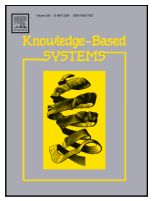
Knowledge-Based Systems 330 (2025) 114599

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Privacy-preserving ground-truth data for evaluating additive feature attribution in regression models with additive CBR and CQV

Mir Riyanul Islam [a,*], Rosina O. Weber [b], Mobyen Uddin Ahmed [a], Shahina Begum [a]

[a] School of Innovation, Design and Engineering, Mälardalen University, Västerås 72220, Västmanland, Sweden
[b] Drexel University, Philadelphia, PA, 19802, USA

ARTICLE INFO

Keywords:
Additive CBR
Additive feature attribution
CQV
Ground-truth evaluation
XAI

ABSTRACT

Explainable artificial intelligence (XAI) methods produce information outputs based on a target artificial intelligence model to be explained. The most popular information output is produced by XAI methods of the category feature attribution, which produce the relative contribution of each input feature in a local instance. These relative contributions indicate how important each input feature is in a decision; this type of information is expected to provide explanatory value to users. In real-world regression tasks, feature attribution methods are crucial for comprehending model predictions. However, robust evaluation of such methods remains challenging due to a lack of ground-truth data and widely accepted evaluation metrics, such as accuracy for classification or mean absolute error for regression. This paper proposes a novel approach for generating synthetic, privacy-preserving ground-truth datasets for regression problems that retain original feature behaviour, enabling rigorous feature attribution evaluation without compromising sensitive information. We introduce additive case-based reasoning (AddCBR) as a model-aligned and interpretable baseline to benchmark additive feature attribution methods. This work also demonstrates the first use of the coefficient of quartile variation (CQV) as a statistical measure to quantify the consistency and stability of feature attribution methods. Altogether, these contributions form a comprehensive evaluation methodology for objectively assessing and comparing feature attribution methods in regression models. By providing a controlled evaluation pipeline with reliable baselines and metrics, this work addresses the current lack of consensus and benchmarking in XAI evaluation for regression models.

1. Introduction

Recent literature on explainable artificial intelligence (XAI) explores how artificial intelligence (AI) algorithms can explain their decisions [1–6]. In essence, XAI methods attempt to extract information beyond an AI algorithm's raw output (e.g., a predicted class or numeric outcome) by revealing *why* the model made its decision–for example, by identifying the features that most influenced a given prediction. Inherently interpretable models, like decision trees and case-based reasoning (CBR), often offer information like tree paths [7] or cases [8,9] that may carry explanatory value. However, when AI models are either not interpretable or not sufficiently interpretable, post-hoc XAI methods can extract additional information. Among such methods, additive feature attribution methods such as SHAP [10] and LIME [11] assign each input feature a contribution value whose sum approximates the model's output. This characteristic allows what is termed *local accuracy* [10] that indicates how closely the additive feature attribution method's output aligns with the AI model's prediction for each instance. However, there is ongoing debate about whether *local accuracy* alone is an adequate measure for quantitative quality of feature attribution [12,13].

Despite the popularity of feature attribution methods, there is a lack of widely accepted evaluation standards for them [6,14–16]. Constructing reliable ground-truth benchmarks for evaluation remains a significant challenge [17–21], and it is recommended that XAI methods must be evaluated in each domain and application, because it is consensual that explanations are application-specific and contextual (e.g., dependent on user, domain, and task) [19,22–26]. In practice, these factors make XAI evaluation challenging, and such evaluations are frequently neglected entirely [6,16]. In a recent scoping review, Mainali and Weber [16] found that 81 % of works describing machine learning applications as *explainable* do not evaluate the quantitative quality of the information outputs produced by their XAI methods.

Much of the existing work on feature attribution methods has focused on classification tasks. Applying similar methods to regression

* Corresponding author.
E-mail addresses: mir.riyanul.islam@mdu.se (M.R. Islam), rosina@drexel.edu (R.O. Weber), mobyen.uddin.ahmed@mdu.se (M.U. Ahmed), shahina.begum@mdu.se (S. Begum).

https://doi.org/10.1016/j.knosys.2025.114599
Received 26 June 2025; Received in revised form 30 August 2025; Accepted 1 October 2025
Available online 6 October 2025

