault-free liquid level in the receiver. Engineered features were proposed and proven more effective in long-term predictions compared to using raw sensor data. Subsequently, cross-validation combined with Bayesian optimization was employed to hyper-tune the models and select the most efficient one. The Extremely Randomized Trees method emerged as the top performer among these models. Leak detection in two real-world installations was presented, enhancing the applicability and relevance of our approach in real-world scenarios.

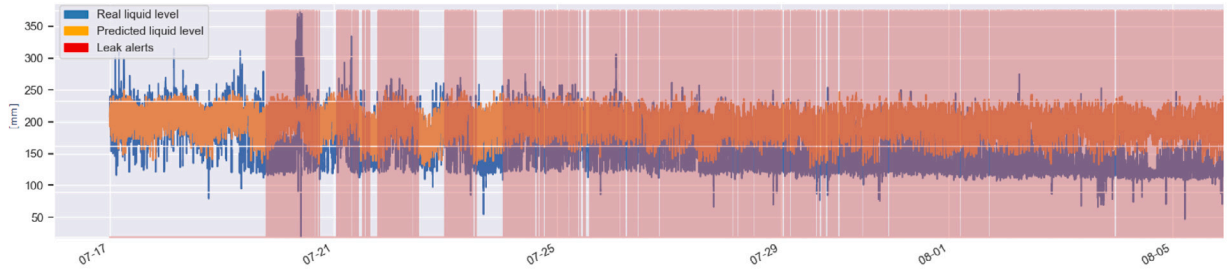
We aim in future work to go deep into leak detection and diagnosis by working on differentiating refrigerant leakage from faults and phenomena that could have similar symptoms (e.g., fluid migration to colder areas when the system is OFF), as well as quantifying the lost fluid.

Funding

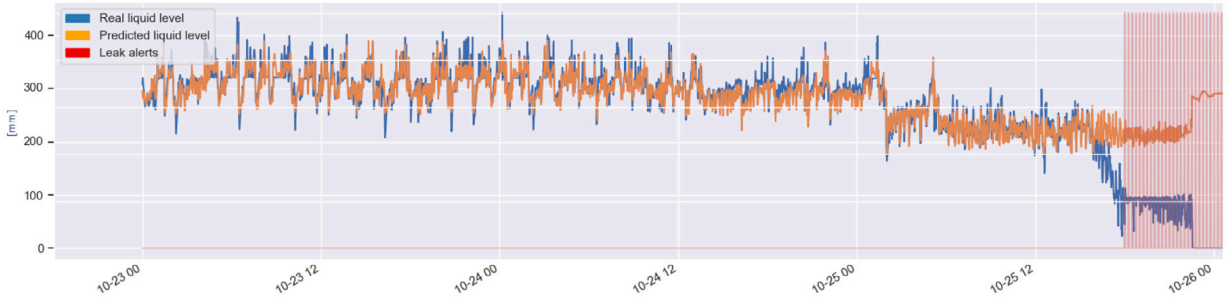
This research is funded by Clauger and the Ministry of Higher Education and Research as part of a doctoral program (CIFRE), France.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a) Gradual leak (installation D).



(b) Rapid leak (installation C).

Fig. 7. Example of leak detection in two installations.

Acknowledgment

We express our appreciation to the technical team of Clauger for their collaboration, with a special mention to Mr. Benoit Lacroix for his valuable guidance.

References

Behfar, A., Yuill, D., Yu, Y., 2017. Automated fault detection and diagnosis methods for supermarket equipment (RP-1615). Sci. Technol. Built. Environ. 23 (8), 1253–1266. http://dx.doi.org/10.1080/23744731.2017.1333352.

Behfar, A., Yuill, D., Yu, Y., 2019. Automated fault detection and diagnosis for supermarkets–method selection, replication, and applicability. Energy Build. 198, 520–527. http://dx.doi.org/10.1016/j.enbuild.2019.06.011.

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D., 2015. Hyperopt: a python library for model selection and hyperparameter optimization. Comput. Sci. Discov. 8 (1), 014008. http://dx.doi.org/10.1088/1749-4699/8/1/014008.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer New York, NY.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32. http://dx.doi.org/10.1023/A:1010933404324.

