REVIEW

Anomaly detection in quasi-periodic energy consumption data series: a comparison of algorithms

Niccolò Zangrando, Piero Fraternali, Marco Petri, Nicolò Oreste Pinciroli Vago and Sergio Luis Herrera González*

*Correspondence:
sergioluis.herrera@polimi.it
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133, Milan, Italy
Full list of author information is available at the end of the article

Abstract

The diffusion of domotics solutions and of smart appliances and meters enables the monitoring of energy consumption at a very fine level and the development of forecasting and diagnostic applications. Anomaly detection (AD) in energy consumption data streams helps identify data points or intervals in which the behavior of an appliance deviates from normality and may prevent energy losses and break downs. Many statistical and learning approaches have been applied to the task, but the need remains of comparing their performances with data sets of different characteristics. This paper focuses on anomaly detection on quasi-periodic energy consumption data series and contrasts 12 statistical and machine learning algorithms tested in 144 different configurations on 3 data sets containing the power consumption signals of fridges. The assessment also evaluates the impact of the length of the series used for training and of the size of the sliding window employed to detect the anomalies. The generalization ability of the top five methods is also evaluated by applying them to an appliance different from that used for training. The results show that classical machine learning methods (Isolation Forest, One-Class SVM and Local Outlier Factor) outperform the best neural methods (GRU/LSTM autoencoder and multistep methods) and generalize better when applied to detect the anomalies of an appliance different from the one used for training.

Keywords: Anomaly detection; Time series; Machine Learning.

Introduction

Appliance-level energy consumption monitoring is a core component of the control system of smart buildings [1, 2]. The consumption data can be either directly collected with such devices as smart plugs, or inferred with non intrusive load monitoring (NILM) algorithms able to break down the household aggregate consumption signal into the contributions of individual appliances [3]. The analysis of energy consumption data series enables forecasting and diagnostic applications, such as load prediction [4], anomaly detection (AD) [5] and predictive maintenance [6].

AD in temporal data series is the task of identifying data points or intervals in which the time series deviates from normality. AD finds application in different fields such as healthcare, where it applies to the analysis of clinical images [7] and of ECG data [8], cybersecurity, where it is used for malware identification [9], manufacturing, where it helps monitoring machines and prevent break downs [10], and in the utility industry, where it supports the early identification of critical

events such as appliance malfunctioning [11] and water leakage [12] [13]. In the energy field, AD may be combined with energy load forecasting to improve accuracy [14], or integrated as a component for detecting non nominal energy fluctuations for enhancing decision making in energy transfer between microgrids [15]. Energy consumption time series can be collected from home appliances and building systems with complex periodic or quasi-periodic behavior, such as coolers, water heaters and fridges, which present specific challenges when performing anomaly detection. Machine learning and neural models trained on normal data may overfit with respect to the length of the period. This phenomenon makes the model sensible even to small variations of the cycle duration, which can happen during normal functioning [16]. As a consequence, the detector may emit a high number of false positive alerts when such small variations occur and also may degrade its performances sensibly when used to detect anomalies of an appliance of the same type but with a different cycle duration.

The literature on AD in temporal data series still lacks a systematic comparison of algorithms belonging to different families on quasi-periodic data sets. Therefore the development of an AD application in such a scenario still has to confront with design decisions such as the choice of the most effective algorithm, the minimum duration of the time series to use for training, the minimum size of the signal prediction/reconstruction window needed to identify the anomalous behavior, and the portability of the chosen algorithm from one appliance to another one with "similar" behavior. This paper tries to fill the gap in the literature about AD in quasi-periodic time series by systematically comparing the performances of 12 algorithms representative of different families of approaches. The experiments were performed on 3 distinct data sets regarding the fridges power consumption.

The aim of the experiments is to address the following questions:

- **Q1** How do the selected algorithm compare in the AD task on quasi-periodic time series under multiple performance metrics?
- **Q2** For the algorithms that require training, what is the relationship between the length of the training series and the performances?
- **Q3** For the algorithms that exploit a window-based approach for the prediction, what is the relationship between the length of the window and the performances?
- **Q4** What is the generalization capability of the methods? How does performance degrade when a method trained on an appliance is tested on the time series produced by a distinct appliance of the same type?

The essential findings can be summarized as follows:

- The classical ML algorithms Isolation Forest (ISOF), One-Class SVM (OC-SVM), and Local Oulier Factor (LOF) outperform the best neural models (GRU/LSTM autoencoder and multisteps methods)
- Two weeks of training data are sufficient for most methods, with the multisteps approaches attaining a modest improvement if one month of data is used.
- The length of the prediction/reconstruction window has a different impact on neural and non-neural methods.
- ISOF and OC SVM are less dependent on the training set with respect to the neural models, which have a sensible performance decay when tested on an appliance different from the one used for training.

- The top result of all the experiments is attained by ISOF on the Fridge3 time series, trained with a sub-sequence of length equal to one month and with a window size of 2 × period: Precision = 0.947, Recall = 0.965, F_1 score = 0.956.

The above mentioned findings can help understand better the requirements and performances of AD algorithms on quasi-periodic data series so as to design more effective household energy consumption applications, e.g., by equipping the mobile apps that are nowadays bundled with smart plug products with functionalities for consumption monitoring, energy saving recommendations and alerting of potential appliance malfunctioning.

The rest of the article is organised as follows: Section Related work overviews the state of the art in anomaly detection. Section Experimental settings describes the experimental configuration, including the description of the dataset and of the evaluated algorithms. Section Experimental results discusses the results of the performed experiments. Section Qualitative analysis of results discusses qualitatively a few examples of the predictions made by the reviewed methods. Finally, Section Conclusions draws the conclusions and illustrates our future work.

Related work

Anomaly detection in temporal data series exploits data collected with a broad spectrum of sensors in diverse fields, such as weather monitoring, natural resources distribution and consumption (e.g., water and natural gas), network traffic surveillance, and electrical load measurement [17] [18] [19] [20]. As an example, the work in [19] discusses the use of residential home smart meters for data collection and highlights how such series often exhibit anomalous behaviors. Raw data must be pre-processed to get ready for further analysis. Besides the usual operations of data cleaning and validation, a prominent task is data annotation, which associates data points or intervals with the specifications of significant events, such as change points and anomalies. For example, Rimor [21] is a time-series data annotator supporting the labelling of data with anomaly tags, which can be used as ground truth for training and evaluating predictive models.

AD can be conducted in both univariate [22] and multivariate time series [23] [24] [25]. In the case of multivariate time series, exploiting variable correlation may be necessary for reducing the number of parameters needed to model the problem [26]. Examples of multivariate time series dimensionality reduction techniques are principal components analysis [27] [26], canonical correlation analysis [28], and factor modelling [29].

AD approaches can be classified in two main families [27]: non-regressive and regressive. Non-regressive approaches rely on the fundamental statistical quantities computed on the time series (e.g., mean and variance) and combine them with fixed thresholds, but their effectiveness is limited [27]. The authors of [30] proposed a statistical AD framework using the Dickey-Fuller test, the Fourier transform, and the Pearson correlation coefficient to analyze periodic time series. Performance evaluation on five NAB datasets [31] showed that the proposed approach performs well on the NAB Jumps periodic data set and outperforms the models it was compared to. Other types of non-regressive techniques are ML methods for time series

analysis. In [32] the Local Outlier Factor (LOF) method was employed to identify anomalous events in the marine domain and attained 83.4% precision. The Isolation Forest (ISOF) algorithm has been applied to streaming data in [33], achieving an AUC score of 0.98 in one of the test dataset. In [34] the One-Class Support Vector Machines (OC-SVM) has been implemented for the identification of network anomalies, and for the test set, the outliers identified perfectly match the human visual detection result.

Regressive approaches compute a model of the time series generation process. In the case of AD, an autoregression model is used to forecast the variable of interest from its past values. Autoregressive models include methods based on Autoregressive Moving Average (ARMA) [35] [36] [37] and on Neural Networks, such as Autoencoders (AE) [38] [39] and Recurrent Neural Networks (RNNs) [40] [41]. Forecasting-based AD approaches are divided into single-step and multi-step methods depending on the number of predicted points. The former strategy is preferable for short-term forecasting (i.e., minutes, hours, and days) and the latter for long-term data series analysis.

