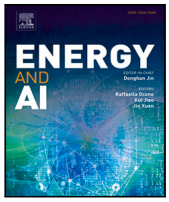
Energy and AI 15 (2024) 100324

Contents lists available at ScienceDirect

Energy and AI

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



Identifying the validity domain of machine learning models in building energy systems

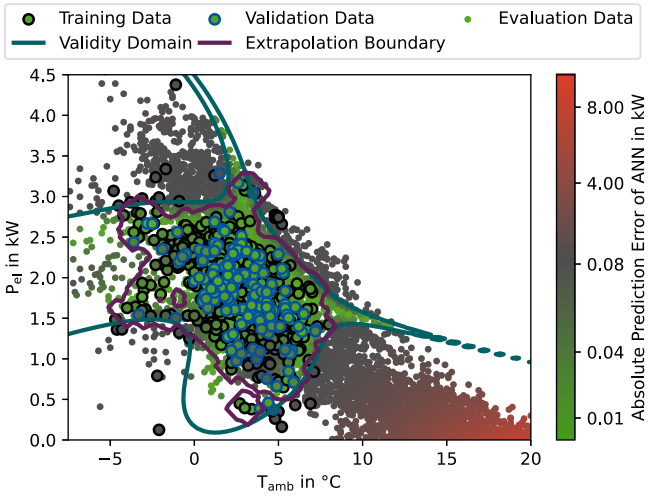
Martin Rätz *, Patrick Henkel, Phillip Stoffel, Rita Streblow, Dirk Müller

RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Mathieustraße 10, Aachen, 52074, Germany

HIGHLIGHTS

- Build knowledge about the validity domain of machine learning models.
- Calibrating novelty detection algorithms towards the model's validity domain.
- Sharpened definition for the validity domain and the extrapolation boundary.
- Visualization scheme for two-dimensional extrapolation detection tasks.
- Assessment of 15 novelty detection algorithms.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Extrapolation detection
Validity domain
Novelty detection
Machine learning
Artificial neural network
Data-driven model predictive control
Building energy systems

ABSTRACT

The building sector significantly contributes to climate change. To improve its carbon footprint, applications like model predictive control and predictive maintenance rely on system models. However, the high modeling effort hinders practical application. Machine learning models can significantly reduce this modeling effort. To ensure a machine learning model's reliability in all operating states, it is essential to know its validity domain. Operating states outside the validity domain might lead to extrapolation, resulting in unpredictable behavior. This paper addresses the challenge of identifying extrapolation in data-driven building energy system models and aims to raise knowledge about it. For that, a novel approach is proposed that calibrates novelty detection algorithms towards the machine learning model. Suitable novelty detection algorithms are identified through a literature review and a benchmark test with 15 candidates. A subset of five algorithms is then

Abbreviations: ML, Machine Learning; ANN, Artificial Neural Network; MPC, Model Predictive Control; DDMPC, Data-Driven Model Predictive Control; COP, Coefficient of Performance; BOPTEST, Building Optimization Testing Framework; kNN, k Nearest Neighbors; OCSVM, One-Class Support Vector Machine; SVM, Support Vector Machine; SVDD, Support Vector Data Description; DSVDD, Deep Support Vector Data Description; ABOD, Angle-Based Outlier Detection; LOF, Local Outlier Factor; CBLOF, Cluster-Based Local Outlier Factor; LRD, Local Reachability Density; COF, Connectivity-Based Outlier Factor; IF, Isolation Forest; FB, Feature Bagging; PCA, Principal Component Analysis; RNN, Replicator Neural Network; PDF, Probability Density Function; PWE, Parzen Window Estimation; GMM, Gaussian Mixture Model; GPR, Gaussian Process Regression; MCD, Minimum Covariance Determinant; HBOS, Histogram-based Outlier Scoring; KDE, Kernel-based Density Estimation; ECOD, Empirical-Cumulative-distribution-based Outlier Detection

* Corresponding author.
E-mail address: mraetz@eonerc.rwth-aachen.de (M. Rätz).

https://doi.org/10.1016/j.egyai.2023.100324
Received 24 July 2023; Received in revised form 10 November 2023; Accepted 19 November 2023
Available online 25 November 2023

evaluated on building energy systems. First, on two-dimensional data, displaying the results with a novel visualization scheme. Then on more complex multi-dimensional use cases. The methodology performs well, and the validity domain could be approximated. The visualization allows for a profound analysis and an improved understanding of the fundamental effects behind a machine learning model's validity domain and the extrapolation regimes.

1. Introduction

The building sector is a significant contributor to climate change. It accounts for approximately 30% of global final energy demand and approximately 28% of the global CO_2 emissions [1]. System models are utilized in various applications to reduce the environmental impact of the building sector. Examples of these applications include model predictive control (MPC) and predictive maintenance. Physical models of buildings are typically based on simplified assumptions and can struggle to capture the complexity and variability of real-world systems. Additionally, physical models may require extensive calibration and adjustment to be accurate for different building types and configurations, which can be time-consuming and costly. By contrast, Machine Learning (ML) algorithms can be trained on data from various building types and configurations with the same automatable approach, making ML models a promising choice for their application in the building sector [2–5].

To ensure an ML model's reliability in all states, it is important to know the validity domain of ML models [2,6–8]. According to Brooks et al. [9], a distinction can be made between two areas in the data space:

- In areas with sufficient training data, predictions of ML models are reliable. Such predictions are called interpolations.
- In areas where no or insufficient training data are available, predictions are unreliable and are called extrapolation [9].

Accordingly, only areas where the ML model interpolates are part of the validity domain. A complete exploration of the system states is commonly expected for training the ML models [10,11]. This is an unrealistic scenario in building energy systems. Most operating strategies in the field specify constant set points or run unchanging sequences. As a result, only a subset of the possible operating states is available to the ML model for training. This is especially critical if the original control shall be substituted with a data driven model predictive control (DDMPC), as the DDMPC exploits a much larger range of operating states to improve operation, finally leading to a malfunctioning DDMPC as the ML model is not valid in these new operating states. Extrapolation is particularly prevalent in high-dimensional data. As the dimensionality increases, the number of points needed to cover a region sufficiently grows exponentially [10]. With the continuously growing complexity of building energy systems, this is becoming a persistent problem.

This paper aims to identify these extrapolation regimes to improve the application of ML models in building energy systems. For that, first, the fundamentals for extrapolation detection are given in Section 2, including an expounding review of novelty detection algorithms in Section 2.2. The state of the art of extrapolation detection is outlined in Section 3. A novel extrapolation detection methodology that calibrates novelty detection algorithms towards the ML model is presented in Section 4. In Section 5.1, 15 novelty detection algorithms are benchmarked on public benchmark data sets. With that benchmark and the theoretical information from the novelty detection review, a subset of five suitable algorithms is selected in Section 5.2. This subset of algorithms is evaluated on building energy systems in Section 6. The use cases are described in Section 6.1. The evaluation is conducted, first, on two-dimensional data allowing for visualization and comprehension education in Section 6.2, then on more complex multi-dimensional use cases in Section 6.3. The findings of this paper are concluded in Section 7.

2. Fundamentals of extrapolation detection

As mentioned in the previous section, regions of extrapolation refer to areas within the systems state space containing insufficient training data and, therefore, leading to unreliable predictions [9]. In that sense, we define extrapolation detection with the following:

Extrapolation detection refers to the process of identifying instances where the ML model is extrapolating beyond the range of the training data.

The literature differentiates two main types of uncertainty, aleatoric and epistemic uncertainty [11,12]. Aleatoric uncertainty refers to the statistical uncertainty and is induced, e.g., through noise. It is an uncertainty present within the interpolation region. Epistemic uncertainty refers to the systematic uncertainty and is induced by the lack of system knowledge, e.g., in extrapolation regions. Hence, extrapolation detection can also be described as the process of identifying data points with high epistemic uncertainty, which is considered significantly more difficult than estimating the aleatoric uncertainty [13].

We identified two categories of identifying uncertainties, *model-centric* and *data-centric* approaches.

Model-centric approaches use the ML model itself to determine uncertainties. For that, the ML model is alternated, mostly through probabilistic techniques. In literature, this is often referred to as predictive uncertainty or probabilistic prediction, e.g., through monte carlo dropout [14] or ML model ensembling [15]. The output is often some confidence interval containing both aleatoric and epistemic uncertainty in one measure.

Data-centric approaches, on the other hand, are applied to the data directly, e.g., through calculating the distances between data points, leaving the ML model untouched. The literature often refers to this with outlier, anomaly, out-of-distribution or novelty detection. Data-centric approaches usually determine epistemic uncertainty exclusively. Some literature equates this to identifying the validity domain of an ML model [6–8,16].

2.1. Specifications for the application in building energy systems

The previous section introduces the differentiation between model-centric and data-centric approaches. The main advantage of some model-centric approaches is the generation of a confidence interval that can estimate the amplitude of the ML model's prediction error. Also, their uncertainty prediction originates directly from the ML model, which is beneficial to approximate their validity. Though, there are several arguments for preferring data-centric over model-centric approaches for extrapolation detection in building energy systems: Data-centric approaches exclusively determine epistemic uncertainty, which is desired for detecting the validity domain. Hüllermeier et al. [11] emphasize the importance of distinguishing epistemic and aleatoric uncertainty, of which model-centric approaches are incapable. Separating the ML model from the extrapolation detector allows for more focused development of each, a more straightforward implementation, improved comparability between studies, and the comparison of the extrapolation capability of different ML models. Also, as there is no single best ML algorithm, it is beneficial if extrapolation detection can be applied to all types of ML models. It also preserves all options for developing extrapolation-capable ML models. Through data-centric approaches, the ML model can stay less complex, as it does not need probabilistic properties to produce an uncertainty measure. This, in turn, is very beneficial for its usage in DDMPC.

Additionally, Sluijterman et al. [17] show that model-centric approaches, demonstrated for monte carlo dropout and naive bootstrapping, are not designed for detecting epistemic uncertainty and may lead to false uncertainty measures in extrapolation regions. Even though model-centric approaches can potentially approximate epistemic uncertainty well, they might as well fail in other cases [15,18].

Out of the data-centric approaches, novelty detection is specialized in having no or few outliers during training, which is the case in extrapolation detection. Hence, this paper focuses on using novelty detection algorithms to provide for extrapolation detection. We consider an extrapolation detector an algorithm that aims to approximate the validity domain of an ML model. In that sense, a novelty detection algorithm becomes an extrapolation detector if calibrated towards the ML model, i.e., its hyperparameters are set in a way it approximates the validity domain of an ML model instead of differentiating inlier and outlier in an unsupervised manner.

To identify the validity domain of building energy system ML models through novelty detection algorithms, the following characteristics are considered beneficial:

1: System unspecific The detection should work reasonably for most systems within the context of building energy systems. For that, it should not state rigid assumptions about the data distribution.

2: Computational cheap inference The inference is critical for online applications like DDMPC. The inference time defines how often, e.g., per hour, that online application can be triggered.