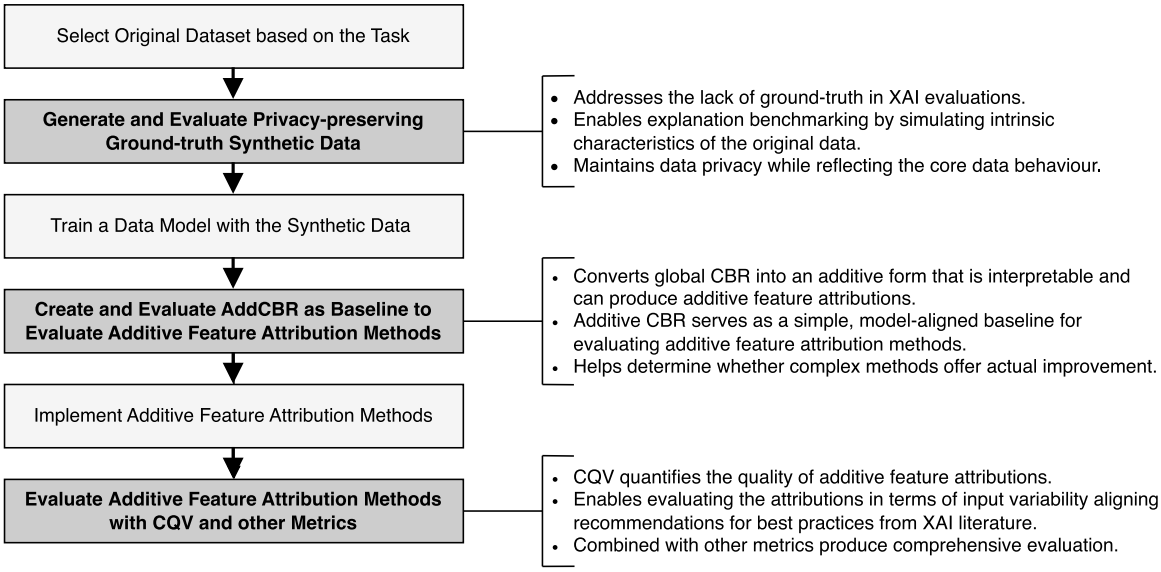


Fig. 1. Overview of the proposed pipeline to evaluate additive feature attributions for regression. The innovations of this work are highlighted in blocks of darker shade and bold fonts. Also, the rationales of the innovations are outlined on the right.

problems requires careful adaptation [27]. Particularly, to ensure the model's output is preserved on the original scale of the data-referred to as the *property of conservation*-is essential, but in practice this can be easily violated by steps including normalization or standardization. Additionally, literature indicates that the outputs of popular additive feature attribution methods can be highly sensitive to the choice of data models and to feature collinearity in the data [28]. To address all the aforementioned challenges, the overarching goal of this work is to establish a privacy-preserving, model-aligned, and statistically grounded framework for evaluating additive feature attribution methods in regression models with the innovations outlined in the following subsection.

1.1. Innovations

This work introduces three key innovations that advance the state-of-the-art in XAI for regression tasks:

1. *Privacy-preserving ground-truth data for evaluating feature attribution in regression models:* We propose a novel approach to create synthetic datasets for regression tasks, which address privacy concerns while preserving the explanatory role of each feature. The synthetic dataset replicates the characteristics of the original dataset and can be used to compute feature attributions as ground-truth to evaluate feature attribution methods for explaining regression tasks. This approach aligns with recent recommendations to use synthetic data for benchmarking XAI techniques [29] and protect sensitive information from the original dataset [30,31].
2. *Additive form of CBR as a model-aligned baseline to evaluate additive feature attribution methods:* We extend our additive CBR (AddCBR) method, first introduced in a workshop paper by the same authors [32]. Specifically, AddCBR is designed to serve not only as an interpretable explanation method but also as a model-aligned baseline for evaluating additive feature attribution methods. The benefit of AddCBR is that it uses the prediction model parameters to generate feature attributions formatted in the same way as additive feature attribution methods like SHAP [10] or LIME [11] that we aim to evaluate. AddCBR offers a transparent, feature attribution-aware reference for the evaluation of additive feature attribution methods. Additionally, in this work, we validate AddCBR's reliability with a feature deletion study that exhibits how removing each feature affects the model's predictions, confirming that feature attribution from AddCBR truly reflects each feature's influence.
3. *Statistical metric for evaluating the additive feature attribution methods:* We propose the use of a statistical measure, the coefficient of quartile variation (CQV) [33], to evaluate the quantitative quality and consistency of feature attributions. By comparing the variability of two sets of attribution values (for instance, the attributions from our baseline versus those from another method), CQV yields a single quantitative indicator of how similar or stable the two sets are. To the best of our knowledge, this is the first application of CQV in evaluating XAI methods, offering a novel way to determine whether different explanation outputs are in agreement.

To present the value of the outlined innovations, we implement an evaluation pipeline for the additive feature attribution methods. Despite being potentially applicable to both regression and classification application tasks; in this work, the implementation of the proposed approach focuses only on regression problems. Fig. 1 provides an overview of our evaluation pipeline for additive feature attribution methods in regression. The workflow starts with the acquisition of an original dataset that is used to create the model whose decisions one needs to explain. From the original data, we generate a synthetic dataset that captures the behaviours within the original dataset, thus preserving the privacy and reflecting ground-truth behaviours. A new data model is then trained on this synthetic data. Next, we derive the AddCBR baseline feature attributions from the synthetic data model and examine their fidelity. After establishing this baseline, we apply the chosen additive feature attribution methods to the data model. Finally, the baseline is used to evaluate the outcome of the additive feature attribution methods using the CQV metric and other established measures from the literature. In our experiments, we consider two popular additive feature attribution methods. The first method is SHAP [10] that is grounded in Shapley values [34] and is expected to perform well for regression tasks. And, LIME [11]—the other method that we anticipate to be less effective for regression tasks because it includes some standardization of the model's values to produce feature attribution.