In the electric load analysis domain, the work in [42] studies the problem of time series forecasting for electric load measurements and shows that Long Short-Term Memory (LSTM), a deep learning model, outperforms AutoRegressive Integrated Moving Average (ARIMA), a statistical-based model, on three data sets obtained from the Open Power System Data on electric load in Great Britain, Poland, and Italy [18]. [43] shows the importance of an Fast Fourier Transform (FFT) based periodicity pre-processor to extract the period in smart grids time series. [44] proposes the use of Variational Autoencoders (VAE) for the unsupervised anomaly detection in solar energy generation time series and the results show that the trained model is able to detect anomalous patterns by using the probabilistic reconstruction metrics as anomaly scores. [45] surveys several Artificial Intelligence methods for anomaly detection in buildings' energy consumption, identifying several factors (e.g., occupancy and outdoor temperatures) that influence time series behavior.

In the specific field of periodic data series analysis, [46] employs a periodicity preprocessor to find the time series period and segment the data into windows. Then it exploits a combination of an RNN and a CNN to detect anomalies achieving an F_1 score near 0.9 on all the test datasets. [43] also uses a periodicity pre-processor, based on the Fourier transform, and maps multiple periods onto a single cycle to identify deviations across subsequent periods. [44] uses Bi-LSTM to detect anomalies and proposes the use of attention maps to explain the results. [47] encodes periodic time series using letters as a data size reduction technique. The classification process led to robust results with a global accuracy that ranged between 80% and 90%. These works show the advantages of pre-processing to exploit the data periodicity and of dimensionality reduction techniques and discuss results interpretability.

The proliferation of time series analysis methods and of AD specific approaches has spawned a stream of research focused on comparing the performance of alternative techniques. For example, the work in [42] compares the multi-step forecasting performance of ARIMA and LSTM-based RNN models and shows that the LSTM model outperforms the ARIMA model for multi-step electric load forecasting. Our

preliminary work [48] compares CNN-powered and RNN-powered AD methods with One-Class Support Vector Machines and Isolation Forest techniques on one quasi-periodic data set, using standard metrics (precision, recall, F_1 score). In this paper we deepen the analysis assessing performances under multiple metrics, investigating the impact of the training sub-sequence duration and of the analysis window size, and contrasting the generalization capacity of the reviewed approaches.

Experimental settings

Data set

The experiments exploit a fridge energy consumption data set collected using smart plugs. The energy consumption data have been collected in Greek residential households using the BlitzWolf BW-SHP2 smart plugs, which allow exporting the time series through an API. The data collection system, the assessed algorithms and the evaluation framework were all implemented in Python. The time series in the data set record the active power consumption of three fridges for over 2 months, with 1 minute data resolution. The time series have been divided into sub-sequences for training, validation, and testing of the methods. Table 1 summarizes the data split.

	Total sequence		Train sub-sequence		Val sub-sequence		Test sub-sequence	
	Start	End	Start	End	Start	End	Start	End
Fridge1	15/01/20	23/03/20	21/01/20	20/02/20	21/02/20	23/02/20	24/02/20	23/03/20
Fridge2	21/01/20	23/03/20	21/01/20	20/02/20	21/02/20	23/02/20	24/02/20	23/03/20
Fridge3	21/01/20	23/03/20	21/01/20	20/02/20	21/02/20	23/02/20	24/02/20	23/03/20

Table 1 The dataset collection period and the train-val and test split

When working in normal conditions, the energy consumption curve of a fridge displays a cyclic behavior alternating between a high consumption state (ON) and a low consumption stage (OFF). Figure 1 shows an example of the consumption data of one appliance.

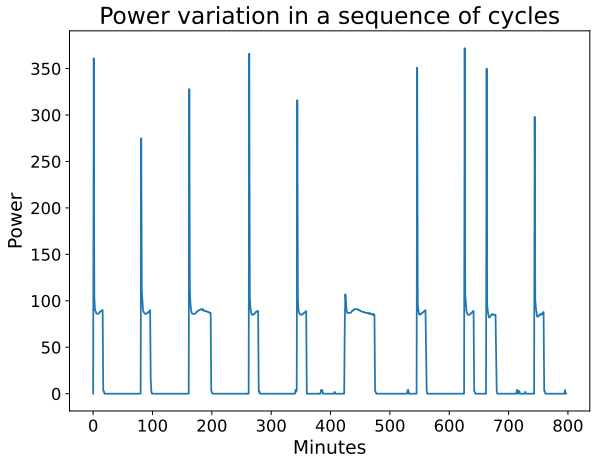

Figure 1 Example of the fridge energy consumption data series. The time series is formed by subsequent ON-OFF cycles and is quasi-periodical.

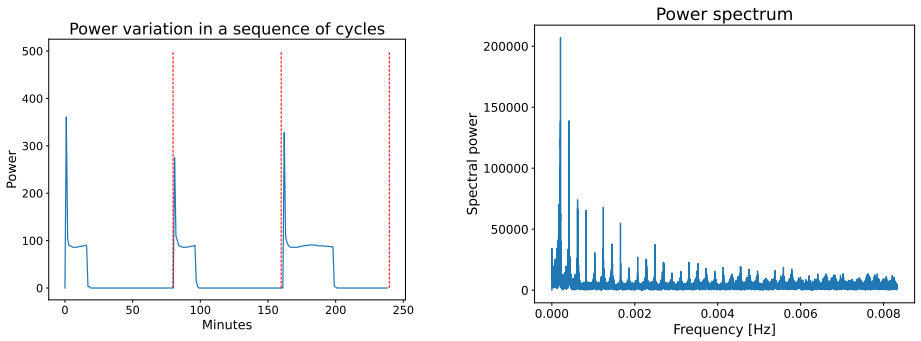

Figure 2 The power spectrum computed by the periodicity pre-processor (right) on the fridge energy consumption time series (left). The period detected for an ON-OFF cycle is about 80 minutes for the analyzed data set.

Data set analysis

Periodicity analysis Normal fridge consumption shows a cyclic behavior. Periodicity analysis aims at detecting the mean period corresponding to an ON-OFF cycle and possibly to other longer patterns (e.g., seasonal effects). It is a preliminary step before the application of AD and requires a non-anomalous sub-series, which can be created by manually removing anomalies from the training sub-sequence. The Fast Fourier Transform (FFT) is applied on the anomaly-free sub-sequence to map the data into the frequency domain and the periodicity is defined as the inverse of the frequency corresponding to the highest power in the FFT, as proposed in [30]. Table 2 summarizes the periodicity, expressed in minutes of the three data sets. The periods range from 45 minutes to 1h 40 minutes. No seasonal affect is found because the train set refers to only one month. Figure 2 shows the power spectrum computed for one of the three appliances.

	Fridge1	Fridge2	Fridge3
Period	100	80	45

Table 2 The periods determined for the energy consumption time series, expressed in minutes.

Ground Truth annotation For training and testing purposes, the energy consumption time series have been annotated with ground truth (GT) metadata to specify the points that deviate from normality. Three independent annotators have labeled the data points, with a Boolean tag (normal/anomalous) and with a categorical label denoting the type of the anomaly, with the interface shown in Figure 3.

Anomaly classes and their distribution The anomalies have been distinguished in the following categories: *Continuous OFF state*, when the appliance is in the low consumption state for a long time, *Continuous ON state*, when the appliance is in the consumption state for an abnormally long time, *Spike*, when the appliance has an abnormal consumption peak possibly preceded by a ramp and followed by a decay period, *Spike + Continuous*, when the appliance has a consumption peak followed by a prolonged ON state, *Other*, when the anomaly does not follow a well-defined pattern. Figure 4 shows the distribution of the anomaly categories in the

Figure 3 The interface of the GT anomaly annotator at work on the fridge time series. The user can specify the anomalies and add meta-data to them. The user has annotated the currently selected GT anomaly, shown in red, with the *Continuous ON state* label.