Chakraborty, Debaditya, Elzarka, Hazem, 2019. Early detection of faults in HVAC systems using an XGBoost model with a dynamic threshold. Energy Build. 185, 326–344. http://dx.doi.org/10.1016/j.enbuild.2018.12.032.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: The 22nd SIGKDD Conference on Knowledge Discovery and Data Mining. http://dx.doi.org/10.1145/2939672.2939785.

Chen, J., Zhang, L., Li, Y., Shi, Y., Gao, X., Hu, Y., 2022. A review of computing-based automated fault detection and diagnosis of heating, ventilation and air conditioning systems. Renew. Sustain. Energy Rev. 161, 112395. http://dx.doi.org/10.1016/j.rser.2022.112395.

Comstock, M.C., Braun, J.E., 1999. Development of Analysis Tools for the Evaluation of Fault Detection and Diagnostics for Chillers. TechnicalReport ASHRAE Research Project 1043-RP, HL 99-20, Report 4036-3.

ETSU, 1997. Cutting the Cost of Refrigerant Leakage: An Introductory Guide for Users of Small to Medium-Sized Refrigeration Systems. Technical Report 1998–1373, Energy Technology Support Unit, United Kingdom.

Francis, C., Maidment, G., Davies, G., 2017. An investigation of refrigerant leakage in commercial refrigeration. Int. J. Refrig. 74, 12–21. http://dx.doi.org/10.1016/j.ijrefrig.2016.10.009.

Friedman, J., 2001. Greedy function approximation: A gradient boosting machine. Ann. Statist. 29, 1189–1232.

Gao, Y., Han, H., Lu, H., Jiang, S., Zhang, Y., Luo, M., 2022. Knowledge mining for chiller faults based on explanation of data-driven diagnosis. Appl. Therm. Eng. 205, 118032. http://dx.doi.org/10.1016/j.applthermaleng.2021.118032.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63, 3–42. http://dx.doi.org/10.1007/s10994-006-6226-1.

Grace, I.N., Datta, D., Tassou, S.A., 2005. Sensitivity of refrigeration system performance to charge levels and parameters for on-line leak detection. Appl. Therm. Eng. 25 (4), 557–566. http://dx.doi.org/10.1016/j.applthermaleng.2004.07.008.

Han, H., Cui, X., Fan, Y., Qing, H., 2019. Least squares support vector machine (LS-SVM)-based chiller fault diagnosis using fault indicative features. Appl. Therm. Eng. 154, 540–547. http://dx.doi.org/10.1016/j.applthermaleng.2019.03.111.

Horn, D., Wagner, T., Biermann, D., Weihs, C., Bischl, B., 2015. Model-based multi-objective optimization: Taxonomy, multi-point proposal, toolbox and benchmark. In: International Conference on Evolutionary Multi-Criterion Optimization. http://dx.doi.org/10.1007/978-3-319-15934-8_5.

Hutter, F., Hoos, H.H., Leyton-Brown, K., 2011. Sequential model-based optimization for general algorithm configuration. In: Coello, C.a.C. (Eds) Learning and Intelligent Optimization. LION 2011, In: Lecture Notes in Computer Science, http://dx.doi.org/10.1007/978-3-642-25566-3_40.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2017. An Introduction to Statistical Learning: With Applications in R. Springer.

Jović, A., Brkić, K., Bogunović, N., 2015. A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics. MIPRO, IEEE, pp. 1200–1205. http://dx.doi.org/10.1109/MIPRO.2015.7160458.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems. NIPS, http://dx.doi.org/10.5555/3294996.3295074.

Koronaki, I.P., Cowan, D., Maidment, G., Beerman, K., Schreurs, M., Kaar, K., Chaer, I., Gontarz, G., Christodoulaki, R.I., Cazauran, X., 2012. Refrigerant emissions and leakage prevention across Europe–Results from the RealSkillsEurope project. Energy 45 (1), 71–80. http://dx.doi.org/10.1016/j.energy.2012.05.040.