1.2. Motivation

This work is motivated by a real-world regression problem in aviation: predicting flight take-off time delays. The aviation industry incurs, on average, approximately 100 Euros per minute for the Air Traffic Flow Management [35]. In the United States, a 2019 Federal Aviation

Administration[1] report presented that the estimated cost due to delay, considering passengers, airlines, lost demand, and indirect costs, was 33 billion dollars [36]. Such high stakes underscore the significance and provide the rationale for increased attention towards predicting take-off time and reducing delays [37]. More generally, regression tasks play a crucial role in many AI applications, yet the explainability of regression outcomes has received relatively limited attention compared to that of classification tasks.

A key challenge in developing and evaluating regression models in safety-critical domains like aviation is the limited availability of high-quality data. Original datasets are often proprietary, domain-specific, and computationally expensive to process due to high dimensionality [27,29]. These constraints hinder the iterative experimentation needed to improve prediction and explanation methods. Moreover, a fundamental issue in the evaluation of feature attribution methods is the absence of ground-truth attributions [21]. When combined with the need to preserve data privacy and the importance of maintaining the output in its original measurement unit, i.e., *property of conservation*, these challenges motivate our use of privacy-preserving synthetic data as a stand-in for ground-truth in explanation evaluation. Furthermore, given the lack of consensus on standardized evaluation criteria for XAI methods [6], we present AddCBR as a model-aligned baseline and use the CQV metric to provide a quantitative assessment of the consistency of different feature attribution methods. To validate the utility of the proposed benchmark and evaluation framework, we focus our analysis on a single, domain-specific dataset. This allows for a detailed investigation of the underlying features and their role in supporting meaningful evaluation of XAI methods in a real-world regression context.

The remainder of this article is organised as follows–Section 2 reviews relevant background and related work. Section 3 introduces the proposed approach for generating privacy-preserving ground-truth data and presents the evaluation of the generated data with respect to the original dataset. Section 4 describes the construction of the AddCBR baseline for evaluating additive feature attribution methods and presents the experiments demonstrating its effectiveness as an evaluation benchmark. Section 5 outlines the proposed evaluation criteria for the additive feature attribution methods and discusses the implementation with experimental results. Finally, Section 6 concludes the paper with a summary of the findings and potential future directions for advancing the research field of XAI in the context of evaluation.

2. Background and related work

This section presents the formal definitions of the regression models and the additive feature attribution methods used to explain their output, which are referred to in the subsequent sections. Following the definitions, we describe the state-of-the-art XAI methods used in this study, along with their evaluation approaches. We also review the works utilizing benchmark datasets for XAI evaluation.

2.1. Formal definition

The regression model Ω is defined for a dataset of n observations indexed by $i \in \{1, \dots, n\}$. The ith observation is described by a set of m independent features or attributes $a_1, \dots, a_m$, represented by the vector $x_i = [x_{i1}, \dots, x_{im}] \in \mathcal{X}$, where, x_{ij} is the value of attribute a_j drawn from its distribution D_j, where $j \in \{1, \dots, m\}$. This distribution can be continuous or discrete, depending on the nature of the attributes. The feature space is defined as $\mathcal{A} = D_1 \times \dots \times D_m$. The corresponding target value is $y_i \in \mathcal{Y} \subseteq \mathbb{R}$. Given the dataset, the objective of Ω is to learn a mapping function $r : \mathcal{X} \to \mathcal{Y}$ that accurately estimates the target variable y_i from the input feature vector x_i, i.e., $r(x_i) = \hat{y}_i$. Finally, an explanation problem is a tuple $(r, (x_i, y_i))$ intended to be solved using an explanation function $g(z_i)$. Here, $z_i \in \{0, 1\}^m$ simplified binary representation of x_i obtained via a transformation function $h(x_i)$. The function $g(z_i)$ computes the feature attributions ($\phi \in \mathbb{R}^m$) using Eq. (1) [10]:

$$g(z_i) = \phi_0 + \sum_{j=1}^{m} \phi_j z_{ij} \tag{1}$$

where ϕ_0 corresponds to the bias term (average model output for the dataset) and ϕ_j attributes the effect of the jth feature on the prediction. Finally, the sum of all the feature attributions and the bias term approximates the output $r(x_i)$ of the regression model.

2.2. XAI methods

The purpose of explaining an AI model's decision is to make the behaviour of the model intelligible to users [1]. For this reason, many authors have stated that the explainability problem is user-, application-, and domain-specific [9,23,25]. This realization alone justifies the recommendation that XAI methods should be evaluated for each specific implementation. Particularly, when considering feature attribution methods, many studies have shown that their results can have several limitations (e.g., [21,38–40]), underscoring evaluation as a major requirement [21,39,41–44]. In this section, we describe the main XAI approaches evaluated in this work, as well as the approaches for evaluating explanations considering different perspectives.

The domain of XAI contains a wide spectrum of methods that can be categorised along various conceptual dimensions. A fundamental distinction between methods is whether they explain a model's overall decision strategy (i.e., global) [45,46] or an individual instance (i.e., local) [11,47]. Later in this article, we show that an interpretable CBR methodology yields a strong alignment between global and local feature attributions. By contrast, we did not observe this alignment with Extreme Gradient Boosting (XGBoost) or any other AI or XAI methods.