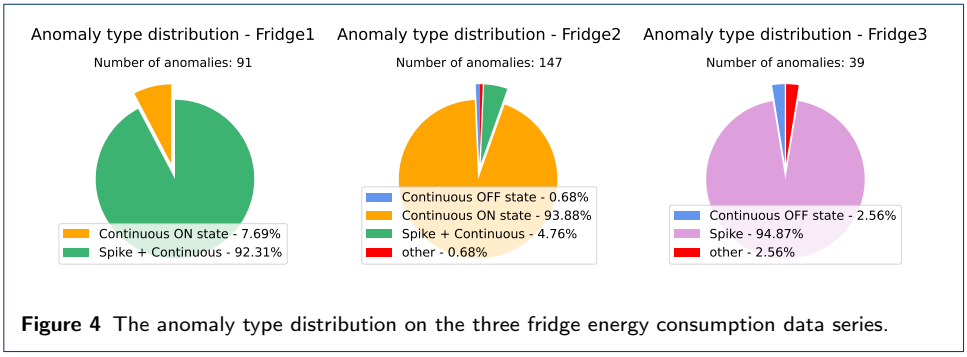

Figure 4 The anomaly type distribution on the three fridge energy consumption data series.

data set of the three fridges. The plots highlight the different anomalous behavior of the appliances. Fridge2 is mainly subject to continuous ON cycles. Fridge 1 shows a similar pattern, but the prolonged ON states are preceded by an abrupt increase in the consumption. Fridge3 is subject to a more detectable anomalous behavior because almost 95% of the anomalies are of spike type, which are easier to detect also visually.

GT anomaly duration distribution Figure 5 shows the GT anomaly duration distribution on the data series of the three fridges. The distributions of Fridge1 and Fridge2 are centered close the time series period, which suggests the presence of anomalies shorter than an ON-OFF cycle. The distribution of Fridge3 is centered around values higher than the mean ON-OFF cycle duration, which is typical of the transient behavior caused by high consumption spikes.

Compared algorithms

Algorithm list and definitions

The algorithm selection considered the most common methods used in the reviewed studies and their nature (statistical, regressive, neural) so as to achieve a balanced representation of the different approaches.

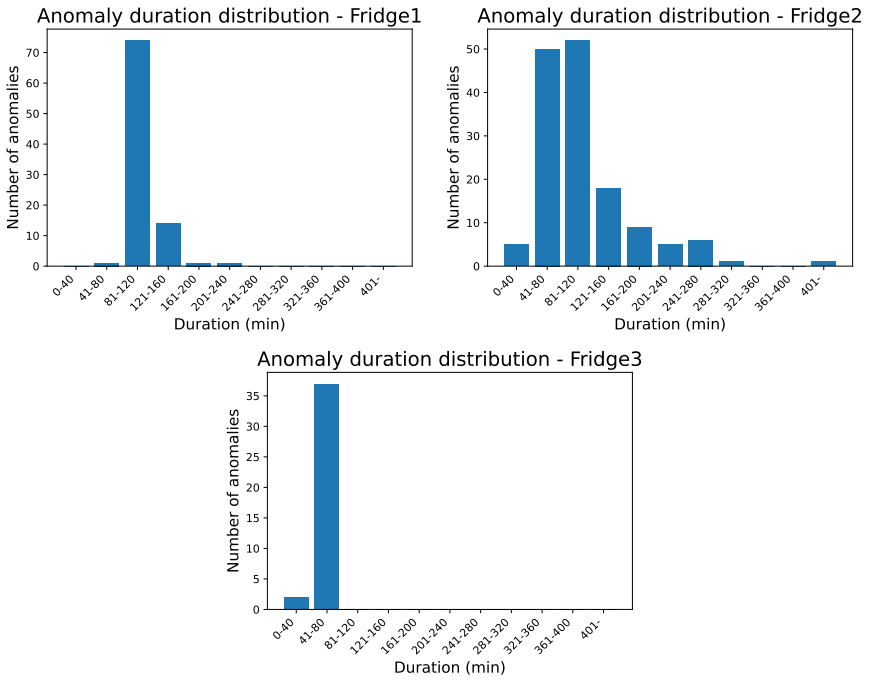

Figure 5 The anomaly duration distribution on the fridge energy consumption data sets. The distributions of Fridge1 and Fridge2 are centered close the time series period, which suggests the presence of anomalies shorter than an ON-OFF cycle whereas the distribution of Fridge3 is centered around values higher than the mean ON-OFF cycle duration.

1 **Basic Statistics** is an extension of the method presented in [30] for periodic series. The first step analyzes the anomaly-free training data series to determine the periodicity. Then, the anomaly-free train set is divided into non-overlapping windows of the same size as the period and the Pearson product-moment correlation coefficient is computed on all the pairs of contiguous windows to check whether the time series is periodic within the two windows. If it is periodic, the ratio $R_{std} = \frac{|Std_{current} - Std_{previous}|}{Std_{previous}}$ is computed. An anomaly occurs if R_{std} exceeds a threshold τ, defined as follows. R_{std} is calculated for each window pairs in the train set and the maximum value (R_{max}) allowed in a non-anomalous time series is found. Then the threshold τ is determined on the validation set by performing a grid search. Given a set of possible thresholds $\tau_\alpha = R_{max}(1 + \alpha)$, with α ranging from 0 to 10 with step 0.1, the threshold τ is defined as the value corresponding to the best F_1 score obtained by applying the anomaly definition rule on the validation set. Finally, the same rule is applied to the test set using the computed threshold value.

2 **AutoRegressive (AR)** [49] is an autoregression model exploiting past data to predict current data. The prediction model is defined as:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \tag{1}$$

where c, ϕ_i are the model parameters and ε_t is a white noise term. Anomalies are computed from the prediction error by thresholding.

3 **AutoRegressive Integrated Moving Average (ARIMA)** [42, 49] is a model exploiting past data, differencing of the original time series and a linear combination of white noise terms. A model ARIMA(p, d, q) is defined as:

$$y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{2}$$

where y'_t is the differenced time series, ε_t is a white noise term and c, ϕ_i, θ_j are the model parameters. Anomalous points are defined as in AR.

4 **Local Outlier Factor (LOF)** [50] is a clustering algorithm based on the identification of the nearest neighbors and of local outliers.

5 **One-Class SVM (OC SVM)** [51] is the use of support vector machine (SVM) for novelty detection.

6 **Isolation Forest (ISOF)** [52] is an ensemble method that creates different binary trees for isolating anomalous data points.

7 **Gated Recurrent Unit (GRU)** [53] is a class of Recurrent Neural Network (RNNs) that exploit update gate and reset gate to decide what information should be passed to the output.

8 **Gated Recurrent Unit multisteps (GRU-MS)** is based on GRU and is used to predict multiple consecutive data points in the future.

9 **Long Short-Term Memory (LSTM)** [54] is another class of RNNs exploiting a cell with an input gate, an output gate and a forget gate. Both GRU and LSTM are designed to take advantage of the past context of the data and to avoid the gradient vanishing problem of RNNs.

10 **Long Short-Term Memory multisteps (LSTM-MS)** is based on LSTM and is used to forecast several consecutive data points.

11 **GRU-Autoencoder (GRU-AE)** [55] is a hybrid model using an autoencoder and a GRU network.

12 **LSTM-Autoencoder (LSTM-AE)** [56] is another hybrid model coupling an autoencoder and an LSTM network.

Training procedure and parameter settings

The hyperparameters of the ISOF, OC SVM, LOF, and ARIMA models are set with Bayesian search employing the hold-out set method. For each configuration, the chosen hyperparameters are used to fit the model and the performances are evaluated on the validation set. LOF, OC SVM and ISOF are assessed using the maximum F_1-score whereas the ARIMA models using the mean squared error (MSE) on predictions. The hyperparameters yielding the maximum F_1 or the lowest MSE are selected.

ARIMA is trained on anomaly-free data to learn normal patterns as done in [57].