3: Computational feasible training The algorithm should be trainable with commonly available hardware and within a reasonable time. For online learning, training should be performed within minutes or hours; for the initial training, it may be tolerable to wait for days.

4: Resilient against extrapolation The extrapolation detection algorithm cannot be based on extrapolation-prone approaches itself, e.g., training an Artificial Neural Network (ANN) to detect extrapolation. In other words, the approach must be valid for detecting epistemic uncertainty.

2.2. Review of novelty detection

As mentioned in the previous section, novelty detection can provide algorithms to perform extrapolation detection. An expounding review of potential algorithms is given in this section. Novelty detection algorithms aim to distinguish known data points from unknown data points. Unknown data points are those which are significantly different from the known data based on their characteristics. Depending on the application, the unknown data points are called outliers [19], anomalies [20], or novel data points [21]. In this paper, we uniformly use the terms inlier and outlier. In novelty detection, the class of outliers is underrepresented in the training data [22]. Therefore, novelty detection algorithms learn to distinguish between the inlier and outlier classes by assuming that all training data points are inliers. This scenario is called one class classification [23].

In the context of extrapolation detection, novelty detection algorithms identify the validity domain of an ML model. An outlier indicates an data point outside the ML model's validity domain. An inlier indicates an data point inside the validity domain. In extrapolation detection, such a data point is an input sample for the ML model. Similarly to classical novelty detection, extrapolation detection generally considers available data points to be inliers and is thus trained only on inliers.

The variety of novelty detection algorithms can be categorized by their assumption on the characteristic of an outlier, e.g., an outlier is far away from other points. Though, the categories are not uniformly defined in the literature [19,21]. Also, some algorithms use assumptions from different categories and cannot be precisely classified. Fig. 1 shows one possible categorization. Each category is presented with explanatory basics, its underlying assumption, advantages and disadvantages. We also introduce representative algorithms of each relevant category, which are evaluated in Section 5. Related full-length reviews can be found from Chandola et al. [20] (anomaly detection), Goldstein et al. [24] (unsupervised anomaly detection), Aggarwal [19] (outlier detection), Khan and Madden [23] and Tax [25] (one-class classification), and Pimentel et al. [21] (novelty detection).

2.2.1. Distance-based

Distance-based approaches use the distance to other data points to evaluate the novelty score of a specific data point. The algorithms differ in selecting these other points, which are used to determine the distance and the way the distance is calculated. Distance-based approaches can be divided into three subcategories: k Nearest Neighbors (kNN), cluster-based, and relative density [21].

Nearest neighbor algorithms are among the most commonly used methods for novelty detection [21]. To determine the novelty score, the distance to the k nearest neighbors of a data point is determined based on the chosen distance function [26], e.g., the Euclidean distance [20]. Another algorithm that indirectly belongs to the kNN category is Angle-Based Outlier Detection (ABOD) [27]. It is designed explicitly for high-dimensional datasets, using the variance of the angles between the connection vectors of a data point and all nearest neighbors to determine the novelty score.

Assumption: Inliers have near neighbor points, while outliers lie far from other points.

Advantages: No a priori assumption about the underlying data distribution is required [21]. Often provides more accurate classifications than algorithms from the other two subcategories of distance-based approaches [19]. Commonly provides reliable classifications [19] even with a small amount of training data.

Disadvantages: May be affected by the curse of dimensionality [28] if the dataset has a large number of dimensions [21]. Does not scale well with the number of training data since all data points must be stored for nearest neighbor point determination [29].

Cluster-based approaches use a clustering algorithm to divide the data points into a given number of clusters such that similar data points belong to the same cluster. The k-means algorithm is often used to determine the clusters [21].

A well-known method from this category is Cluster-Based Local Outlier Factor (CBLOF) [30]. CBLOF determines the novelty score via the size of the cluster to which the data point is assigned and the distance to the nearest cluster center. A detailed overview of other cluster algorithms is provided by Ding et al. [29] and Goldstein et al. [24].

Assumptions: Inliers belong to a cluster, while outliers cannot be assigned to a cluster [20]. Inliers are close to a cluster center, while outliers appear far from them [20]. Inliers belong to large and dense clusters, while outliers belong to small or sparse clusters [20].

Advantages: Compared to the kNN algorithm, cluster-based approaches have a faster inference because a test data point is compared only with the previously determined clusters and not with all training data points [20].

Disadvantages: The classification accuracy of cluster-based approaches depends heavily on the effectiveness of the chosen clustering algorithm [20].

Relative density algorithms compute the relative density of a data point. The major difference between cluster-based approaches and algorithms based on relative density is that cluster algorithms partition the data points, while relative density-based algorithms partition the data space [24]. A well-known representative from this category is the Local Outlier Factor (LOF) [31]. It computes the Local Reachability Density (LRD) of a data point and evaluates it relative to the LRD of the k nearest neighbors.

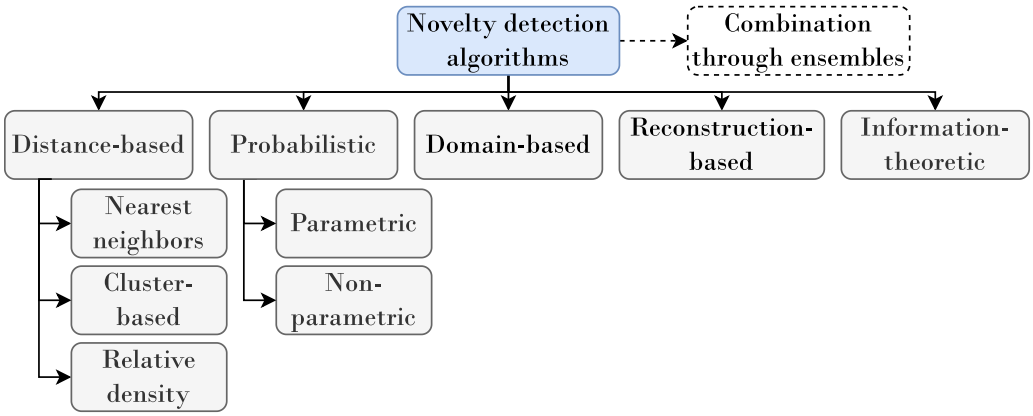


Fig. 1. Overview of novelty detection categories.

Assumption: A data point in a low-density neighborhood is an outlier, while inliers lie in a dense neighborhood [20].

Advantages: Due to their local approach, these algorithms are particularly suitable for identifying local outliers [19].

Disadvantages: Weak in identifying global outliers [24].

2.2.2. Probabilistic

Probabilistic approaches fit a statistical model to the training data. They often determine the empirical probability density function (PDF) of the data.

The idea is that outliers are unusual events at the edges of the distribution function [32]. Two subcategories can be distinguished: parametric and non-parametric.

Parametric approaches make an assumption about the underlying distribution function and determine a fixed number of parameters based on the data. In contrast, non-parametric approaches make no assumption about the distribution, and the complexity of the distribution function increases with the number of data points [20]. Parametric algorithms are, for example: Minimum Covariance Determinant (MCD) and Gaussian Mixture Model (GMM). MCD is used for estimating the mean and covariance matrix [33]. GMM consists of a mixture of weighted Gaussian distributions [21], which can be interpreted as a probabilistic version of clustering algorithms (see 2.2.1) [19].

Non-parametric algorithms are, for example: Histogram-based Outlier Scoring (HBOS), Kernel-based Density Estimation (KDE) and Empirical-Cumulative-distribution-based Outlier Detection (ECOD). HBOS [34] uses the intuitive approach of histograms, which despite its simplicity achieves good classification accuracies [20]. However, a disadvantage is that the individual attributes are assumed to be independent of each other. KDE distributes a large number of kernel functions over the data space [21] to obtain an empirical estimate of a PDF. A commonly used implementation of KDE is the Parzen Window Estimation (PWE) [35]. KDE has good accuracy [25] but time-intensive testing due to evaluating the kernel function for each training data point [36]. ECOD computes an empirical distribution per dimension of the data and aggregates the probability of data point laying in the tails of these distributions across dimensions. It does not require hyperparameters [32], which is beneficial as hyperparameter tuning is difficult in typical novelty detection tasks. Additionally, the well-known algorithm Gaussian Process Regression (GPR) [37] can be named here. Even though we consider it model-centric, as it can be used as the ML model for modeling the system while simultaneously providing an uncertainty measure, the calculation of the uncertainty measure relates to the non-parametric probabilistic approaches [38].

Assumption: Inliers are located in regions of high probability density, while outliers are in regions of low probability density [20].

Advantages: Probabilistic approaches are based on mathematical principles and can effectively identify outliers if an accurate estimate of PDF can be determined [21]. Probabilistic approaches often achieve high classification accuracy.

Disadvantages: For accurate estimation of PDF, these approaches require sufficient training data points [21]. In particular, parametric approaches are based on the assumption that a particular distribution function generates the data. This assumption often does not hold for high-dimensional data [20].

2.2.3. Domain-based

Domain-based approaches determine the decision boundary as an envelope separating inliers and outliers in the data space. A well-known representative of domain-based approaches is the One-Class Support Vector Machine (OCSVM) [39], which transfers the operation of a Support Vector Machine (SVM) [40] to a one-class scenario. A similar algorithm is Support Vector Data Description (SVDD) [25]. For which Ruff et al. [41] developed the deep learning approach Deep Support Vector Data Description (DSVDD). Here, the mapping into the feature space is not given by a kernel function but learned using a ML model based on the data.

Assumption: Given a feature space, a classifier can be trained to distinguish between inlier and outlier classes [20].

Advantages: Unlike probabilistic approaches, domain-based ones require fewer training data points because they only determine an envelope rather than the entire PDF [6]. Domain-based approaches can achieve a fast inference because the test data points are only compared to the envelope determined in training [29].

Disadvantages: By focusing on classification, domain-based approaches lack interpretable novelty scores [20]. Depending on the hyperparameters, the envelope does not enclose the inliers tightly enough, and thus outliers are classified as inliers, and vice versa [29].

2.2.4. Reconstruction-based

Reconstruction-based approaches model the training data using dimensionality reduction, which maps the data into a low-dimensional subspace. The back transformation causes a reconstruction error. Given a test data point to be evaluated, this reconstruction error is taken as the novelty score [21]. Well-known representatives are the Replicator Neural Network (RNN) [42] and the Principal Component Analysis (PCA) [43]. RNN are also called autoencoder [44], a neural network with nonlinear dimensionality reduction capability. PCA is a technique that performs an orthogonal base transformation of the data into a low-dimensional subspace [21].

Assumption: The data can be mapped into a low-dimensional subspace in that inlier and outlier differ significantly [20].

Advantages: Dimensionality reduction can counteract the curse of dimensionality. No a priori assumptions are made about the data distribution [21].

Disadvantages: For successful dimension reduction, appropriate values must be found for the hyperparameters that control the mapping into the lower-dimensional subspace [21]. Due to the dimension reduction, the interpretability of the novelty degree is low [19].