Momeni, M., Jani, S., Sohani, A., Jani, S., Rahpeyma, E., 2021. A high-resolution daily experimental performance evaluation of a large-scale industrial vapor-compression refrigeration system based on real-time IoT data monitoring technology. Sustain. Energy Technol. Assess. 47, 101427. http://dx.doi.org/10.1016/j.seta.2021.101427.

Navarro-Esbri, J., Torrella, E., Cabello, R., 2006. A vapour compression chiller fault detection technique based on adaptative algorithms. Application to on-line refrigerant leakage detection. Int. J. Refrig. 29 (5), 716–723. http://dx.doi.org/10.1016/j.ijrefrig.2005.12.008.

Singh, V., Mathur, J., Bhatia, A., 2022. A comprehensive review: Fault detection, diagnostics, prognostics, and fault modeling in HVAC systems. Int. J. Refrig. 144, 283–295. http://dx.doi.org/10.1016/j.ijrefrig.2022.08.017.

Snijders, T.A.B., 1988. On cross-validation for predictor evaluation in time series. Lect. Notes Econ. Math. Syst. 307, 56–69. http://dx.doi.org/10.1007/978-3-642-61564-1_4.

Srinivasan, S., Vasan, A., Sarangan, V., Sivasubramaniam, A., 2015. Bugs in the freezer: Detecting faults in supermarket refrigeration systems using energy signals. In: Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems. pp. 101–110. http://dx.doi.org/10.1145/2768510.2768536.

Stoecker, Wilbert F., 1998. Industrial Refrigeration Handbook. McGraw-Hill Education.

Takeuchi, S., Saito, T., 2018. Fault diagnosis method based on scaling law for online refrigerant leak detection. In: 2018 17th IEEE International Conference on Machine Learning and Applications. ICMLA, IEEE, pp. 1087–1094. http://dx.doi.org/10.1109/ICMLA.2018.00177.

Tassou, S.A., Grace, I.N., 2005. Fault diagnosis and refrigerant leak detection in vapour compression refrigeration systems. Int. J. Refrig. 28 (5), 680–688. http://dx.doi.org/10.1016/j.ijrefrig.2004.12.007.

Taylor, S.J., Letham, B., 2018. Forecasting at scale. Amer. Statist. 72 (1), 37–45. http://dx.doi.org/10.1080/00031305.2017.1380080.

Tran, D.A.T., Chen, Y., Ao, H.L., Cam, H.N.T., 2016. An enhanced chiller FDD strategy based on the combination of the LSSVR-DE model and EWMA control charts. Int. J. Refrig. 72, 81–96. http://dx.doi.org/10.1016/j.ijrefrig.2016.07.024.

Tran, D.A.T., Chen, Y., Chau, M.Q., Ning, B., 2015. A robust online fault detection and diagnosis strategy of centrifugal chiller systems for building energy efficiency. Energy Build. 108, 441–453. http://dx.doi.org/10.1016/j.enbuild.2015.09.044.

Whitman, Bill, Johnson, Bill, Tomczyk, John, Silberstein, Eugene, 2012. Refrigeration and Air Conditioning Technology. Cengage Learning.

Yao, W., Li, D., Gao, L., 2022. Fault detection and diagnosis using tree-based ensemble learning methods and multivariate control charts for centrifugal chillers. J. Build. Eng. 51, 104243. http://dx.doi.org/10.1016/j.jobe.2022.104243.

Yoo, J.W., Hong, S.B., Kim, M.S., 2017. Refrigerant leakage detection in an EEV installed residential air conditioner with limited sensor installations. Int. J. Refrig. 78, 157–165. http://dx.doi.org/10.1016/j.ijrefrig.2017.03.001.

Zhao, Y., Wang, S., Xiao, F., 2013a. A statistical fault detection and diagnosis method for centrifugal chillers based on exponentially-weighted moving average control charts and support vector regression. Appl. Therm. Eng. 51 (1–2), 560–572. http://dx.doi.org/10.1016/j.applthermaleng.2012.09.030.

Zhao, Y., Xiao, F., Wang, S., 2013b. An intelligent chiller fault detection and diagnosis methodology using Bayesian belief network. Energy Build. 57, 278–288. http://dx.doi.org/10.1016/j.enbuild.2012.11.007.