As introduced, we focus on the additive feature attribution methods for regression on tabular data. As an example of a method that is indicated for regression [27], we use SHAP [10], since it is based on Shapley values [34]. As we will validate, despite criticisms (e.g., [21,40]), SHAP performs well for the regression task.

SHAP–Shapley Additive Explanations [10] is a suite of methods for computing the relative contributions of individual features to a given prediction, so that their sum approximates the model's output. SHAP borrows concepts from cooperative game theory [48]. With non-linear black box models, SHAP provides feature importance values as a global explanation. It also produces local explanations for individual predictions using Shapley values [48] to fairly assign the impact among features. Because the calculation of the Shapley values requires iteration over 2^m sample space, SHAP approximates the feature contributions for high-dimensional data. For smaller values of m, the feature contributions are exact. SHAP computes the marginal contributions of the features to generate an explanation in the form of feature attribution for models' output. The marginal contribution of each feature is the difference between the prediction from the model with and without the respective feature. Finally, SHAP assigns each feature an overall contribution equal to its average marginal contribution across all possible feature combinations. SHAP is available as a Python tool[2]. It provides a dedicated *Explainer* implementation for text and image data. For tabular data, *KernelExplainer* is model-agnostic, and *TreeExplainer* is designed for tree-based models, both singular and ensembles. In this work, we use the *TreeExplainer* since the data model is built with XGBoost.

The second additive feature attribution method we consider is LIME–Local Interpretable Model-agnostic Explanations [11]. It is developed based on the assumption that the behaviour of an instance can be explained by fitting an interpretable model (e.g., linear regression) with a simplified representation of the instance and its closest neighbours. While making a single prediction, LIME first generates an interpretable

[1] https://www.faa.gov/

[2] https://shap.readthedocs.io

Table 1
Methods, metrics or axioms used for evaluating XAI methods with references to the works in which they were proposed or employed.

Evaluation Method/Axiom/Metric	References
Sensitivity analysis	[38]
Example images	[11]
Satisfiability/Model counting	[54–56]
Correlation, completeness and complexity	[57]
Conservation, continuity	[51]
Concept Activation Vectors (CAV)	[58]
Fidelity	[26,59–62]
Maximum Fidelity Gap	[60]
Effective Complexity	[63]
Gold features	[11]
Post-hoc accuracy	[64–67]
Perturbation analysis for vision	[68]
Remove and Retain (ROAR)	[50]
Retain and Debias (ROAD)	[69]
Random Logit Test	[70,71]
Perturbation on time-series	[72]
Implementation invariance, sensitivity	[73]
Input invariance	[74]
Simulated users	[11]
Amazon Mechanical Turk users	[11,64]
In-depth interviews	[75]

representation of the input instance. In doing so, LIME standardizes the values of the input features to create z, causing it to lose the original proportion of the feature values, which is important for regression [27]. In the next step, LIME perturbs the simplified input z and uses the black box model to predict on these perturbed samples, thus generating a training dataset for the interpretable model. LIME then draws samples from the generated data based on their similarity to select the closest neighbours. Lastly, an interpretable model (e.g., linear regression) is trained on these selected neighbouring samples. With the weights or coefficients corresponding to each feature from the trained model, LIME presents the local explanation. LIME is available as a Python package[3] capable of generating explanations for tabular, image, and text data.

2.3. Evaluation of XAI methods

In this section, we describe some evaluation methods from the literature that have been applied to different XAI methods, particularly to additive feature attribution methods. In a recent study, Zhou et al. [21] pointed out the fact that the main obstacle in evaluating feature attribution methods is the lack of ground-truth or ideal feature attribution values. To overcome this, Zhou et al. [21] proposed a dataset modification procedure to generate such ground-truth. In another study, the authors used a benchmark dataset as ground-truth for evaluating the explanations on the neural network outputs [49]. The literature in XAI presents a wide range of methods for evaluating the feature attribution methods, which are listed in Table 1. However, ground-truth datasets for XAI are not widely used, with a few notable exceptions (e.g., [17–21,49]), and such absence is recognized as limiting advances in XAI [50–53]. Notably, the use of gold features by Ribeiro et al. [11] was the closest form of ground-truth, i.e., the most important features used by the prediction model.

Nevertheless, different metrics and reference values have also been used to evaluate XAI methods. Liu et al. [29] conducted a comparative analysis of eight feature attribution methods for regression tasks. They evaluated these methods with various metrics (e.g., *faithfulness*, *monotonicity*, etc.), yielding valuable insights into their performance. Letzgus et al. [27] leveraged Shapley values [48] as a reference to evaluate their proposed method and highlighted the inconsistency of XGBoost's feature importance in local prediction scenarios. Troncoso-García et al. [76] used association rules to evaluate the explanations for time-series predictions, demonstrating evidence of LIME's inconsistency in generating explanations, which resembles the findings by Deng et al. [77].