ISOF, LOF and OC SVM work on spatial data and thus the univariate time series is projected onto a space $\mathbb{R}^n$ with $n \geq 1$ [22, 58]. A window of size n is used to extract from the time series $N - n + 1$ vectors of length n of consecutive points,

where N is the length of the time series. Then, the spatial algorithms are trained on the projected vectors. At test time, the test set is projected onto $\mathbb{R}^n$ and the score of each projected vector is computed. The anomaly score of a point in the time series is defined as the average of all the anomaly scores of the vectors that contain the point. For all the neural models, training is performed on anomaly-free data.

Table 3 summarizes the relevant features and parameters of the compared methods.

Algorithm	**Configuration parameters**
Basic Statistics	Pearson product-momentum correlation coefficient minimum value (0.2)
AR	p order in [25, 305]
ARIMA	p order in [25, 305], d = 1, q = 0
LOF	number of neighbours in [1, 300]
OC SVM	gamma in [0.001, 0.9726], tol in $[10^{-10}, 0.1]$, nu in [0.001, 0.5]
ISOF	number of trees in [20, 200], max samples in [150, 400]
GRU	2 GRU layers both with 32 units. Training: 500 epochs with patience = 30 and batch size = 64.
LSTM	2 LSTM layers both with 32 units. Training: 500 epochs with patience = 30 and batch size = 64.
GRU-MS	2 GRU layers with 64 and 32 units, and 10 units for the output layer. Training: 500 epochs with patience = 30 a batch size = 64.
LSTM-MS	2 LSTM layers with 64 and 32 units, and 10 units for the output layer. Training: 500 epochs with patience = 30 and batch size = 64.
GRU-AE	2 GRU layers with 128 and 64 units. Training: 500 epochs with patience = 30 and batch size = 64.
LSTM-AE	2 LSTM layers with 128 and 64 units. Training: 500 epochs with patience = 30 and batch size = 64.

Table 3 Relevant configuration parameters of the compared methods.

Anomaly definition, GT matching, and performance metrics

Anomaly definition strategies. An anomaly definition strategy specifies how the output of the anomaly detector and the data points of the time series are compared in order to identify whether a point is anomalous. AD algorithms adopt different strategies to identify abnormal points:

- Confidence: an anomaly score is directly provided as output by the model.
- Absolute and Squared Error [59]: the anomaly score is defined as the absolute or squared error between the input and the predicted/reconstructed value.
- Likelihood [41]: each point in the time series is predicted/reconstructed l times and associated with multiple error values. The probability distribution of the errors made by predicting on normal data is used to compute the likelihood of normal behavior on the test data, which is used to derive an anomaly score.
- Mahalanobis [60]: each point in the time series is predicted/reconstructed l times. For each point, the anomaly score is calculated as the square of the Mahalanobis distance between the error vector and the Gaussian distribution fitted from the error vectors computed during validation.
- Windows strategy [61]: a score vector of dimension l is associated with each point. Each element s_i of the score vector is the mean absolute or mean squared error of the i-th predicted/reconstructed window that contains the point.

A threshold τ is then applied to the calculated score(s) for classifying the point as normal or anomalous. Table 4 shows the anomaly definition strategies of the compared methods.

Anomaly detection criteria and thresholds. The criteria are the ones adopted in order to identify an anomaly. They are strongly related to the nature of the used algorithm. The anomaly identification criteria used by the compared methods are classified in:

- *Prediction error*: prediction models identify anomalies based on the difference between the predicted value and the observed one. Anomalies are identified based on the residuals between the input and the generated data: the higher the difference, the higher the likelihood of an anomaly.
- *Reconstruction error*: this criterion applies to all the models that aim at generating an output as close as possible to the input, such as the autoencoder-based models. As for the prediction models, the larger the residual, the higher the probability of an anomaly.
- *Dissimilarity*: dissimilarity models classify anomalous points by comparing them with the features or with the distribution of normal points or by matching them with the clusters computed from the normal time series.

Table 4 summarizes the detection criteria used by the different algorithms.

Algorithm	Anomaly detection criterion	Anomaly definition strategy
Basic Statistics	Dissimilarity	Confidence
AR	Prediction error	Absolute Error
ARIMA	Prediction error	Absolute Error
LOF	Dissimilarity	Confidence
OC SVM	Dissimilarity	Confidence
ISOF	Dissimilarity	Confidence
GRU	Prediction error	Absolute Error
LSTM	Prediction error	Absolute Error
GRU-MS	Prediction error	Likelihood
LSTM-MS	Prediction error	Likelihood
GRU-AE	Reconstruction error	Windows strategy
LSTM-AE	Reconstruction error	Windows strategy

Table 4 Anomaly detection criteria and definition strategies adopted for each algorithm.

GT matching. To evaluate the predictions as true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), a *Point to Point* matching strategy has been adopted: each anomalous point is compared only to the corresponding one in the input data series using the GT label.

Performance metrics The evaluation adopts the most widely used machine learning metrics, *precision*, *recall*, and F_1 *score*, defined as follow:

$$precision = \frac{TP}{TP + FP} \ , \ recall = \frac{TP}{TP + FN} \ , \ F_1 score = 2 * \frac{precision * recall}{precision + recall} \quad (3)$$

Experimental results

In this section we summarize the responses to the four questions introduced in the Introduction. For space reasons we condense the results of the 144 (12 methods $\times$ 3 training periods $\times$ 4 window sizes) experiments on 3 data sets and discuss only the essential findings. The complete list of results is published at the address: https://github.com/herrera-sergio/AD-periodic-TS.

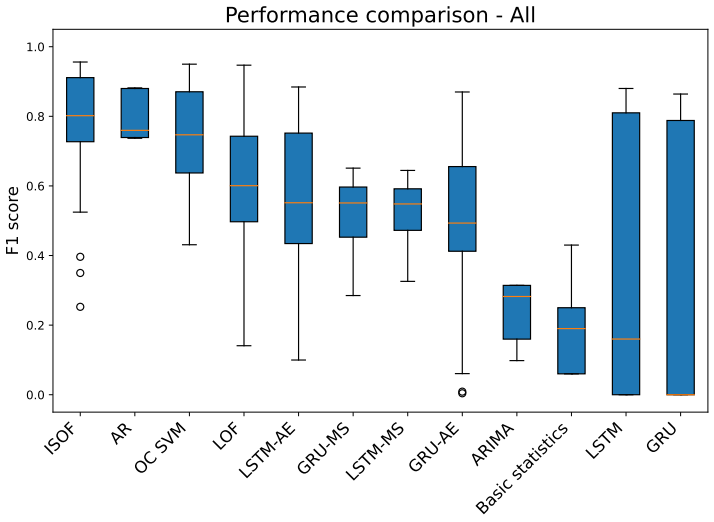

Figure 6 Comparison of the performances of all the algorithms on all the appliances and across all the training duration periods and window sizes. The methods are ordered in descending order of the median values of the F_1 score.

Q1: comparative performances

Figure 6 shows the comparison of the methods over all the data sets and across all the training duration values and sizes of the sliding window. The ISOF method consistently achieves the best F_1 score, followed by OC SVM and LOF. The AE and MS neural methods have comparable performances. The multi-step approaches exhibit a more consistent behavior yielding smaller values of the standard deviation and the GRU-AE method performs slightly worse than the other approaches. The neural methods that predict only one point in the future (LSTM and GRU) have low performance and a rather inconsistent behavior. This is expected due to the high sampling frequency, which makes one step prediction ineffective to detect anomalies. Of the remaining non-neural methods, ARIMA and Basic Statistic are positioned at the low end of the performance range.

The top result on all the experiments is attained by ISOF on the Fridge3 time series, trained with a sub-sequence of length equal to one month and with a window size of 2 $\times$ period: Precision = 0.947, Recall = 0.965, F_1 score = 0.956.

A special case is that of AR. The training of the method converges only for the shortest duration of the training sub-sequence (a half period). However, the trained model delivers on average a good F_1 score. It can be observed that AR grossly fails in the accuracy of the predicted values but nonetheless the error of the points that belong to a normal sub-sequence is very different from the error of the points that lie within an anomalous sub-sequence, which results in good AD performances.