2.2.5. Information-theoretic

Algorithms from this category examine the information content of a data set using various measures from information theory, such as Kolmogorov complexity [45], entropy, or relative entropy. A detailed overview of this category is given by Aggarwal et al. [19] and Pimentel et al. [21] give a detailed overview of this category.

Assumption: Outliers cause significant deviation in the information content of a data set [20].

Advantages: No a priori assumptions are made about the data distribution [21].

Disadvantages: The accuracy of these approaches depends heavily on the choice of information measure [20]. A sufficient number of outliers within the training data is needed for correct classification [21]. Information-theoretic approaches often only provide a classification; an interpretable novelty score is challenging to determine [20,21]. They tend to be computationally expensive [21].

2.2.6. Ensembles

Ensembles combine novelty detection algorithms, which can vary in complexity, to increase overall prediction accuracy. According to Zimek et al. [46] there are ensembles of classifiers that:

1. are trained with different subsets of data variables.
2. are trained with different subsets of data samples.
3. are based on random methods.
4. use the same method but with different parameterizations.
5. are based on different algorithms.

Popular categories are the first and third. An algorithm of the first category is Feature Bagging (FB) [47]. FB combines the output of different classifiers, each trained with a different random subset of data variables. Isolation Forest (IF) can be allocated in the third category [48]. Depending on the definition, IF can be called an ensemble or a standalone novelty detection algorithm. It creates multiple trees and determines the novelty score based on the average path length from the root to the terminating node. IF works with the assumption that outliers are in the minority and have values significantly different from inliers. For a presentation of the other categories, please refer to [25,46].

3. Literature review on extrapolation detection

This section proceeds from novelty detection algorithms to their application as extrapolation detectors. A literature review is conducted to identify the state of the art of extrapolation detection. Table 1 provides a summary of the literature on identifying validity domains using data-centric approaches. Distance-based approaches, in the form of kNN are proposed by Simutis et al. [49], Lee et al. [50], Teixeira et al. [51], and Rall et al. [52]. Non-parametric probabilistic approaches, in the form of KDE are proposed by Leonard et al. [7], Bishop et al. [36] and Lee et al. [50]. Domain-based approaches, like the convex hull [6,8,53,54] and SVMs, in the form of OCSVM [6] and SVDD [55,56], are also proposed. Some references combine model and data-centric approaches to be used simultaneously to increase robustness [7,53,54]. Here, the validity domain is identified by novelty detection to detect and avoid extrapolations. Interpolations are additionally provided with a model-centric uncertainty measure.

Table 1 shows that the use cases considered in the literature are often from the chemical sector and none related to building energy systems. In a separate literature review, we extended the scope to model-centric approaches and identified several publications applying model-centric uncertainty considerations in the context of building energy systems.

Typically, a GPR is the algorithm of choice, e.g., Maddalena et al. [57] use GPR in DDMPC and optimize towards low variances, i.e., penalizing predictions with high uncertainty in the DDMPC solver. They evaluate the methodology for controlling an industrial cooling plant for a hospital surgery center. Jain et al. [58] apply GPR in a stochastic DDMPC to manage a demand response event on buildings simulated through EnergyPlus. Also Nghiem et al. [59] use GPR to perform demand response on building energy simulations with stochastic DDMPC.

Though, GPR is problematic for large data sets, as its calculation effort scales exponentially with the amount of data [57,60]. Sparse GPR [60,61] can produce relief but may restrain the uncertainty measure's expressiveness. To the best of our knowledge, no publication suggests data-centric extrapolation detection for building energy systems.

There are larger related research fields, e.g., a lot of research on quantifying an ML model's uncertainty uses model-centric approaches [11,13,15,17,18,62,63]. There is also abundant literature using novelty detection algorithms in general [21,64–66]. But only a small subset uses novelty detection for detecting the validity domain of an ML model, shown in Table 1. And, to the best of our knowledge, none uses novelty detection algorithms with hyperparameters calibrated towards the ML model itself. Schweidtmann et al. [6], for example, gradually decrease the hyperparameter until the number of support vectors does not decrease much more [67]. Hence, they employ an approach that does not calibrate towards the validity domain of the ML model. Also, there is only one comparison between novelty detection algorithms for extrapolation detection. Schweidtmann et al. [6] compare OCSVM and convex hulls. Furthermore, two-dimensional visualizations to increase understanding are only employed by Teixeira et al. [51] (iso-uncertainty lines), Malak et al. [55] (geometric shapes), Bae et al. [53] (carpet plots), Pineda et al. [54] (iso-uncertainty lines), Schweidtmann et al. [6] (geometric shapes), and none enables to assess the accuracy of the ML model or to compare it with the extrapolation boundary.

Given the identified research gaps, this paper contributes to the state of the art

- by proposing a method to calibrate novelty detection algorithms towards the ML model to detect its validity domain.
- by applying data-centric extrapolation detection to building energy systems.
- by comparing multiple novelty detection algorithms. First through an algorithm review (Section 2.2), then through a benchmark with 15 algorithms (Section 5.1), and finally through an in-depth comparison of five algorithms on two and multi-dimensional use cases (Section 6).
- by proposing a novel visualization scheme for two-dimensional use cases enabling in-depth educational analysis.

4. Methodology

The proposed extrapolation detection methodology that calibrates a novelty detection algorithm towards an ML model is presented in the following. An overview of that process is shown in Fig. 2.

First, an ML Model is trained and tested, which is described in Section 4.1. After that, the actual extrapolation detection methodology follows. Beginning with the specification of the validity domain of the ML model using a threshold. Section 4.2 explains that process. The resulting classification ground truth, consisting of data points labeled as either inlier or outlier, is then used for the training, hyperparameter tuning, and evaluation of a novelty detection algorithm. This process is explained in Section 4.3.

This methodology is applied to two- and multi-dimensional one-year building energy systems datasets in Section 6. The one-year dataset is split into a model generation and an evaluation period. The model generation period is used to generate the ML model and the extrapolation

Table 1
Literature on extrapolation detection using at least one data-centric approach.

Ref.	Authors	Year	Method used	Application
[7]	Leonard et al.	1992	KDE model-centric	Chemical reactor
[8]	Courrieu et al.	1994	convex hull	Basic literature
[36]	Bishop et al.	1994	KDE	Oil pipeline
[49]	Simutis et al.	1995	kNN	Brewery fermenter
[50]	Lee et al.	2005	kNN KDE	Chemical reaction
[51]	Teixeira et al.	2006	clustering	Bioprocess
[55]	Malak et al.	2010	SVDD	None
[56]	Roach et al.	2012	SVDD	Engineering design
[52]	Rall et al.	2019	kNN	Membranes
[53]	Bae et al.	2020	convex hull model-centric	Bioreactor
[54]	Pineda et al.	2021	convex hull model-centric	Basic literature
[6]	Schweidtmann et al.	2021	convex hull OCSVM	Desulfurization

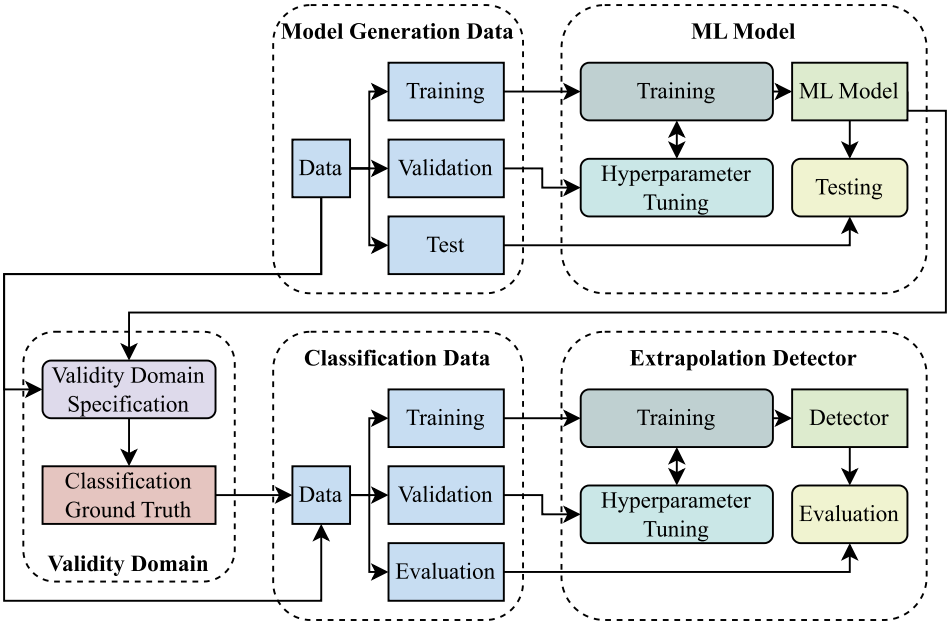


Fig. 2. Overview of the proposed extrapolation detection methodology. A novelty detection algorithm is calibrated towards the ML model to detect its validity domain.

detector and would, in a real application, represent the available data. The evaluation period is used to evaluate the extrapolation detector and would, in a real application, represent the continuous operation of the machine learning and extrapolation detection for, e.g., MPC. It represents the data gathered during the system's operation after generating the ML model and the extrapolation detector, i.e., unavailable data in a real application.

For the benchmark test, the ground truth is given by the benchmark datasets. Hence, only the hyperparameter tuning of this methodology is used for the benchmark tests.

4.1. Training of the ML model

The model generation period data is shuffled and split into 80% training, 10% validation, and 10% test data points. The validation data points are used for hyperparameter tuning, while the test data points are used for ML model testing. All data points are normalized.

In this work, we exemplary consider ANNs as ML models. We train the ANNs on the training data $\boldsymbol{X} = [x_0, \dots, x_N]$, $x_i \in \mathbb{R}^D$, with N data points and the input dimensionality D.

With the output of the ANNs $\hat{f}(\cdot)$ we predict a function value $f(\cdot)$ of an unknown function f at a test point $\hat{x}$.

In this work, the model architecture is tuned using grid search. Per use case, 20 ANNs with one hidden layer are trained with a varying number of neurons between 4 and 32. The logistic sigmoid function and relu function [38] are activation function candidates of the hidden layer. A linear activation is used for the output layer. The input layer uses batch normalization to stabilize and speed up the training process [68]. The ANNs are implemented in Python using the package Keras [69] and the Adam optimizer [70]. Based on the test results, we use the best ANN as ML model.

There are more sophisticated procedures for generating and tuning ML models, e.g., [71–73], which are not the focus of this paper.

4.2. Determination of the true validity domain

We assume that, after training, the ML model is sufficiently accurate within the seen data regime for the intended application. This is commonly approximated by assessing the accuracy of the test data set. Based on this assumption, most of the training, validation, and test data points lie inside the validity domain.

All predictions $\hat{f}(\cdot)$ with a prediction error $|\hat{f}(\cdot) - f(\cdot)|$ higher than a threshold σ are considered outside of the validity domain.

This work proposes a heuristic to determine the threshold σ. It is assumed that 5% of the data points of the model generation period lie outside the validity domain. Thus, we calculate a use case dependent accuracy threshold σ, which represents the boundary of the validity domain.