To summarise, the efficacy of XAI methods should be based not only on their theoretical constructs but also on demonstrated empirical performance. Furthermore, there's a growing concern about whether these explanations can yield valuable insights and actionable decisions [78,79]. Finally, the majority of works applying machine learning models that claim they are explainable do so based on implementing popular libraries for additive feature attribution methods (e.g., SHAP [10], LIME [11]) without even questioning their validity or performing any evaluation [16]. In addition, none of the widely used metrics directly assesses the consistency or dispersion of feature attributions. This is an important omission, as high variance in an XAI method's output can undermine trust. To address this gap, we propose using the CQV to quantify the stability of feature attribution values, given that it is a robust measure of relative dispersion that is less sensitive to outliers than the regular coefficient of variation [33].

2.4. Benchmark datasets

Many authors agree that the lack of benchmark datasets to evaluate XAI methods is detrimental [50–53]. Jeyakumar et al. [80] used human labelling as benchmarks to evaluate several XAI methods against their newly proposed one across image, text, audio, and sensory datasets. In other studies, benchmarks were proposed for time-series classification [81,82] and for natural language tasks [83]. Amparore et al. [84] released a library providing several evaluation metrics for local linear explanation methods and presented its use by comparing SHAP [10] and LIME [11]. Particularly, several tools are developed for XAI evaluation with benchmark synthetic datasets in the recent years (e.g., BAM [52], XAI-Bench [29], OpenXAI [85], GraphXAI [86], $\mathcal{M}^4$ [87], XAI-Units [88], etc.). Among these tools, XAI-Bench [29], GraphXAI [86] and XAI-Units [88] used the synthetic datasets as ground-truths for evaluating different XAI methods. To this end, there remains a notable scarcity of synthetic datasets specifically designed for explanation benchmarking with ground-truth for regression tasks, let alone addressing the concerns with privacy regulations (e.g., GDPR [89]). These limitations directly motivates our first contribution, and to the best of our knowledge, no prior work has proposed to capture the underlying characteristics of the data as representatives of ground-truths while generating synthetic datasets as we describe in Section 3.

2.5. Case-based reasoning

CBR [90] has its roots in the memory-based methods, and it implements similarity heuristics, i.e., it reuses previous solutions to solve a similar new problem. Determining the similarity between problems is domain-specific, which is why CBR systems frequently employ the weighted Euclidean distance, where the details of the problem context are reflected in the feature weights. These weights used in determining the similarity between problems are global to features, which makes the decisions interpretable at the global level.

CBR has three major aspects that make it interpretable. First, it can produce a case as an example to explain a decision. Second, it can explain how similar the provided example is to the local instance that is being explained. And, when explaining the similarity, it can provide global weights for all the features in that local instance. Third, it has a small set of global weights to explain its global behaviour. For these reasons, it is possible to create a CBR system that is functionally equivalent to a model based on tree-based models (e.g., XGBoost). However, creating a functionally equivalent CBR system for a neural network is challenging. The only work that we are aware of, on building a CBR twin system that is functionally equivalent to a neural network, was done by Kenny and Keane [91]. The problem with adopting the concept of the

[3] https://github.com/marcotcr/lime

CBR twin in the methodology of this work is about reusing the input representations. While building a CBR twin, they reuse the abstract input representation from the neural network after training, thus losing one important aspect of transparency, which is having associated weights to each feature.

Considering the interpretable characteristics of CBR, it is often used to generate example-based explanations [91,92]. However, the global weights do not support local explanations in a similar way to the additive models present. For local explanations, we proposed the additive form of CBR, namely AddCBR, in our previous work [32]. The additive form is attained by re-scaling the feature values after the prediction is made by the weighted CBR regression model. Thus, by creating AddCBR, we are identifying a local representation for the instance that is being explained.

By directly leveraging the underlying model's structure and reasoning process, AddCBR produces explanations inherently aligned with how the model makes its predictions. This inherent alignment makes AddCBR a particularly suitable baseline for evaluating other additive feature attribution methods, as it faithfully reflects the model's decision logic. This work demonstrates the value of AddCBR for the first time as a baseline to evaluate the additive feature attribution methods. The experiments validating AddCBR as the baseline are described in Sections 4.3 and 4.4.

3. Generation of privacy-preserving ground-truth data

The proposed evaluation pipeline for additive feature attribution for regression begins with the generation of privacy-preserving ground-truth data that contains the underlying behaviours in the original dataset. The data generation process includes four major steps: i) selection of an original dataset, ii) capturing the behaviour in the original data, iii) synthetic data generation, and iv) evaluation of the generated synthetic data. All these steps are discussed in detail in the following subsections.

3.1. Selection of original dataset

The original datasets from which the prediction models are trained are often domain-specific, proprietary, or computationally expensive due to high dimensionality [27,29]. These issues restrict the use of the original datasets for iterations of experimental studies required for the development of the applications and methods. To mitigate the issues, the original dataset is used as a seed for generating synthetic datasets.

The selected original dataset for this study was acquired from Aviation Data for Research Repository[4] that was collected and processed by EUROCONTROL[5] from the Enhanced Tactical Flow Management System flight data messages containing all flights in Europe throughout the year 2019, from May to October. The dataset consists of fundamental details of the flights, flight status, preceding flight legs, Air Traffic Flow Management regulations, weather conditions, calendar information, etc. Specifically, the dataset contains 7,613,584 instances with 42 features, with the target variable in the dataset being the flight takeoff time delay. A brief description of the features used in this study is presented in Supplementary Table S1.1 and the detailed description of the dataset can be found in the works by Koolen and Coliban [93] and Dalmau et al. [37].