Figure 7 shows the performance break down by appliance. As expected all methods, but ARIMA and Basic Statistics, perform better on the Fridge3 data set, which contains more recognizable anomalies mostly of a single type ($\approx 95\%$ of type spike). On the Fridge1 and Fridge2 data sets the performances follow the same ranking as in Figure 6, with the same top-4 methods (ISOF, AR, OC SVM and LOF) and almost equivalent performances of the MS and AE methods. On the Fridge3 data set the

methods that predict one step in the future (LSTM and GRU) work better. This analysis highlights that the performances of the models are affected by the considered appliance. Indeed, in Fridge1 the performances are more subject to variations, while in Fridge3 are more consistent. Moreover, ARIMA and Basic Statistics show low performances independently on the complexity of the dataset, which suggests their inadequacy for this kind of problem.

The results are in line with those of the work of Kharitonov et al.[10] in which the authors compare the performances of alternative techniques to detect failures using manufacturing machine logs and observed that k-nearest neighbors (KNN) and LOF performed better, while autoencoders could not be considered for deployment in a real-case scenario. Similarly, Elmrabit et al.[62] found that classical machine learning techniques outperformed deep learning for the AD task in cybersecurity datasets.

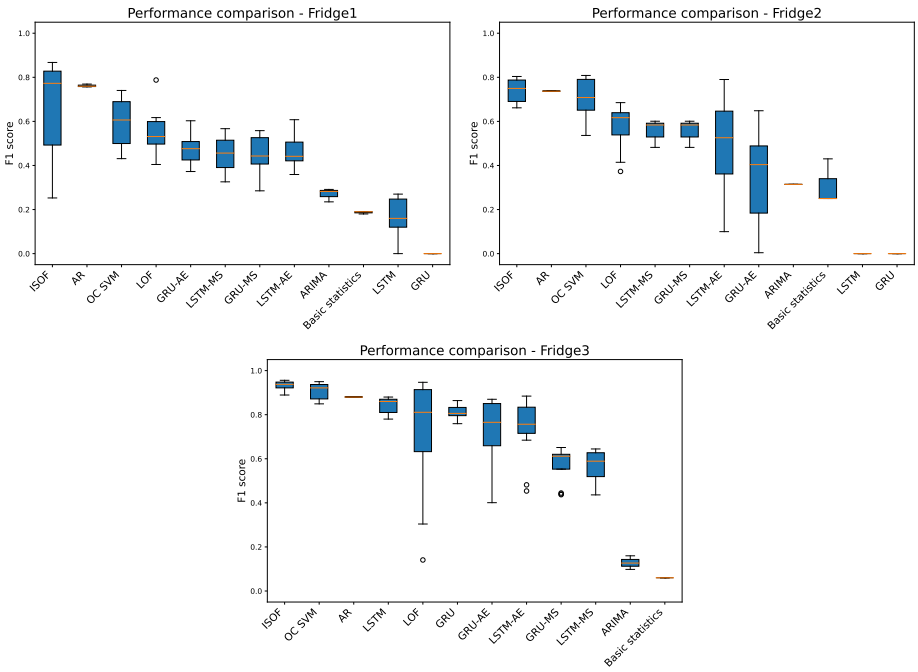

Figure 7 Break down of the performance of all the algorithms by appliance. The methods are ordered by descending median value of the F_1 score.

Q2: Training sub-sequence duration

Figure 8 shows the variation of the F_1 metrics for the 10 methods that could be trained with all the three sub-sequences (2 weeks, 3 weeks, one month). The results show that the 2 weeks training period is sufficient for most of the methods. Only the multisteps (MS) methods attain a very slight average performance improvement if the training period length extends to 1 month. The results on the time series of Fridge1 and Fridge2 show a similar trend. All the detailed results can be found in the mentioned project repository.

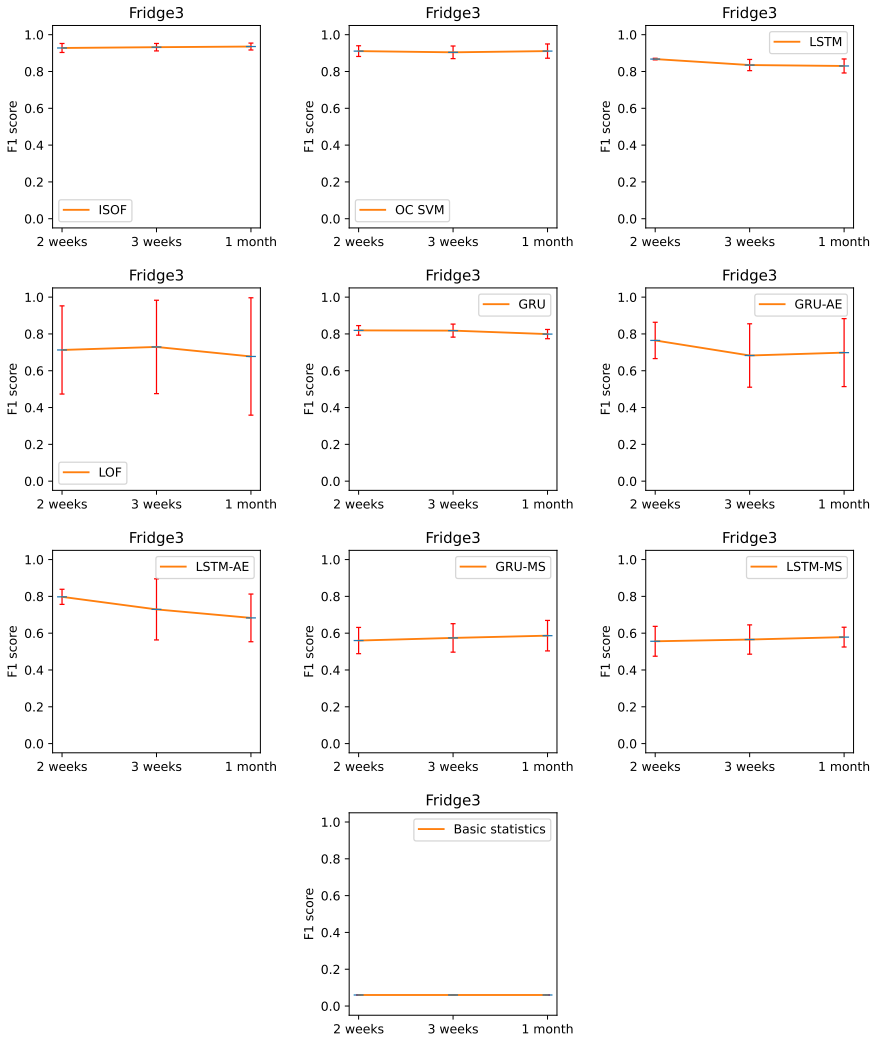

Figure 8 Variation of the F_1 score with the duration of the training sub-sequence. The AR and ARIMA method did not complete the training with all the periods.

Q3: Window length

Figure 9 shows the variation of the F_1 metrics with the sliding window size (half a period, one period, two and three periods), limited to the 9 methods that could be trained completely. The results show a difference in the pattern between neural and non-neural methods.

With ISOF and OC SVM the F_1 score decreases when the window size increases. With a value greater than half a period the methods progressively loose effectiveness: the variance increases and the F_1 score decreases. This is likely the effect of the worse trade-off between the noise and the context knowledge enclosed in the window.