A data point resulting in a prediction error above that threshold is outside the validity domain and vice versa. The data points labeled as either inlier or outlier form the classification ground truth for the hyperparameter tuning in Section 4.3.

In practical applications, we suggest choosing this threshold based on the prediction accuracy required for fulfilling the task, determining this required prediction accuracy requires expert knowledge and might be a research question in itself. With our heuristic, we ensure that we have data points outside of the validity domain to be utilized for hyperparameter tuning of the novelty detection algorithms, described in the following Section 4.3.

4.3. Extrapolation detection

Based on the training data $\boldsymbol{X}$, a novelty detection algorithm assigns a novelty score $c(\hat{x})$ to a test point $\hat{x}$. The higher the novelty score, the more unknown a data point is with respect to the training data. Depending on a novelty threshold ρ, the test point is classified:

$$C(\hat{x}) = \begin{cases} 0 \text{ (Inlier)} & \text{if } c(\hat{x}) \leq \rho \\ 1 \text{ (Outlier)} & \text{if } c(\hat{x}) > \rho \end{cases} \tag{1}$$

Inliers indicate interpolation, whereas outliers indicate extrapolation. Therefore the threshold ρ defines the decision boundary of the algorithm, i.e. the *extrapolation boundary*.

An extrapolation detector has to balance two conflicting targets: *precision* and *recall*. The *precision* is the number of correctly detected extrapolations divided by the number of all detected extrapolations. The *recall* is the number of correctly detected extrapolations divided by the number of all extrapolations.

We evaluate the performance of an extrapolation detector by using the F-score for classification as a trade-off between precision and recall:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2}$$

With the classification ground truth described in Section 4.2, we tune the hyperparameters of the novelty detection algorithms, after which we consider them extrapolation detectors.

We propose individual data splits for the ML model and the extrapolation detector, shown in Fig. 3. For the training of the extrapolation detector, the data splitting used for generating the ML model would not be meaningful. Novelty detection algorithms learn the difference between inliers and outliers only based on the characteristics of inliers, i.e., valid data points. Therefore, they are based on the assumption that all training data points are inliers. This is called one-class classification. However, after determining the validity domain, some of the available data points might be outliers. Therefore, all outliers are moved from the training to the validation split. As the extrapolation detector is evaluated on the evaluation period, the test data from the ML model generation is also added to the validation data of the extrapolation detector. Preliminary experiments showed that it is beneficial to include the extrapolation detector's training data in the validation data set. Hence, this data is also added. Finally, the validation period of the extrapolation detector equals the model generation period.

The extrapolation detector is trained with the training data. Its hyperparameters are scored, i.e., tuned, on the validation data using the F-score. For calculating the F-score, some outliers are required in the validation data. The decision threshold ρ is also considered a hyperparameter in addition to the novelty detection algorithm's hyperparameters. For e.g., kNN, these are the distance metric, the number of neighbors, and whether to use the maximum or mean distance of the neighbors.

Table 2
Overview of used benchmark data sets.

Name	Data points	Dimensions	Outlier share
Ionosphere (Ionos)	351	33	36%
WBC	378	30	6%
Pima	768	8	35%
Vowels	1456	12	3%
Letter	1600	32	6%
Cardio	1831	21	10%
Satimage-2 (Satim-2)	5803	36	1%
Satellite (Satel)	6435	36	32%
Pendigits (Pendig)	6870	16	2%
Shuttle	49 097	9	7%

The novelty detection algorithms are implemented in Python using the packages *PyOD* [74] and *Scikit-learn* [75]. For the hyperparameter tuning, we use the Tree Of Parzen Estimators through the package *Hyperopt* [76].

4.4. Detector tuning using ideal data

To assess the potential of optimally tuned algorithms, we also perform a hyperparameter tuning under ideal information availability. For this *ideal tuning*, the evaluation data is added to the validation data. In that way, the algorithm's hyperparameters are tuned towards the ground truth of the whole data set while still using the same training points. This is a theoretical consideration since, in a real application, the evaluation data is not available. However, the ideal tuning provides information about how accurately an algorithm can map the validity domain, having the best possible hyperparameter set.

5. Pre-selection of algorithms

To assess the suitability of novelty detection algorithms for extrapolation detection, a benchmark test with the in Section 2.2 introduced representative algorithms is conducted. For the benchmark test, the following algorithms are considered: kNN, ABOD, LOF, MCD, GMM, HBOS, ECOD, KDE, GPR, OCSVM, DSVDD, RNN, PCA, FB (as an ensemble of kNNs) and IF. The goal is to select a subset of a maximum of five suitable algorithms for the building energy system use cases in Section 6.

5.1. Benchmark-test

Novelty detection benchmark tests from literature, e.g., the comprehensive benchmark by Han et al. [64], commonly do not include a hyperparameter tuning of the novelty detection algorithms. The following benchmark test employs hyperparameter tuning. The benchmark evaluates the classification performance and the required training and test time. The used data sets are summarized in Table 2 and taken from the *ODDS (Outlier Detection DataSets)* library [77]. The data sets originate from a diverse field of applications, e.g., the data set "Cardio" consists of measurements of fetal heart rates and uterine contraction features on cardiotocograms classified by expert obstetricians. The selection of data sets shall represent typical building energy system data sets in terms of the number of samples and features. To exemplify, a one-year data set with a 15 min interval results in 35040 samples, and Stoffel et al. use 9 to 26 features [78] and Zhang uses 24 features [79].

The algorithms' hyperparameters are tuned as described in Section 4.3. The results are an average of 10 independent experiments with randomly shuffled data points to avoid stochastic perturbation. The benchmark test is conducted on a virtual machine with 16 GB RAM and an Intel Xeon CPU E5-2690 v4 @ 2.60 GHz.

Table 3 presents the evaluation via the F-Score, Table 4 presents the training time, and Table 5 shows the test time for 100 data points. The color scheme ranks the algorithms within each data set, i.e., per row. The results are discussed in Section 5.2.

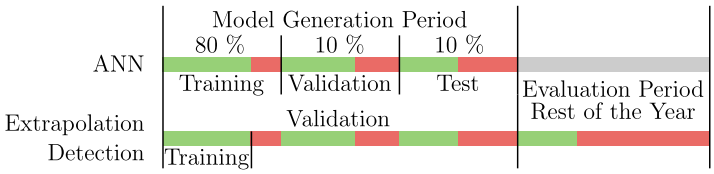


Fig. 3. Overview of the data used for training, validation, testing, and evaluation. Green bars indicate inliers, whereas red bars indicate outliers. It is assumed that more outliers lie in the evaluation than in the model generation period. The scale of the periods is only schematic. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Benchmark-Test: Accuracy evaluation via the F-Score.

	Distance			Probabilistic						Domain		Reconstr.		Ensemble	
Name	kNN	ABOD	LOF	MCD	GMM	HBOS	ECOD	KDE	GPR	OCSVM	DSVDD	RNN	PCA	FB	IF
Ionos	0.847	0.822	0.721	0.780	0.650	0.619	0.557	0.861	0.774	0.801	0.784	0.685	0.682	0.837	0.756
WBC	0.401	0.424	0.457	0.274	0.219	0.370	0.321	0.375	0.464	0.646	0.275	0.408	0.401	0.445	0.413
Pima	0.564	0.553	0.540	0.556	0.541	0.545	0.529	0.563	0.556	0.552	0.527	0.527	0.543	0.567	0.561
Vowels	0.707	0.664	0.615	0.135	0.288	0.124	0.081	0.807	0.708	0.758	0.074	0.084	0.099	0.653	0.212
Letter	0.437	0.373	0.387	0.236	0.199	0.163	0.132	0.424	0.401	0.434	0.195	0.123	0.125	0.472	0.153
Cardio	0.657	0.652	0.648	0.386	0.432	0.373	0.492	0.686	0.645	0.729	0.189	0.629	0.624	0.646	0.592
Satim2	0.892	0.903	0.808	0.713	0.040	0.762	0.727	0.911	0.884	0.929	0.643	0.808	0.822	0.883	0.867
Satel	0.573	0.543	0.533	0.496	0.520	0.595	0.482	0.584	0.492	0.577	0.491	0.482	0.482	0.577	0.533
Pendig	0.906	0.774	0.828	0.124	0.127	0.239	0.197	0.912	0.822	0.904	0.054	0.208	0.228	0.941	0.465
Shuttle	0.971	0.973	0.966	0.407	0.946	0.977	0.696	0.969	-	0.972	0.472	-	0.922	0.968	0.966
Mean	0.695	0.668	0.650	0.411	0.396	0.477	0.421	0.709	0.638	0.730	0.370	0.439	0.493	0.699	0.552

Table 4
Benchmark-Test: Training-Time in seconds.

	Distance			Probabilistic						Domain		Reconstr.		Ensemble	
Name	kNN	ABOD	LOF	MCD	GMM	HBOS	ECOD	KDE	GPR	OCSVM	DSVDD	RNN	PCA	FB	IF
Ionos	0.00	0.40	0.01	0.05	0.01	0.01	0.00	0.00	0.03	0.01	4.74	6.79	0.00	0.04	0.27
WBC	0.01	0.36	0.01	0.08	0.01	0.01	0.00	0.00	0.09	0.01	4.91	7.26	0.00	0.05	0.28
Pima	0.01	1.04	0.01	0.08	0.12	0.00	0.00	0.00	0.24	0.02	5.60	8.12	0.00	0.05	0.28
Vowels	0.02	0.41	0.02	1.14	0.03	0.00	0.01	0.00	1.61	0.17	9.44	12.91	0.01	0.13	0.33
Letter	0.02	0.72	0.02	2.11	0.01	0.01	0.01	0.01	3.04	0.24	9.10	14.81	0.02	0.20	0.34
Cardio	0.02	0.87	0.02	0.68	0.01	0.01	0.01	0.01	2.10	0.08	10.21	16.12	0.03	0.53	0.34
Satim2	0.28	11.10	0.20	4.23	2.59	0.02	0.06	0.04	49.97	0.60	26.40	47.15	0.07	1.76	0.58
Satel	0.18	5.11	0.18	3.57	2.60	0.02	0.03	0.03	36.28	3.07	18.27	28.62	0.02	1.89	0.62
Pendig	0.24	9.25	0.25	3.44	0.04	0.01	0.03	0.03	107.7	1.24	26.16	40.53	0.02	1.04	0.62
Shuttle	1.03	101.71	1.18	15.64	19.30	0.02	0.11	0.11	-	20.81	153.91	-	0.05	3.97	2.58
Mean	0.18	13.10	0.19	3.10	2.47	0.01	0.03	0.02	22.34	2.62	26.87	20.26	0.02	0.97	0.62

5.2. Evaluation and selection

To evaluate and select a performant subset of algorithms, we consider their theoretical properties, presented in Section 2.2, and their performance on the benchmark tests shown in the Tables 3–5. This subset of performant algorithms is then further analyzed in Section 6. In the context of extrapolation detection, an algorithm is considered performant if it can obtain good accuracy throughout different data sets with small computational effort during inference, reasonable computational requirements during training, and a reliable approach for extrapolation detection, as stated in Section 2.