Prior to capturing the behaviours of the data, in the preprocessing step, instances with missing values and noise were removed from the dataset to ensure data quality and integrity. Also, the dataset was made free from the outliers so that the actual behaviour of the data could be captured.

[4] https://www.eurocontrol.int/dashboard/rnd-data-archive
[5] https://www.eurocontrol.int/

3.2. Capturing the data behaviours with density-based clustering

In the proposed approach, the behaviour of the data is the prime factor in evaluating the additive feature attribution methods. The data preserves different behaviours in the instances collectively, and the explanation is expected to recognize them. We recommend using any clustering method that captures underlying behaviours in the data and aligns well with the characteristics of the dataset and the specific task. The instances closest to the centroids of the clusters are those expected to represent best the underlying behaviours of the cluster, which can be used as the seeds to generate synthetic data. The premise is that each cluster captures a different behaviour, and their respective explanations have to be consistent with those behaviours.

Though synthetic data generation is founded on statistical principles, where sampling is done to preserve the underlying behaviours or distribution of the real data [94]. However, more recent studies confirm that both distribution- and cluster-based methods can effectively replicate multimodal and skewed structures in the data, while also noting that small biases (e.g., in cluster or feature frequencies) may occur if residual shifts are not addressed [95,96]. Also, the clustering-based synthesis follows the statistical rationale, ensuring that core data behaviour is preserved in the synthetic data [95]. These behaviours are the ground-truth information that is crucial for XAI evaluation. Considering the recommendations, the synthetic data generation process in this study starts with density-based clustering as an unsupervised method for capturing the behaviours of the flight delay prediction dataset. The process is described in the following subsections, which include mitigating the potential biases through the selection of the appropriate number of clusters and samples to form the seed datasets for synthetic data generation.

3.2.1. Density-based clustering

Several clusters were formed within the dataset using density-based clustering. Density-based clustering can discover clusters of arbitrary shape without requiring a pre-set number of clusters and treats outliers as noise, which makes it robust for complex temporal data with anomalies [97].

Dynamic time warping (DTW) [90] is used as the distance measure that determines the similarity between the data points, while clustering based on their intrinsic characteristics. DTW identifies a mapping between measurements from two time series such that the cumulative value of a given distance function is the minimum [90]. It also provides a flexible, non-linear alignment between time series, allowing comparison of sequences that are misaligned or of differing lengths and thus handling variability in timing or speed [98]. Particularly, the original dataset (described in Section 3.1) consists of temporal sequences of flight-status parameters where DTW is well-suited because it aligns time series that may be out of phase or evolve at different speeds. This makes it a widely adopted measure for clustering or classifying time-series data, as confirmed in the literature on time-series clustering [99]. Prior work on a similar application to ours, flight trajectory analysis, also uses DTW for clustering temporal data [100].

Thus, density-based clustering combined with DTW distance has been shown to effectively group similar behaviour patterns in multivariate datasets with time dependencies (e.g., clustering flight operation sequences) even when the data are noisy or irregular [97]. These clustering and distance techniques were adopted considering the nature of the original data, thus aiming at identifying patterns and grouping similar instances.

3.2.2. Selection of appropriate number of clusters

To determine the optimal number of clusters, both the x (features) and y (takeoff delay) values were considered. Initially, the Elbow method suggested by Madhulatha [101] and Yuan and Yang [102] was chosen to select the optimal number of clusters. However, the use of the Elbow method is criticised with experimental results by Schubert [103] and suggested using other methods. Afterwards, other methods were

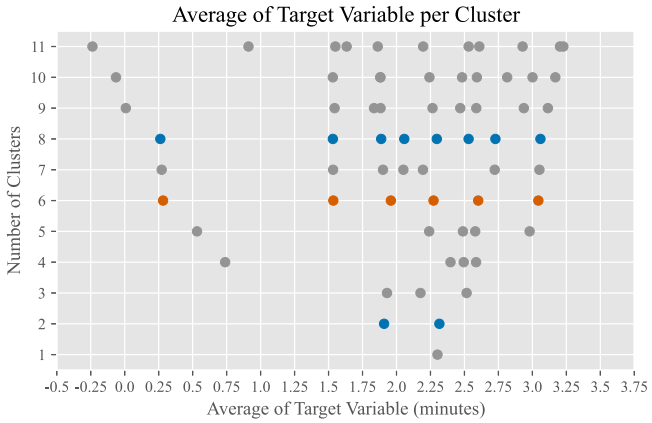


Fig. 2. Illustration of the separation of the average values of the target variable for each cluster, considering one to 11 clusters in the dataset. The dataset with six clusters (red dots) was chosen for primary analysis as the separations among the average values were more prominent than the others. The datasets with two and eight clusters are considered in the appendix (blue dots). Other datasets (grey dots) are omitted as the average values are closer and overlapping.

compared to select the optimal number of clusters, e.g., the Variance Ratio Criterion (VRC) proposed by Calinski and Harabasz [104] and the Jump Method by Sugar and James [105]. Nonetheless, all three methods produced different values for the optimal number of clusters. This hindered the selection of an optimal number of clusters using a method from the literature. For clarity, the plots for the used methods are illustrated in the Supplementary Material (Figure S2.1).