The AE methods deliver the best F_1 score when the window size equals twice the duration of the period. A similar trend is also displayed by MS methods, with LSTM-MS showing a slight monotonic increase up to the three periods. The one step neural methods GRU and LSTM are rather insensitive to the window size, but

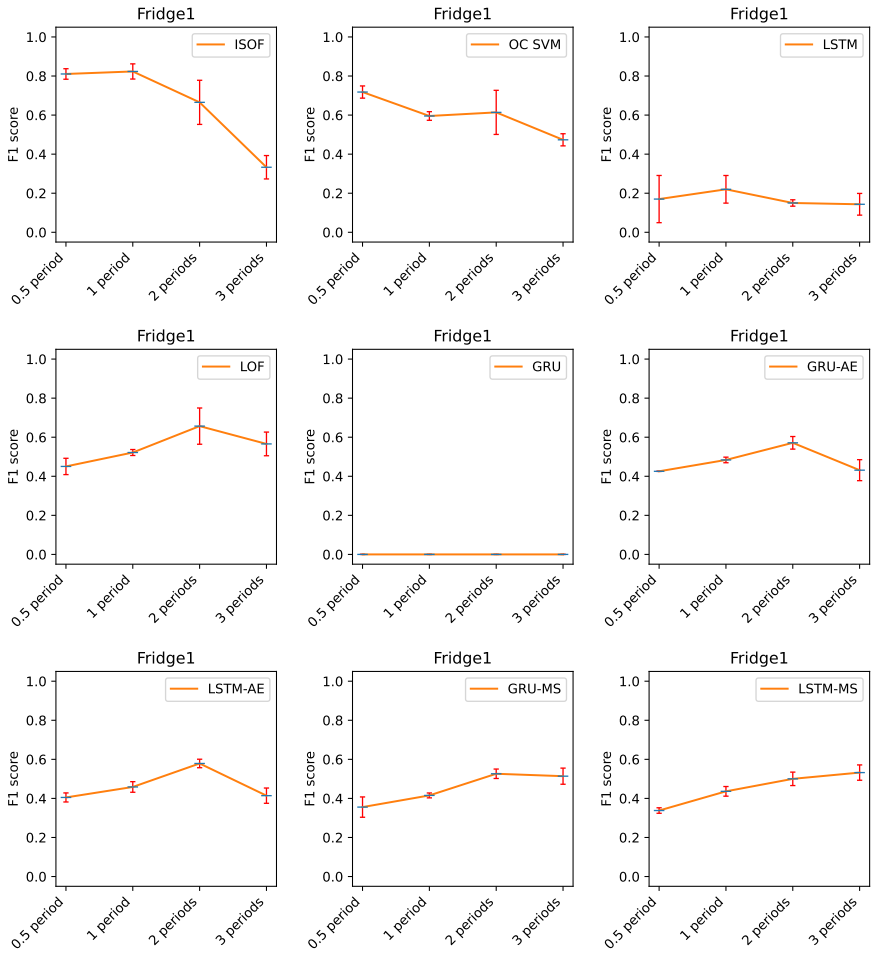

Figure 9 Variation of the F_1 score with the size (in periods) of the sliding window. The AR and ARIMA method did not complete the training with all the periods.

their performance is at the lower end of the range. The LOF approach exhibit the same trend as the AE and MS neural methods.

The value at the (2 × period) point of the neural methods shows that such a duration gives sufficient context for encoding the periodic features of the time series well and that going beyond that size is either counterproductive or yields a modest benefit. In the AE methods, the negative effect of the window size extension may be also due to the dimensionality reduction to a latent space operated by the neural architecture, which may become less effective when the dimension of the original space gets too large.

The results on the time series of Fridge2 and Fridge3 show a similar trend. All the detailed results can be found in the mentioned project repository.

Q4: Generalization

The generalization experiments assess the top-5 methods (ISOF, OC SVM, LOF LSTM-AE and GRU-AE) on a dataset different from the one on which the methods have been originally trained. Each method is tested in two variants: the original

version trained on the first appliance and a version in which the threshold value is fine-tuned on the validation data series of the target appliance.

Figure 10 contrasts the F_1 scores obtained by the baseline version of the algorithm, i.e., the one trained and tested on the same dataset, the F_1 scores achieved by fine tuning the threshold on the validation set of the target appliance, and the F_1 scores obtained without any fine tuning. The top performing method (ISOF) is also the one that generalizes best, even without fine tuning the threshold. In general, ISOF and OC SVM are less dependent on the training set with respect to the neural models, which have a sensible performance decay when tested on a different appliance. The degradation is more sensible when the test appliances is Fridge3, which has almost all anomalies of type spike, which are absent in Fridge1 and Fridge2.

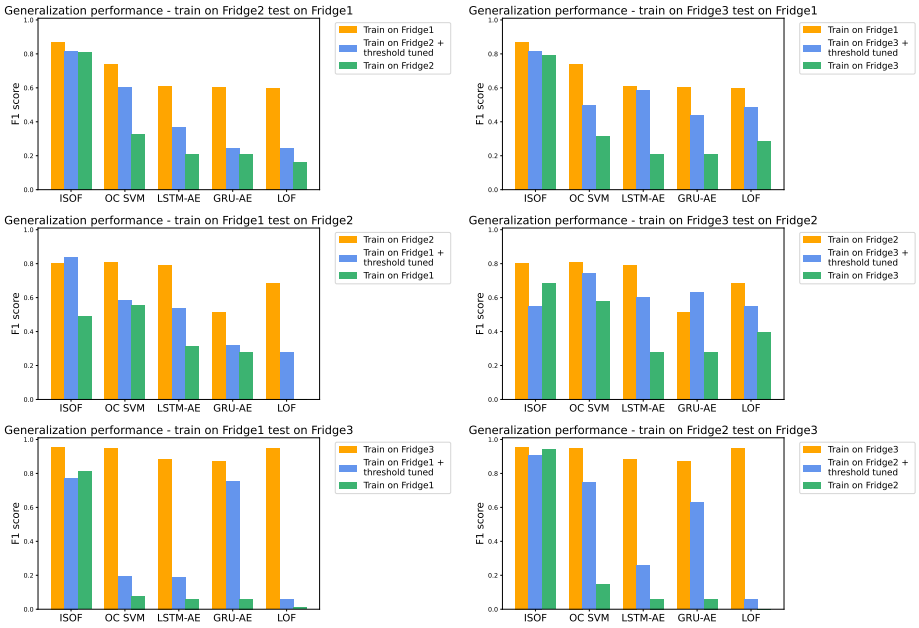

Figure 10 Comparison of the generalization performance of the top-5 methods. The orange bar represents the baseline F_1 score (i.e., training and testing done on the same dataset), the blue bar denotes the F_1 score achieved by fine tuning the threshold on the validation set of the target appliance, and the green bar shows the performances obtained using the trained algorithm without fine tuning.

Qualitative analysis of results

To get a qualitative appreciation of the different behavior of the best models, Figure 11 directly compares the anomalies detected by ISOF, OC SVM and LSTM-AE with the GT anomalies. The detected anomalies are highlighted with a color that depends on the method and the GT anomalies are circled in red.

The plot on the left column show a situation in which all the three methods are able to detect more or less the same anomalous data points. The detected points match well the GT annotations. The plots on the right column show how the methods react to a change of the duration of the ON-OFF cycle (an acceleration in the displayed example, which may be caused by a different load of the fridge or by a change in the set point of the thermostat). Only the ISOF method is robust to such

an occurrence. The other methods instead signal many normal points as anomalous, because they consider the entire cycle variation as an anomaly. Given that the time series of the appliances are quasi-periodic, as shown in the power spectrum of Figure 2, the robustness with respect to small variations of the ON-OFF cycle is a very relevant benefit of the ISOF method.