Distance-based approaches have high overall performance, out of which kNN is the most performant algorithm in both accuracy and computational effort. ABOD is developed for high-dimensional datasets. Apparently, the evaluated datasets are not yet affected enough by the curse of dimensionality to see this advantage. As Goldstein et al. [24] already stated, global classifiers like kNN are preferred to local classifiers like LOF when no information about the nature of the outliers is available, which is the case. Also, kNN often provides reliable classifications even with a small amount of training data [19,64]. To conclude, ABOD and LOF are similarly accurate as kNN but do not show further advantages over kNN. Thus, we consider kNN for the building energy use cases. Cluster-based approaches are not investigated in this benchmark due to two reasons. First, in literature, kNN often provides more accurate classifications than cluster-based approaches [19]. Second, the computation times of kNN in the benchmark are already very good, so the faster inference of cluster-based approaches [20] does not provide a significant advantage.

Within the **probabilistic** algorithms, MCD, GMM, HBOS, and ECOD have overall low accuracies. These algorithms can be considered relatively simple, e.g., HBOS and ECOD assume univariate dependencies. KDE has the second-highest average accuracy overall, is computationally uncritical, and is hence selected for the building energy use cases.

Table 5
Benchmark-Test: Test-Time in seconds for 100 test data points.

	Distance			Probabilistic						Domain		Reconstr.		Ensemble	
Name	kNN	ABOD	LOF	MCD	GMM	HBOS	ECOD	KDE	GPR	OCSVM	DSVDD	RNN	PCA	FB	IF
Ionos	0.015	0.229	0.006	0.002	0.002	0.001	0.009	0.004	0.003	0.002	0.186	0.181	0.001	0.138	0.135
WBC	0.012	0.125	0.005	0.002	0.001	0.001	0.009	0.005	0.024	0.002	0.184	0.204	0.001	0.122	0.130
Pima	0.012	0.255	0.002	0.001	0.001	0.000	0.003	0.004	0.002	0.004	0.088	0.086	0.000	0.099	0.063
Vowels	0.013	0.036	0.002	0.000	0.000	0.000	0.003	0.010	0.008	0.009	0.052	0.057	0.000	0.107	0.036
Letter	0.018	0.065	0.002	0.000	0.000	0.000	0.006	0.017	0.010	0.013	0.041	0.041	0.001	0.153	0.035
Cardio	0.016	0.068	0.002	0.000	0.000	0.000	0.005	0.020	0.010	0.004	0.040	0.040	0.001	0.285	0.030
Satim2	0.029	0.251	0.005	0.000	0.001	0.000	0.011	0.079	0.068	0.008	0.032	0.032	0.001	0.251	0.014
Satel	0.044	0.153	0.006	0.000	0.001	0.000	0.008	0.063	0.041	0.054	0.014	0.014	0.000	0.312	0.013
Pendig	0.025	0.175	0.005	0.000	0.000	0.000	0.005	0.073	0.074	0.013	0.014	0.014	0.000	0.096	0.011
Shuttle	0.017	0.280	0.003	0.000	0.000	0.000	0.003	0.232	-	0.024	0.006	-	0.000	0.039	0.004
Mean	0.020	0.164	0.004	0.001	0.001	0.000	0.006	0.051	0.027	0.013	0.066	0.074	0.001	0.160	0.047

GPR shows average accuracy and fails to compute large data sets. GPR is a non-parametric algorithm that grows in size with the dimension of the data set [38]. This results in a RAM overflow for the large data set Shuttle. As mentioned in Section 2.2, GPR can be interpreted as a hybrid between model-centric and data-centric approaches. It is a popular algorithm, especially if an uncertainty measure is desired in addition to the prediction value. Also, it has been applied to building energy systems in literature [57–59]. Hence, we regard GPR as a baseline algorithm. And in turn, consider it for the building energy use cases.

Out of the **domain-based** algorithms, DSVDD exhibits both low accuracies and high to infeasible computational costs, see data set Shuttle. While OCSVM convinces with the best mean accuracy overall and fast inference. Domain-based approaches have the advantage over probabilistic approaches in that they generally require fewer training data points [6]. In addition, domain-based approaches can achieve a fast test phase because the test data points are only compared to the envelope determined in training, see [29]. Hence OCSVM is considered for the building energy use cases. The convex hull is often used in literature, presumably due to its simplicity. Though, it has the disadvantage of overestimating the validity domains [6]. In contrast, OCSVM can represent non-convex and thus much more complex envelopes [25]. This is why the convex hull is not considered for the benchmark.

Both **reconstruction-based** algorithms misclassify often. While PCA is at least computed easily, RNN could not even be computed for the largest data set. Some literature states (see Section 2.2) that one strength of reconstruction-based algorithms is their capability of handling high-dimensional data. As said earlier, the curse of dimensionality seems not yet to have a significant influence on the other algorithms, outplaying the reconstruction-based algorithms. Moreover, reconstruction-based approaches have some drawbacks, like difficult hyperparameter optimization [21]. Also, they are built with extrapolation-incapable models like ANNs, resulting in less reliable results. Reconstruction-based algorithms are neither theoretically nor practically promising for this work's goal and are not considered for the building energy use cases.

Ensemble algorithms have average (IF) to good (FB) accuracy. The used implementation of FB combines kNN algorithms. Compared to kNN, it achieves similar accuracies at slightly higher computational effort. As a result, we decided to use the less complex kNN algorithm instead of its ensemble. IF is considered very performant in the research community and is among the best algorithms for the PyOD benchmark test [64]. One reason why it is so popular and regularly considered performant is that IF has no hyperparameters, which is beneficial, as these are hard to tune in regular novelty detection use cases [64]. An advantage of IF is that the algorithm works reliably even with high-dimensional data sets with many irrelevant attributes and without outliers in the training data [48]. Compared to the other algorithms developed for high-dimensional data, like ABOD, RNN, and PCA, it shows a good mix of accuracy and computational efficiency. Due to its popularity, and as a representative of high-dimensional capable algorithms and representative of ensemble methods, we consider IF for the building energy use cases.

Information theoretic approaches typically detect the presence of novel data points only if there is a significantly large number of novel data points [21], which is not the case in extrapolation detection. In addition, it is considered difficult to assign a novelty score to a test point [21]. Given their disadvantages and the fact that they have also not been considered in the literature, see Table 1, we do not consider them for this work.

On a **general** view, the following findings are made. The tested data set sizes, which are typical for MPC in building energy systems, are still low-dimensional enough not to provoke the curse of dimensionality. The larger the data set, the more difficult the training. The computational effort, both training and inference, is higher for smaller outlier shares. With the slowest algorithm taking 153.92 s, all training times are acceptable for typical online learning intervals in building energy systems. Though, some GPR and RNN remained uncomputable for large data sets, see empty cells, e.g., Gaussian Process Regression through RAM overflow Regarding the inference time, all computable models are acceptable for common DDMPC approaches with time steps larger than 5 min. Still, for ABOD, GPR, DSVDD, and RNN, the inference time should be kept in mind as a possible bottleneck. The slowest algorithm took 0.312 s to infer 100 data points.

In summary, the following algorithms are considered for the building energy use cases:

- k Nearest Neighbors (kNN)
- Kernel-based Density Estimation (KDE)
- Gaussian Process Regression (GPR)
- One-Class Support Vector Machine (OCSVM)
- Isolation Forest (IF).

6. Analysis of pre-selected algorithms

6.1. Use cases

The suitability of the pre-selected algorithms for extrapolation detection is evaluated using two- and multi-dimensional building energy system datasets. The two-dimensional use case (Section 6.1.1) allows for graphical analysis of the resulting validity domain and extrapolation boundary, while the multi-dimensional use cases (Section 6.1.2) represent typical applications of ML models as process models in building energy systems.

Table 6
Features considered as input variables in the multi-dimensional use case. An X indicates that the current time step is considered. A number indicates that a lag is considered, e.g., −1 (−2) represents the previous (second previous) time step.

Feature	Description	ΔT_{zone}	P_{el}
T_{zone}	Zone temperature	X, −1, −2	X
T_{amb}	Ambient temperature	X, −1	X
u_{hp}	Heat pump modulation	X, −1, −2	X
$u_{hp,log}$	Logistic heat pump modulation	–	X
$\dot{q}_{sol,dir}$	Specific direct solar radiation	X	–
t_{day}	sin / cos (results in 2 features)	X	–
t_{week}	sin / cos (results in 2 features)	X	–
	Dimensionality	13	4

6.1.1. Two-dimensional

In the two-dimensional use case an ANN is trained to predict the heating-power of an air–water-heatpump $\dot{Q}_{heat}$ with an ideal coefficient of performance (COP):

$$\dot{Q}_{heat} = COP_{Carnot} \cdot P_{el} \tag{3}$$

$$COP_{Carnot} = \frac{T_{supply}}{T_{supply} - T_{amb}} \tag{4}$$

Input variables are the electric power P_{el}, supply temperature T_{supply}, and the ambient temperature T_{amb}. To reduce the input variables to a two-dimensional use case, the supply temperature is kept constant at 40 °C.

With that, we generate an hourly data set representing the correlation between ambient temperature and heating power for a typical single-family home for one year.

6.1.2. Multi-dimensional

In the multi-dimensional use cases, the *BESTEST Hydronic Heat Pump case* from the *Building Optimization Testing Framework* (Boptest) [80] is used for data generation. The test case is based on the BESTEST case 900 building with 192 m^2 [81] extended by an air-to-water modulating heat pump and an underfloor heating system. The training, validation, and test data is generated using the baseline controller with a 30-minute time step. For further information about BOPTEST, the reader is referred to the corresponding publication [80].

For that use case, two ANNs are trained for different parts of the system, one predicting the change of the zone temperature within a 30-minute interval (ANN-ΔT_{zone}) and another predicting the electric power of the heat pump (ANN-P_{el}).

To account for the slow thermal dynamics of the systems, also time-lagged measurements are considered as input variables to the ANNs. Table 6 shows the used input variables.

We consider the zone temperature T_{zone}, ambient temperature T_{amb}, the specific direct solar radiation $\dot{q}_{sol,dir}$, and the heat pump's modulation u_{hp}. Additionally, we introduce the logistic modulation $u_{hp,log}$ as a feature to support the learning of the heat pump's minimal power consumption. The logistic function continuously approximates a step that outputs 0 if $u_{hp} = 0$ and 1 if $u_{hp} > 0$. To estimate the periodic internal gains caused by user behavior, we use the time of the day t_{day} and the time of the week t_{week} encoded as sin/cos as additional inputs.

6.1.3. Algorithm evaluation process

For each use case, three different model generation periods with a varying number of samples are used, shown in Table 7. The ANNs are trained with data from the winter season, so extrapolations are expected in the summer and transition periods. The shorter the period, the fewer operation points are covered for the ML model to train on, so more extrapolation is expected.

Table 7
Model generation periods of the building energy system use cases.