3.2.3. Refining clusters based on the target variable

As several methods from the literature for selecting the number of clusters did not reveal an optimal number of clusters with significant differences (Supplementary Figure S2.1), the focus shifted to the values of the target variable y. The average values of y were observed per cluster for the different numbers of clusters in the dataset as illustrated in Fig. 2. By breaking down the flight take-off delays into intervals, distinct classes are created where each class represents a specific behaviour.

3.2.4. Selection of datasets

The dataset with six clusters was chosen as the main dataset because of the clear separation of average y values in it, as illustrated in red dots in Fig. 2. Additionally, two other datasets with different numbers of clusters (two and eight clusters) were generated for sensitivity analysis, which are illustrated with blue dots in Fig. 2, and the results from experiments with these datasets are presented in the Supplementary Material (Section S3). Other datasets were omitted due to the overlapping cluster-wise average values of y (grey dots in Fig. 2).

3.3. Synthetic data generation for regression tasks

Generally, in the real world, data often contains noise and outliers that challenge the prediction models to learn crucial underlying behaviours of the data. In the preceding step, the original data was clustered based on these target behaviours. The synthetic data generation starts with identifying the seed instances from each cluster representing individual behaviour. An equal number of seed instances is randomly picked from each cluster to ensure that all the behaviours in the original data are equally represented in the synthetic data. The next step is to perturb the seed instances by maintaining an average of zero for the changes, thus retaining the original behaviours. In the final step, generate the target values randomly within the range of each cluster from the original data. This ensures representing the distinct behaviour of the cluster and minimising the overlap, i.e., similar data points in neighbouring clusters.

Table 2
Summary of the generated synthetic datasets for evaluation.

Criteria	Choice for evaluation	Sensitivity analysis	
No. of clusters	6	2	8
No. of seed instances	300	900	225
No. of perturbations	100	100	100
No. of total instances	180,000	180,000	180,000
No. of training instances	144,000	144,000	144,000
No. of testing instances	36,000	36,000	36,000

Here, three different synthetic datasets are generated based on the captured behaviour from the original dataset. The data generation was performed in two steps that are discussed in the following subsections.

3.3.1. Selection of random seed instances

Random seed instances were selected from each cluster within the selected datasets (i.e., two-, six-, and eight-cluster datasets). The number of seed instances selected for generating the synthetic data for a single cluster was different for the three synthetic datasets. However, within a single dataset, an equal number of seeds was selected. This balanced sampling was chosen to ensure that all behavioural patterns-including rare but operationally significant cases in small clusters-are preserved, while preventing large clusters from dominating the dataset. This strategy protects the representation of both over- and under-represented clusters, maintaining diversity in the generated data. Also, the selection of an equal number of seed samples from the clusters contributes to mitigating potential biases in the cluster-based synthetic data generation method [96]. Moreover, the clusters capture distinct patterns in the original data, with only minor patterns excluded when reducing the number of clusters (e.g., from eight to six). A comparative evaluation between the original and synthetic data showed that omitting these patterns was not detrimental, as confirmed by quantitative analysis (see Section 3.4), indicating similar distributions across clusters. These results demonstrate that the synthetic data preserves representative patterns, including those from both over- and under-represented clusters, thereby supporting the validity of our approach. Thus, we selected an equal number of seeds from each cluster for the three selected datasets (see Table 2).

3.3.2. Perturbation and synthetic data generation

As the last step, perturbations were applied to the selected instances to generate synthetic data. The continuous features were perturbed only while keeping the categorical features unaltered, and this choice of action is dataset-specific. Particularly, the binary and categorical variables (e.g., presence or absence indicators, airport codes, or system message types) were preserved in their original form to maintain semantic validity and avoid generating unrealistic combinations. In the flight dataset used in this work, the categorical features contain information about the airports, different system messages, etc., which influence the values of the continuous features within the range in individual clusters. To mitigate the issue of exceeding the value range of continuous features influenced by the categorical features, they are kept unchanged. For the continuous features, to preserve their original distribution in the synthetic data, we adopted a use and evaluate strategy starting from the simplest approach of sampling from a normal distribution rather than more sophisticated methods (e.g., Gaussian mixture models, Monte Carlo simulations). Following the Occam's Razor Principle [106]), we selected, used, and evaluated the simplest approach. We were motivated to adopt the simplest one after visually inspecting the original data distributions of the continuous features and noticing they resemble the bell-shaped curve of normal distributions.