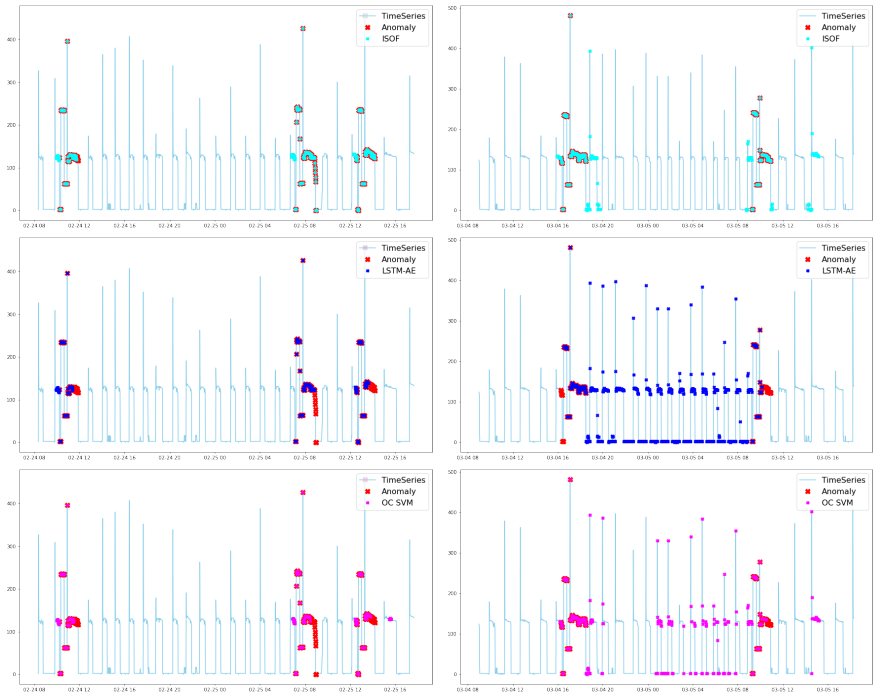

Figure 11 Qualitative analysis of the predictions of three methods on Fridge1: ISOF, LSTM-AE, OC SVM. ISOF (top) is more robust to the variations of the duration of the cycles, while the others show a weakness in the identification of the anomalous points, in fact, LSTM-AE (middle) and OC SVM (bottom) label numerous normal points as anomalous.

Conclusions

In this paper we have discussed the results of the experimental comparison of 12 AD methods on three quasi-periodic data series collected with smart plugs connected to three distinct fridges. The comparison has first assessed the prediction performances, measured with the F_1 score metrics, which confirmed that the non-neural machine learning methods ISOF, OC SVM and LOF attain the best results, followed by the autoencoder-based and multi-step neural methods (GRU-AE, GRU-MS, LSTM-AE, LSTM-MS). In particular, the ISOF method trained with a sub-sequence of length equal to one month and with a window size of 2 $\times$ period attained a very good result on a fridge data series containing mostly spike anomalies (Precision = 0.947, Recall = 0.965, F_1 score = 0.956).

Next we evaluated the impact of the duration of the sub-sequence used for training the algorithms, which shows that the 2 weeks training period is sufficient for most of the methods and that the AR and ARIMA algorithms did not complete the training within reasonable time with time series of longer duration.

The impact of the sliding window size was also investigated. Non-neural machine learning algorithms require a shorter window (half of the period is enough), whereas neural models deliver the best performance with a larger window size (two periods in most cases).

Finally, the generalization ability of the top performing methods has been assessed too. The best method (ISOF) is also the one that preserves its performances intact when applied to a different appliance, even without fine-tuning the threshold on the target appliance.

Future work will further pursue the investigation of AD algorithms on quasi-periodic data series, focusing also on their runtime performance on hardware with memory and processing constraints. The objective is designing a timely, accurate and efficient system for dispatching mobile phone alerts about the potential malfunctioning of home appliances to real-world users.

LIST OF ABBREVIATIONS

AD: Anomaly Detection
AE: Autoencoders
AR: Autoregressive
ARIMA: Autoregressive Integrated Moving Average
ARMA: Autoregressive Moving Average
Bi-LSTM: Bidirectional Long Short-Term Memory
CNN: Convolutional Neural Network
ECG: Electrocardiography
FFT: Fast Fourier Transform
FN: False Negative
FP: False Positive
GRU: Gated Recurrent Unit
GRU-AE: Gated Recurrent Unit Autoencoder
GRU-MS: Gated Recurrent Unit multisteps
GT: Ground Truth
ISOF: Isolation Forest
KNN: K-Nearest Neighbors
LOF: Local Outlier Factor
LSTM: Long Short-Term Memory
LSTM-AE: Long Short-Term Memory Autoencoder
LSTM-MS: Long Short-Term Memory multisteps
MAE: Mean Absolute Error
MS: Multisteps
MSE: Mean Squared Error
NILM: Non Intrusive Load Monitoring
NN: Neural Networks
OC SVM: One-Class Support Vector Machine
RNNs: Recurrent Neural Networks
SE: Squared Error
SVM: Support Vector Machine
TN: True Negative
TP: True Positive
VAE: Variational Autoencoders

DECLARATIONS

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Availability of data and materials
All the material relative to this article is publicly available in the following repository https://github.com/herrera-sergio/AD-periodic-TS. The dataset used for the study are private and permission for publication was not granted, it will be included in the repository if permission is granted in the future.

Competing interests
The authors declare that they have no competing interests.

Funding
This paper is part of the funded project PRECEPT (No.958284) by the funding agency European Union's Horizon 2020 Framework.

Authors' contributions
NZ analyzed the dataset and prepared the split of the data set for training/testing; led the implementation of the algorithms and the evaluation of the models. PF designed the research and the experimentation procedure; analyzed the results and made a major contribution to the writing of the manuscript. MP implemented the regressive algorithms, performed the training of the algorithms and the evaluation. NOPV implemented procedure for the identification of the period on the data sets, implemented the statistical algorithm, performed the training of the algorithm and the evaluation. SLHG contributed to the analysis of the data and design of the experiments; collaborated with the training of the algorithms and prepared the first draft of the document. All authors read and approved the final manuscript.

Acknowledgements
This work has been supported by the European Union's Horizon 2020 project PRECEPT, under grant agreement No. 958284.

References
1. Shah AS, Nasir H, Fayaz M, Lajis A, Shah A. A review on energy consumption optimization techniques in IoT based smart building environments. Information. 2019;10(3):108.
2. Shaikh PH, Nor NBM, Nallagownden P, Elamvazuthi I, Ibrahim T. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. Renewable and Sustainable Energy Reviews. 2014;34:409–429.
3. Azizi E, Beheshti MTH, Bolouki S. Appliance-Level Anomaly Detection in Nonintrusive Load Monitoring via Power Consumption-Based Feature Analysis. IEEE Trans Consumer Electron. 2021;67(4):363–371. Available from: https://doi.org/10.1109/TCE.2021.3129356.
4. Amasyali K, El-Gohary NM. A review of data-driven building energy consumption prediction studies. Renewable and Sustainable Energy Reviews. 2018;81:1192–1205.
5. Fan C, Xiao F, Zhao Y, Wang J. Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. Applied energy. 2018;211:1123–1135.
6. Cheng JC, Chen W, Chen K, Wang Q. Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. Automation in Construction. 2020;112:103087.
7. Schlegl T, Seeböck P, Waldstein SM, Langs G, Schmidt-Erfurth U. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. Medical image analysis. 2019;54:30–44.
8. Chauhan S, Vig L. Anomaly detection in ECG time signals via deep long short-term memory networks. In: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE; 2015. p. 1–7.
9. Sanz B, Santos I, Ugarte-Pedrero X, Laorden C, Nieves J, Bringas PG. Anomaly detection using string analysis for android malware detection. In: International Joint Conference SOCO'13-CISIS'13-ICEUTE'13. Springer; 2014. p. 469–478.
10. Kharitonov A, Nahhas A, Pohl M, Turowski K. Comparative analysis of machine learning models for anomaly detection in manufacturing. Procedia Computer Science. 2022;200:1288–1297.
11. Mishra M, Nayak J, Naik B, Abraham A. Deep learning in electrical utility industry: A comprehensive review of a decade of research. Engineering Applications of Artificial Intelligence. 2020;96:104000.
12. Seyoum S, Alfonso L, Van Andel SJ, Koole W, Groenewegen A, Van De Giesen N. A Shazam-like household water leakage detection method. Procedia Engineering. 2017;186:452–459.
13. Muniz Do Nascimento W, Gomes-Jr L. Enabling low-cost automatic water leakage detection: a semi-supervised, autoML-based approach. Urban Water Journal. 2022;p. 1–11.
14. Koukaras P, Bezas N, Gkaidatzis P, Ioannidis D, Tzovaras D, Tjortjis C. Introducing a novel approach in one-step ahead energy load forecasting. Sustainable Computing: Informatics and Systems. 2021;32:100616.
15. An interdisciplinary approach on efficient virtual microgrid to virtual microgrid energy balancing incorporating data preprocessing techniques. Computing. 2021;p. 1–42.
16. Liu F, Zhou X, Cao J, Wang Z, Wang T, Wang H, et al. Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional LSTM-CNN. IEEE Transactions on Knowledge and Data Engineering. 2020;.
17. Firth S, Kane T, Dimitriou V, Hassan T, Fouchal F, Coleman M, et al. REFIT Smart Home dataset. 2017 6;Available from: https://repository.lboro.ac.uk/articles/dataset/REFIT_Smart_Home_dataset/2070091.
18. A platform for Open Data of the European power system.;. https://open-power-system-data.org/. Accessed 3 June 2022.
19. Makonin S, Ellert B, Bajić IV, Popowich F. Electricity, water, and natural gas consumption of a residential house in Canada from 2012 to 2014. Scientific data. 2016;3(1):1–12.
20. Shakibaei P. Data-Driven Anomaly Detection From Residential Smart Meter Data; 2020.
21. Rashid H, Batra N, Singh P. Rimor: Towards identifying anomalous appliances in buildings. In: Proceedings of the 5th Conference on Systems for Built Environments; 2018. p. 33–42.
22. Braei M, Wagner S. Anomaly detection in univariate time-series: A survey on the state-of-the-art. arXiv preprint arXiv:200400433. 2020;.
23. Su Y, Zhao Y, Niu C, Liu R, Sun W, Pei D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining; 2019. p. 2828–2837.
24. Li D, Chen D, Goh J, Ng Sk. Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:180904758. 2018;.
25. Blázquez-García A, Conde A, Mori U, Lozano JA. A review on outlier/anomaly detection in time series data. ACM Computing Surveys (CSUR). 2021;54(3):1–33.