Name	Number of data points Two-/Multi-dimensional	Model generation period
Short	336/672	First 2 weeks of January
Middle	744/1488	January
Long	1416/2832	January + February

6.2. Two-dimensional analysis

Multi-dimensional use cases can only be interpreted by performance indicators, of which their indication quality has not yet been analyzed or proven for the setting of extrapolation detection. To gain knowledge of the behavior and perform profound analysis, we propose a novel visualization for two-dimensional use cases, showcased in Fig. 4.

The two features displayed on the two axes span up the system's state space. The ANNs training data points are framed in black, and the validation data points in blue. The rest of the data, in that use case the rest of the year, is displayed unbordered. The fill color of the dots represents the absolute deviation of the ANN prediction from the original system, i.e., the Carnot formula, for that particular data point. Notably, there are accurate predictions (green fills) at most of the training and validation points. However, there are also data points, especially at the outer regions of the state space, where the prediction accuracy of the ANN decreases (red fills). Likewise, this visualization enables us to assess the accuracy of the ANN with respect to the location in the state space.

According to Section 4.2, a validity boundary can be drawn as an iso-accuracy border at the accuracy threshold σ, separating the high accuracy region from too low accuracy regions. In the depicted use case, the threshold σ is at an absolute error of 0.083 kW. This validity domain is shown in turquoise. The validity domain goes tightly around the training data in parts. In other parts, though, it goes far off the training data. These far-off areas can be interpreted as coincidentally good extrapolation, as the behavior outside the known data points is unknown by definition.

For this use case, these red dots refer to operation in summer, with high outdoor temperatures and low electric power of the heat pump, emphasizing that a model trained in winter should not be used for operation in the summer. We also summarize that the validity domain is strongly non-convex, even forming separated bubbles or holes.

The aspects described so far are identical for all five extrapolation detectors, as they only depend on the system and the ML model. Also, this data can only be shown when we analyze a simulated system. In a real application, during operation, only the bordered points would be known. The preliminary objective of the extrapolation detector is to mimic the validity domain with its extrapolation boundary (purple border), introduced in Section 4.3. Though, the extrapolation detectors regularly find some envelope for the training and validation data, mimicking the validity domain in those parts close to the data but cutting off these far-off regions. After understanding that these far-off regions coincidentally contain high ML model accuracies that cannot be determined (high epistemic uncertainty with coincidentally high accuracy), it is desirable that the extrapolation detectors reasonably wrap around the known data points without wrapping too close or too far. With these findings, we propose a sharpened definition of the validity domain and the extrapolation boundary.

"**The *Validity Domain*** refers to the area in the state space in which the ML model achieves, on average, sufficiently high accuracies".

In that context, a sufficiently high accuracy refers to the accuracy the ML prediction shall obtain in a particular application. The term "on average" excludes the effect of aleatoric uncertainty, e.g., through noise, to the validity domain, as aleatoric uncertainty randomly affects a data point's accuracy without forming a domain. The average accuracy for a data point, determined over a set of predictions affected

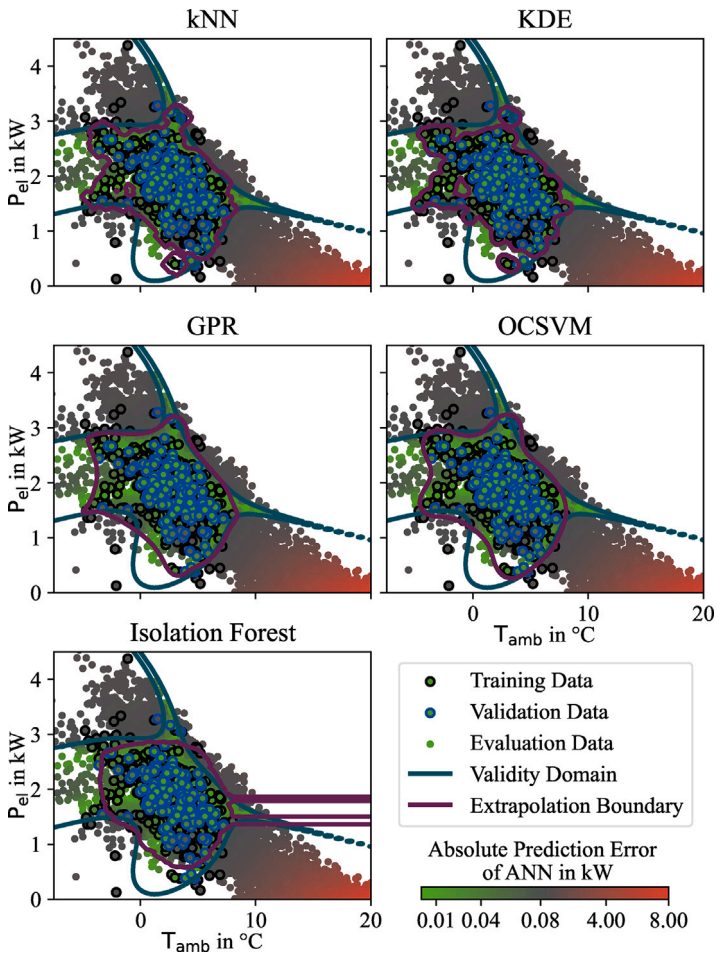


Fig. 4. Two-dimensional analysis using model generation period 'Middle'. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

by aleatoric uncertainty, reveals the expected mean accuracy, i.e., the validity of a data point. The currently applied accuracy threshold σ to determine the ground truth for the validity domain does not average over several samples of the same data point. Hence, for systems containing aleatoric uncertainty, this implementation is only an approximation of the validity domain.

"**The *Extrapolation Boundary*** separates areas with low epistemic uncertainty from areas with high epistemic uncertainty".

Considering the definition of the extrapolation boundary, IF exhibits unusual behavior, forming two horizontal beams. The cause can be deduced from the methodology of that algorithm, which is explained in Section 2.2.6. We suppose the IF creates leaves for a subregion in which not enough data points are prevalent, leading to leaves separating the state space in regard to P_{el} but not for T_{amb}, resulting in the failure of restricting the extrapolation boundary in the direction of T_{amb}. This result leads to the conclusion that IF is not robust against extrapolation in itself. Thus this is an illustrative example of why *robustness against extrapolation in itself* is a requirement for extrapolation detection algorithms, as introduced in Section 2. Due to this unpredictable behavior, we deem IFs unsuitable for extrapolation detection. All other algorithms perform similarly well, wrapping around the known data points in a similar distance like the validity domain. kNN and KDE form rather sharp envelopes, even with bubbles and holes, avoiding overestimating the validity domain. While GPR and OCSVM form rather dull envelopes, avoiding underestimating the validity domain. After all, these results indicate that the proposed methodology, used with suitable novelty detection algorithms, leads to a good extrapolation boundary, mimicking the reasonable borders of the validity domain.

6.3. Multi-dimensional analysis and scoring

Fig. 5 quantifies the results for all use cases by calculating the F-score on the respective evaluation data set. First, the results for the realistic tuning, then the ideal tuning, and then the results of the IF algorithm are discussed.

With *realistic tuning*, all algorithms perform similarly well per use case, with kNN and GPR performing slightly better than the rest, indicating that the extrapolation detection quality depends more on the use case and data than on the choice of algorithm. There is a big gap between the accuracy of the two-dimensional use case and the multi-dimensional ones, tentatively indicating that more complex use cases, e.g., higher dimension, more irrelevant variables, more significant uncertainty within the system, and more complex envelope shapes, are more challenging to perform extrapolation detection on. Surprisingly, the kNN algorithm, which is considered to be particularly susceptible to the curse of dimensionality [21], achieves the most accurate results even for the multi-dimensional use cases. This confirms the reasoning made in Section 5.1, that the curse of dimensionality is not yet considerably applying in the range of the typical number of dimensions.

As expected, *ideal tuning* leads to better results than realistic tuning. Multi-dimensional use cases seem more difficult to tune as the improvement through the ideal tuning is greater. Still not obtaining a reliable trend, as the P_{el} use case is of lower dimension (4 features) than the ΔT_{zone} (11 features) use case but still provides the most significant improvement. With ideal tuning, the performance of all algorithms is even more similar, leading to the belief that the quality of tuning is

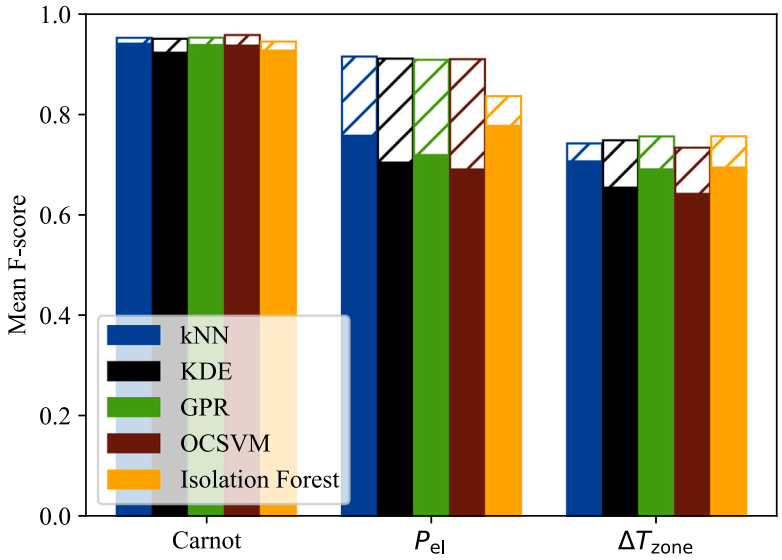


Fig. 5. Mean F-scores of all use cases and algorithms. The filled bars are for the realistically tuned and the hatched for the ideally tuned algorithms.

more important than the choice of algorithm. It also indicates that algorithms like kNN and IF are less sensitive to their hyperparameters than, e.g., KDE or OCSVM due to their rather good performance without ideal tuning. This can be interpreted as an advantage, as their hyperparameters can be set without expert knowledge of the system to be modeled.

The *IF algorithm* obtains high F-scores on all use cases. From the visualization of the two-dimensional use case, we know that IF behaved unreliably. As IF still achieves a high F-score in that use case, we remark that the F-score cannot be considered an ideal measure of accuracy. Though, at the current state of research, we deem it the best option available. In further research, we want to analyze whether combining different measures or using the $F_\beta - score$ can be a more efficacious error metric.

Overall, kNN seems a performant and reliable choice. Also, GPR, being able to be the ML model and extrapolation detector at once, shows good accuracy. However, the problem remains that the calculation time increases with the number of data points and can quickly reach the limits of the calculation hardware, as shown in Section 5.1.

7. Conclusion

This paper introduces a novel methodology of tuning novelty detectors towards the ML model for their application as extrapolation detectors. Through an expounding literature review and benchmark test of 15 different novelty detection algorithms, five suitable algorithms are selected for further analysis. The benchmark shows that there are algorithms performing much better than others. It also shows that most algorithms perform training and inference quickly enough to be considered for most applications of data driven model predictive control and predictive maintenance in building energy systems. Though, some algorithms become computationally infeasible with larger data sets, e.g., Gaussian Process Regression through RAM overflow. The discussion leads to the following subset of suitable algorithms: k Nearest Neighbors, One-Class Support Vector Machine, Gaussian Process Regression, Kernel-based Density Estimation, and Isolation Forest.

These algorithms are evaluated on building energy systems. First, on two-dimensional data, visualized using a novel visualization scheme, then on more complex multi-dimensional use cases. The results indicate that the proposed methodology is capable of approximating the validity domain. Obtaining very high accuracy for the 2-dimensional use case and lower but still good accuracy for the multi-dimensional use cases. The novel visualization scheme enables a profound analysis, which leads to the proposal of a sharpened definition for the validity domain and the extrapolation boundary. The visualization scheme also reveals that one algorithm (Isolation Forest) is unreliable for extrapolation detection. An insight that could not have been drawn from the F-score alone.

Through this analysis, for the remaining algorithms, it is found that the selection of hyperparameters and the use case itself have a more significant impact on the quality of the extrapolation boundary than the choice of algorithm. It is advantageous to have an algorithm that can be tuned without requiring expert knowledge about the modeled system. Therefore, when selecting an algorithm, it is advisable to choose one that is not overly sensitive to its hyperparameter settings. With k Nearest Neighbors being one of the most accurate, computationally cheapest, and quite robust against bad hyperparameter settings, we deem it a reasonable choice if nothing indicates else wise.

All findings have been drawn with caution, bearing in mind the broad variety of data and the resulting variety of algorithm behavior. Still, it would be beneficial to justify and extend these findings on a large variety of data sets with different identified properties, e.g., analyzing the influence of dimensionality, uncertainty, length and position of training and validation data, the ML-model and the shape of its validity domain, etc.

In addition to the direct outlook of this work, we identified several more far-reaching topics that we deem valuable to pursue:

- Quantification of the degree of system state exploration in a building energy system data set, i.e., how good is the coverage with available measurement data compared to the possible operation states.
- Quantification of the extrapolation capability of an ML model of arbitrary type, i.e., how likely is the ML model to perform well in extrapolation areas.
- Improvement of the ML model's extrapolation capability, either through specialized tuning pipelines or through physics-informed ML models.
- Employing extrapolation detectors for the safe operation of data driven model predictive control. We refer to our follow-up paper from Stoffel et al. where a fallback controller is used when extrapolation is detected [82].

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.

Acknowledgments

We gratefully acknowledge the financial support by the Federal Ministry for Economic Affairs and Climate Action (BMWK), promotional reference 03EN1066A and 03EN3060D. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101023666.

References

[1] United Nations Environment Programme. 2020 Global status report for buildings and construction: Towards a zero-emission, efficient and resilient buildings and construction sektor. Nairobi; 2020.

[2] Kathirgamanathan A, de Rosa M, Mangina E, Finn DP. Data-driven predictive control for unlocking building energy flexibility: A review. Renew Sustain Energy Rev 2021;135:110120. http://dx.doi.org/10.1016/j.rser.2020.110120, URL http://www.sciencedirect.com/science/article/pii/S1364032120304111.

[3] Bünning F, Schalbetter A, Aboudonia A, de Badyn MH, Heer P, Lygeros J. Input convex neural networks for building MPC. 2020, arXiv:2011.13227 [cs, eess].

[4] Jain A, Smarra F, Reticcioli E, D'Innocenzo A, Morari M. NeurOpt: Neural network based optimization for building energy management and climate control. 2020, arXiv:2001.07831 [cs, eess].

[5] Stoffel P, Maier L, Kümpel A, Schreiber T, Müller D. Evaluation of advanced control strategies for building energy systems. Energy Build 2023;280:112709. http://dx.doi.org/10.1016/j.enbuild.2022.112709.

[6] Schweidtmann AM, Weber JM, Wende C, Netze L, Mitsos A. Obey validity limits of data-driven models through topological data analysis and one-class classification. Opt Eng 2021. http://dx.doi.org/10.1007/s11081-021-09608-0.

[7] Leonard JA, Kramer MA, Ungar LH. A neural network architecture that computes its own reliability. Comput Chem Eng 1992;16(9):819–35. http://dx.doi.org/10.1016/0098-1354(92)80035-8, URL https://linkinghub.elsevier.com/retrieve/pii/0098135492800358.

[8] Courrieu P. Three algorithms for estimating the domain of validity of feedforward neural networks. Neural Netw 1994;7(1):169–74. http://dx.doi.org/10.1016/0893-6080(94)90065-5, URL https://linkinghub.elsevier.com/retrieve/pii/0893608094900655.

[9] Brooks DG, Carroll SS, Verdini WA. Characterizing the domain of a regression model. The American Statistician 1988;42(3):187–90. http://dx.doi.org/10.2307/2684998, https://www.jstor.org/stable/2684998.

[10] Hooker G. Diagnostics and extrapolation in machine learning [Ph.D. thesis], Stanford University; 2004.

[11] Hüllermeier E, Waegeman W. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach Learn 2021;110(3):457–506. http://dx.doi.org/10.1007/s10994-021-05946-3.

[12] Hora SC. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. Reliab Eng Syst Saf 1996;54(2–3):217–23. http://dx.doi.org/10.1016/S0951-8320(96)00077-4.

[13] Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Inf Fusion 2021;76:243–97. http://dx.doi.org/10.1016/j.inffus.2021.05.008, URL https://www.sciencedirect.com/science/article/pii/S1566253521001081.

[14] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International conference on machine learning. 2016, p. 1050–9.

[15] Pearce T, Zaki M, Neely AD. Bayesian neural network ensembles. 2018, arXiv, arXiv:1811.12188.

[16] Pelillo M. A relaxation algorithm for estimating the domain of validity of feedforward neural networks. Neural Process Lett 1996;3(3):113–21. http://dx.doi.org/10.1007/BF00420280.

[17] Sluijterman L, Cator E, Heskes T. How to evaluate uncertainty estimates in machine learning for regression? 2021, URL http://arxiv.org/pdf/2106.03395v1.

[18] Gal Y. Uncertainty in deep learning [Ph.D. thesis], University of Cambridge; 2016.

[19] Aggarwal CC. Outlier analysis. Springer ebook collection computer science, 2nd ed. Cham: Springer, Springer International Publishing; 2017, http://dx.doi.org/10.1007/978-3-319-47578-3.

[20] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. ACM Comput Surv 2009;41(3):1–58. http://dx.doi.org/10.1145/1541880.1541882, URL https://dl.acm.org/doi/10.1145/1541880.1541882.

[21] Pimentel MA, Clifton DA, Clifton L, Tarassenko L. A review of novelty detection. Signal Process 2014;99:215–49. http://dx.doi.org/10.1016/j.sigpro.2013.12.026, URL https://www.sciencedirect.com/science/article/pii/S016516841300515X.

[22] van Every PM, Rodriguez M, Jones CB, Mammoli AA, Martínez-Ramón M. Advanced detection of HVAC faults using unsupervised SVM novelty detection and Gaussian process models. Energy Build 2017;149:216–24. http://dx.doi.org/10.1016/j.enbuild.2017.05.053, URL https://linkinghub.elsevier.com/retrieve/pii/S0378778816312592.

[23] Khan SS, Madden MG. One-class classification: taxonomy of study and review of techniques. Knowl Eng Rev 2014;29(3):345–74. http://dx.doi.org/10.1017/S026988891300043X, URL https://www.cambridge.org/core/product/identifier/S026988891300043X/type/journal_article.

[24] Goldstein M, Uchida S, Zhu D. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PLoS One 2016;11(4):e0152173. http://dx.doi.org/10.1371/journal.pone.0152173, URL https://dx.plos.org/10.1371/journal.pone.0152173.

[25] Tax DMJ, Duin RPW. Combining one-class classifiers. In: Kittler J, Roli F, Goos G, Hartmanis J, van Leeuwen J, editors. Multiple classifier systems, vol. 2096. Berlin, Heidelberg: Springer Berlin Heidelberg; 2001, p. 299–308. http://dx.doi.org/10.1007/3-540-48219-9_30, URL http://link.springer.com/10.1007/3-540-48219-9_30.

[26] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. ACM SIGMOD Rec 2000;29(2):427–38. http://dx.doi.org/10.1145/335191.335437, URL https://dl.acm.org/doi/10.1145/335191.335437.

[27] Kriegel H-P, S. hubert M, Zimek A. Angle-based outlier detection in high-dimensional data. In: Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining. Las Vegas, Nevada, USA: ACM Press; 2008, p. 444. http://dx.doi.org/10.1145/1401890.1401946, URL http://dl.acm.org/citation.cfm?doid=1401890.1401946.

[28] Bellman RE. Adaptive control processes: A guided tour. Princeton University Press; 1961, http://dx.doi.org/10.1515/9781400874668, URL https://www.degruyter.com/document/doi/10.1515/9781400874668/html.

[29] Ding X, Li Y, Belatreche A, Maguire LP. An experimental evaluation of novelty detection methods. Neurocomputing 2014;135:313–27. http://dx.doi.org/10.1016/j.neucom.2013.12.002, URL https://linkinghub.elsevier.com/retrieve/pii/S0925231213011314.

[30] He Z, Xu X, Deng S. Discovering cluster-based local outliers. Pattern Recognit Lett 2003;24(9–10):1641–50. http://dx.doi.org/10.1016/S0167-8655(03)00003-5, URL https://linkinghub.elsevier.com/retrieve/pii/S0167865503000035.

[31] Breunig MM, Kriegel H-P, Ng RT, Sander J. LOF: identifying density-based local outliers. ACM SIGMOD Rec 2000;29(2):93–104. http://dx.doi.org/10.1145/335191.335388, URL https://dl.acm.org/doi/10.1145/335191.335388.

[32] Li Z, Zhao Y, Hu X, Botta N, Ionescu C, Chen GH. ECOD: Unsupervised outlier detection using empirical cumulative distribution functions. 2022, http://dx.doi.org/10.48550/ARXIV.2201.00382.

[33] Hardin J, Rocke DM. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. Comput Stat Data Anal 2004;44(4):625–38. http://dx.doi.org/10.1016/S0167-9473(02)00280-3, URL https://linkinghub.elsevier.com/retrieve/pii/S0167947302002803.

[34] Goldstein M, Dengel A. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In: KI-2012: poster and demo track, vol. 9. 2012.

[35] Parzen E. On estimation of a probability density function and mode. Ann Math Stat 1962;33(3):1065–76. http://dx.doi.org/10.1214/aoms/1177704472, URL http://projecteuclid.org/euclid.aoms/1177704472.

[36] Bishop CM. Novelty detection and neural network validation. IEE Proc Vis Imag Signal Process 1994;141(4):217. http://dx.doi.org/10.1049/ip-vis:19941330, URL https://digital-library.theiet.org/content/journals/10.1049/ip-vis_19941330.

[37] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Adaptive computation and machine learning, Cambridge, Mass: MIT Press; 2006.

[38] Bishop CM. Pattern recognition and machine learning. Information science and statistics, New York: Springer; 2006.