26. Pena D, Poncela P. Dimension reduction in multivariate time series. In: Advances in distribution theory, order statistics, and inference. Springer; 2006. p. 433–458.
27. Cook AA, Mısırlı G, Fan Z. Anomaly detection for IoT time-series data: A survey. IEEE Internet of Things Journal. 2019;7(7):6481–6494.
28. Box GE, Tiao GC. A canonical analysis of multiple time series. Biometrika. 1977;64(2):355–365.
29. Pena D, Box GE. Identifying a simplifying structure in time series. Journal of the American statistical Association. 1987;82(399):836–843.
30. Kao JB, Jiang JR. Anomaly detection for univariate time series with statistics and deep learning. In: 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE). IEEE; 2019. p. 404–407.
31. Ahmad S, Lavin A, Purdy S, Agha Z. Unsupervised real-time anomaly detection for streaming data. Neurocomputing. 2017;262:134–147.
32. Oehmcke S, Zielinski O, Kramer O. Event detection in marine time series data. In: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). Springer; 2015. p. 279–286.
33. Ding Z, Fei M. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. IFAC Proceedings Volumes. 2013;46(20):12–17.
34. Zhang R, Zhang S, Lan Y, Jiang J. Network anomaly detection using one class support vector machine. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. vol. 1. Citeseer; 2008. .
35. Pincombe B. Anomaly detection in time series of graphs using arma processes. Asor Bulletin. 2005;24(4):2.
36. Kadri F, Harrou F, Chaabane S, Sun Y, Tahon C. Seasonal ARMA-based SPC charts for anomaly detection: Application to emergency department systems. Neurocomputing. 2016;173:2102–2114.
37. Kozitsin V, Katser I, Lakontsev D. Online forecasting and anomaly detection based on the ARIMA model. Applied Sciences. 2021;11(7):3194.
38. Yin C, Zhang S, Wang J, Xiong NN. Anomaly detection based on convolutional recurrent autoencoder for IoT time series. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2020;52(1):112–122.
39. Li L, Yan J, Wang H, Jin Y. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. IEEE transactions on neural networks and learning systems. 2020;32(3):1177–1191.
40. Canizo M, Triguero I, Conde A, Onieva E. Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study. Neurocomputing. 2019;363:246–260.
41. Malhotra P, Vig L, Shroff G, Agarwal P, et al. Long short term memory networks for anomaly detection in time series. In: Proceedings. vol. 89; 2015. p. 89–94.
42. Masum S, Liu Y, Chiverton J. Multi-step time series forecasting of electric load using machine learning models. In: International conference on artificial intelligence and soft computing. Springer; 2018. p. 148–159.
43. Zhang L, Shen X, Zhang F, Ren M, Ge B, Li B. Anomaly detection for power grid based on time series model. In: 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). IEEE; 2019. p. 188–192.
44. Pereira J, Silveira M. Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA). IEEE; 2018. p. 1275–1282.
45. Himeur Y, Ghanem K, Alsalemi A, Bensaali F, Amira A. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. Applied Energy. 2021;287:116601.
46. Zhang S, Chen X, Chen J, Jiang Q, Huang H. Anomaly detection of periodic multivariate time series under high acquisition frequency scene in IoT. In: 2020 International Conference on Data Mining Workshops (ICDMW). IEEE; 2020. p. 543–552.
47. Capozzoli A, Piscitelli MS, Brandi S, Grassi D, Chicco G. Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings. Energy. 2018;157:336–352.
48. Zangrando N, Herrera S, Koukaras P, Dimara A, Fraternali P, Krinidis S, et al. Anomaly Detection in Small-Scale Industrial and Household Appliances. In: Maglogiannis I, Iliadis L, Macintyre J, Cortez P, editors. Artificial Intelligence Applications and Innovations. AIAI 2022 IFIP WG 12.5 International Workshops - MHDW 2022, 5G-PINE 2022, AIBMG 2022, ML@HC 2022, and AIBEI 2022, Hersonissos, Crete, Greece, June 17-20, 2022, Proceedings. vol. 652 of IFIP Advances in Information and Communication Technology. Springer; 2022. p. 229–240. Available from: https://doi.org/10.1007/978-3-031-08341-9_19.
49. Hyndman RJ, Athanasopoulos G. Forecasting: principles and practice, 3rd edition. OTexts; 2021.
50. Breunig MM, Kriegel HP, Ng RT, Sander J. LOF: Identifying Density-Based Local Outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD '00. New York, NY, USA: Association for Computing Machinery; 2000. p. 93–104. Available from: https://doi.org/10.1145/342009.335388.
51. Schölkopf B, Williamson RC, Smola A, Shawe-Taylor J, Platt J. Support Vector Method for Novelty Detection. In: Solla S, Leen T, Müller K, editors. Advances in Neural Information Processing Systems. vol. 12. MIT Press; 1999. Available from: https://proceedings.neurips.cc/paper/1999/file/8725fb777f25776ffa9076e44fcfd776-Paper.pdf.
52. Liu FT, Ting KM, Zhou ZH. Isolation Forest. In: 2008 Eighth IEEE International Conference on Data Mining; 2008. p. 413–422.
53. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:14123555. 2014;.
54. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.
55. Zhang C, Patras P, Haddadi H. Deep learning in mobile and wireless networking: A survey. IEEE Communications surveys & tutorials. 2019;21(3):2224–2287.
56. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078. 2014;.
57. Yaacob AH, Tan IKT, Chien SF, Tan HK. ARIMA Based Network Anomaly Detection. In: 2010 Second International Conference on Communication Software and Networks; 2010. p. 205–209.

58. Oehmcke S, Zielinski O, Kramer O. Event Detection in Marine Time Series Data. In: Hölldobler S, Peñaloza R, Rudolph S, editors. KI 2015: Advances in Artificial Intelligence. Cham: Springer International Publishing; 2015. p. 279–286.
59. Munir M, Siddiqui SA, Dengel A, Ahmed S. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. Ieee Access. 2018;7:1991–2005.
60. Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:160700148. 2016;.
61. Keras. Keras documentation: Timeseries Anomaly detection using an autoencoder;. https://keras.io/examples/timeseries/timeseries_anomaly_detection/. Accessed 3 June 2022.
62. Elmrabit N, Zhou F, Li F, Zhou H. Evaluation of Machine Learning Algorithms for Anomaly Detection. In: 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security); 2020. p. 1–8.