[39] Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Neural Comput 2001;13(7):1443–71. http://dx.doi.org/10.1162/089976601750264965, URL https://direct.mit.edu/neco/article/13/7/1443-1471/6529.

[40] Cortes C, Vapnik V. Support-vector networks. Mach Learn 1995;20(3):273–97. http://dx.doi.org/10.1007/BF00994018, URL http://link.springer.com/10.1007/BF00994018.

[41] Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui SA, Binder A, et al. Deep one-class classification. In: Dy J, Krause A, editors. Proceedings of the 35th international conference on machine learning. Proceedings of machine learning research, vol. 80, PMLR; 2018, p. 4393–402, URL https://proceedings.mlr.press/v80/ruff18a.html.

[42] Hawkins S, He H, Williams G, Baxter R. Outlier detection using replicator neural networks. In: Kambayashi Y, Winiwarter W, Arikawa M, Goos G, Hartmanis J, van Leeuwen J, editors. Data warehousing and knowledge discovery, vol. 2454. Berlin, Heidelberg: Springer Berlin Heidelberg; 2002, p. 170–80. http://dx.doi.org/10.1007/3-540-46145-0_17, URL http://link.springer.com/10.1007/3-540-46145-0_17.

[43] Shyu M-L, Chen S-C, Sarinnapakorn K, Chang L. A novel anomaly detection scheme based on principal component classifier. Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering; 2003.

[44] Sakurada M, Yairi T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis. Gold Coast, Australia QLD, Australia: ACM Press; 2014, p. 4–11. http://dx.doi.org/10.1145/2689746.2689747, URL http://dl.acm.org/citation.cfm?doid=2689746.2689747.

[45] Li M, Vitányi PMB. An introduction to kolmogorov complexity and its applications. Texts in computer science, 4th ed. New York: Springer; 2019 http://dx.doi.org/10.1007/978-3-030-11298-1,.

[46] Zimek A, Campello RJ, Sander J. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. ACM SIGKDD Explor Newsl 2014;15(1):11–22. http://dx.doi.org/10.1145/2594473.2594476, URL https://dl.acm.org/doi/10.1145/2594473.2594476.

[47] Lazarevic A, Kumar V. Feature bagging for outlier detection. In: Proceeding of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining. Chicago, Illinois, USA: ACM Press; 2005, p. 157. http://dx.doi.org/10.1145/1081870.1081891, URL http://portal.acm.org/citation.cfm?doid=1081870.1081891.

[48] Liu FT, Ting KM, Zhou Z-H. Isolation forest. In: 2008 Eighth IEEE international conference on data mining. Pisa, Italy: IEEE; 2008, p. 413–22. http://dx.doi.org/10.1109/ICDM.2008.17, URL http://ieeexplore.ieee.org/document/4781136/.

[49] Simutis R, Havlik I, Schneider F, Dors M, Lübbert A. Artificial neural networks of improved reliability for industrial process supervision. IFAC Proc Vol 1995;28(3):59–65. http://dx.doi.org/10.1016/S1474-6670(17)45602-3.

[50] Lee JM, Lee JH. Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes. Automatica 2005;41(7):1281–8. http://dx.doi.org/10.1016/j.automatica.2005.02.006, URL https://linkinghub.elsevier.com/retrieve/pii/S0005109805000786.

[51] Teixeira AP, Clemente JJ, Cunha AE, Carrondo M, Oliveira R. Bioprocess iterative batch-to-batch optimization based on hybrid parametric/nonparametric models. Biotechnol Prog 2006;22(1):247–58. http://dx.doi.org/10.1021/bp0502328, URL http://doi.wiley.com/10.1021/bp0502328.

[52] Rall D, Menne D, Schweidtmann AM, Kamp J, von Kolzenberg L, Mitsos A, et al. Rational design of ion separation membranes. J Membr Sci 2019;569:209–19. http://dx.doi.org/10.1016/j.memsci.2018.10.013, URL https://linkinghub.elsevier.com/retrieve/pii/S0376738818324293.

[53] Bae J, Lee Hj, Jeong DH, Lee JM. Construction of a valid domain for a hybrid model and its application to dynamic optimization with controlled exploration. Ind Eng Chem Res 2020;59(37):16380–95. http://dx.doi.org/10.1021/acs.iecr.0c02720, URL https://pubs.acs.org/doi/10.1021/acs.iecr.0c02720.

[54] Pineda LR, Serpa AL. Determination of confidence bounds and artificial neural networks in non-linear optimization problems. Neurocomputing 2021;463:495–504. http://dx.doi.org/10.1016/j.neucom.2021.08.075, URL https://linkinghub.elsevier.com/retrieve/pii/S0925231221012650.

[55] Malak RJ, Paredis CJJ. Using support vector machines to formalize the valid input domain of models in data-driven predictive modeling for systems design. In: Proceedings of the ASME international design engineering technical conferences and computers and information in engineering conference - 2009. New York, NY: ASME; 2010, p. 1423–36. http://dx.doi.org/10.1115/DETC2009-87376.

[56] Roach E, Parker RR, Malak RJ. An improved support vector domain description method for modeling valid search domains in engineering design problems. In: Proceedings of the ASME international design engineering technical conferences and computers and information in engineering conference - 2011. New York, NY: ASME; 2012, p. 741–51. http://dx.doi.org/10.1115/DETC2011-48435.

[57] Maddalena ET, Muller SA, Santos RMd, Salzmann C, Jones CN. Experimental data-driven model predictive control of a hospital HVAC system during regular use. 2021, http://dx.doi.org/10.48550/ARXIV.2112.07323, arXiv preprint arXiv:2112.07323.

[58] Jain A, Nghiem T, Morari M, Mangharam R. Learning and control using Gaussian processes. In: 2018 ACM/IEEE 9th international conference on cyber-physical systems. Porto: IEEE; 2018, p. 140–9. http://dx.doi.org/10.1109/ICCPS.2018.00022, URL https://ieeexplore.ieee.org/document/8443729/.

[59] Nghiem TX, Jones CN. Data-driven demand response modeling and control of buildings with Gaussian processes. In: 2017 American control conference. Seattle, WA, USA: IEEE; 2017, p. 2919–24. http://dx.doi.org/10.23919/ACC.2017.7963394, URL https://ieeexplore.ieee.org/document/7963394/.

[60] Galy-Fajou T, Opper M. Adaptive inducing points selection for Gaussian processes. 2021, URL http://arxiv.org/pdf/2107.10066v1.

[61] Quiñonero-Candela J, Rasmussen CE. A unifying view of sparse approximate Gaussian process regression. J Mach Learn Res 2005;6(65):1939–59, URL http://jmlr.org/papers/v6/quinonero-candela05a.html.

[62] Psaros AF, Meng X, Zou Z, Guo L, Karniadakis GE. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. J Comput Phys 2023;477:111902. http://dx.doi.org/10.1016/j.jcp.2022.111902, URL https://www.sciencedirect.com/science/article/pii/S0021999122009652.

[63] Manokhin V. Machine learning for probabilistic prediction [Ph.D. thesis], Zenodo; 2022, http://dx.doi.org/10.5281/ZENODO.6727505.

[64] Han S, Hu X, Huang H, Jiang M, Zhao Y. ADBench: Anomaly detection benchmark. 2022, http://dx.doi.org/10.48550/ARXIV.2206.09426.

[65] Markou M, Singh S. Novelty detection: a review—part 1: statistical approaches. Signal Process 2003;83(12):2481–97. http://dx.doi.org/10.1016/j.sigpro.2003.07.018.

[66] Miljkovic D. Review of novelty detection methods. In: The 33rd international convention MIPRO. 2010, p. 593–8.

[67] Dreiseitl S, Osl M, Scheibböck C, Binder M. Outlier detection with one-class SVMs: An application to melanoma prognosis. In: AMIA annual symposium proceedings, vol. 2010. 2010, p. 172–6.

[68] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. 2015, p. 448–56. http://dx.doi.org/10.48550/ARXIV.1502.03167.

[69] Chollet F, et al. Keras. 2015, URL.

[70] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, http://dx.doi.org/10.48550/ARXIV.1412.6980.

[71] Erickson N, Mueller J, Shirkov A, Zhang H, Larroy P, Li M, et al. AutoGluon-tabular: Robust and accurate AutoML for structured data. 2020, arXiv preprint arXiv:2003.06505 [in Citavi anzeigen].

[72] Rätz M, Javadi AP, Baranski M, Finkbeiner K, Müller D. Automated data-driven modeling of building energy systems via machine learning algorithms. Energy Build 2019;202:109384. http://dx.doi.org/10.1016/j.enbuild.2019.109384.

[73] Meisenbacher S, Turowski M, Phipps K, Rätz M, Müller D, Hagenmeyer V, et al. Review of automated time series forecasting pipelines. WIREs Data Min Knowl Discov 2022;12(6). http://dx.doi.org/10.1002/widm.1475.

[74] Zhao Y, Nasrullah Z, Li Z. PyOD: A python toolbox for scalable outlier detection. 2019, arXiv:1901.01588 [cs, stat].

[75] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. 2012, http://dx.doi.org/10.48550/ARXIV.1201.0490.

[76] Bergstra J, Yamins D, Cox D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: Dasgupta S, McAllester D, editors. Proceedings of the 30th international conference on machine learning. Proceedings of machine learning research, vol. 28, Atlanta, Georgia, USA: PMLR; 2013, p. 115–23, URL https://proceedings.mlr.press/v28/bergstra13.html.

[77] Rayana S. ODDS library. Stony Brook University, Department of Computer Sciences; 2016, URL http://odds.cs.stonybrook.edu.

[78] Stoffel P, Berktold M, Kümpel A, Müller D. An online learning approach for data-driven model predictive control in building energy systems. In: Proceedings of ECOS 2022 - the 35th international conference on efficiency, cost, optimization, simulation and environmental impact of energy systems. 2022, http://dx.doi.org/10.11581/dtu.00000267.

[79] Zhang L. Data-driven building energy modeling with feature selection and active learning for data predictive control. Energy Build 2021;252:111436. http://dx.doi.org/10.1016/j.enbuild.2021.111436.

[80] Blum D, Arroyo J, Huang S, Drgoňa J, Jorissen F, Walnum HT, et al. Building optimization testing framework (BOPTEST) for simulation-based benchmarking of control strategies in buildings. J Build Perform Simul 2021;14(5):586–610. http://dx.doi.org/10.1080/19401493.2021.1986574, URL https://www.tandfonline.com/doi/full/10.1080/19401493.2021.1986574.

[81] Judkoff R, Neymark J. International energy agency building energy simulation test (BESTEST) and diagnostic method, no. NREL/TP–472-6231, 90674. 1995, http://dx.doi.org/10.2172/90674, URL http://www.osti.gov/servlets/purl/90674/.

[82] Stoffel P, Henkel P, Rätz M, Kümpel A, Müller D. Safe operation of online learning data driven model predictive control of building energy systems. Energy AI 2023;14:100296. http://dx.doi.org/10.1016/j.egyai.2023.100296.