Formally, the values x_{ij} of the feature vectors were perturbed from the respective distributions D_j while maintaining an average of zero for the added values to the features, ensuring the behaviour did not change. The y values were generated based on the range of each cluster,

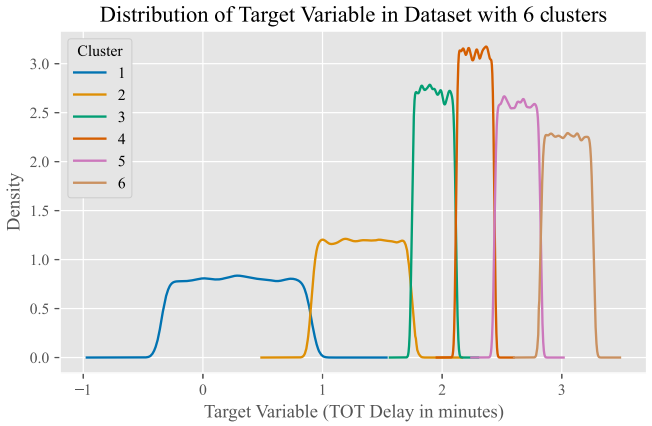


Fig. 3. Distribution of the target variable from the dataset with six clusters, that is the main choice for evaluating the additive feature attribution methods. Each colour represents the respective cluster indicated in the legend.

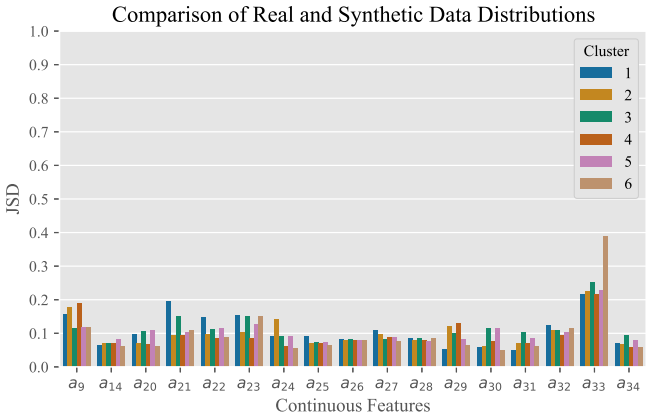


Fig. 4. Bar plot illustrating the JSD measures between the distributions of the continuous features from the original and synthetic datasets. The range of values for JSD is $[0, 1]$, where smaller values denote similarity and higher values denote dissimilarity between the distributions.

representing distinct classes of behaviour. The distribution of the target variable y for the dataset with six clusters is illustrated in Fig. 3, which is the primary choice for the evaluation of the additive feature attribution methods. It can be observed from the figure that the densities of clusters three – six are higher than those of clusters one and two. This higher density was the result of the effect of their smaller interval of the target variable, whereas an equal number of instances were perturbed from all the clusters. The datasets are prepared for the evaluation of the additive feature attribution methods with an equal number of instances in each of them. The summary of the synthetic datasets is presented in Table 2.

3.4. Evaluation of synthetic dataset

The quality of the synthetic data can be assessed in terms of the data distribution. The similarity between the distributions of the synthetic and the original dataset represents how well the synthetic dataset preserves the behaviour of the original data. Since only a few methods are proposed to generate synthetic data with ground-truth for XAI evaluation (described in Section 2.4), our approach is evaluated with a similar experiment done by Liu et al. [29] for evaluating the quality of the simulated dataset. The goal of this experiment is to examine how well the generated data captures the behaviour contained in the real data and assess the performance of our approach with the literature.

Jensen-Shannon Divergence (JSD) [107] was used to measure the similarity between the synthetic and the original data. JSD is a statistical measure that assesses the similarity between two probability distributions. It is derived from the Kullback-Leibler Divergence (KLD) and addresses its limitations, such as being asymmetric and unbounded. JSD overcomes these drawbacks by calculating the average of the KLD between each distribution and their average distribution. Due to its properties, JSD became a valuable tool to quantify and compare the similarity of probability distributions. JSD provides a symmetric and bounded measure of divergence within the range $[0, 1]$, where zero denotes identical distributions and one represents completely different distributions.

To evaluate whether the synthetic data preserves the characteristics of the original dataset, we formulate the following hypothesis:
*H*1: *Each continuous feature in the synthetic data has a similar distribution to the corresponding feature in the original data.*

The synthetic dataset was generated by perturbing the continuous features and the categorical features were kept unchanged as described in Section 3.3.2. Therefore, the hypothesis of this experiment is defined solely for the continuous features. Formally, it is hypothesized that $\hat{D}_j \approx D_j$ where $\hat{D}_j$ and D_j are the distributions of feature a_j from the synthetic and the original data, respectively.

The outcomes of JSD calculations for the six-cluster dataset are depicted in Fig. 4. To align with the JSD value range, the y-axis has been scaled from zero to one. For each continuous feature, the JSD values are presented for individual clusters as the clusters hold different behaviours of the data, resulting in different distributions. The reported JSD values range from 0.05 (minimum) to 0.24 (maximum) with an average ($\pm$ standard deviation) of 0.102 ± 0.05. Notably, the result reflects the high similarity between the distributions of all continuous features in the synthetic dataset and those in the original dataset, as evidenced by their corresponding JSD values.

The JSD values were exclusively calculated for the continuous features, leaving the categorical features as they were kept unaltered during synthetic data generation. As depicted in Fig. 4, it becomes evident that, with the exception of a_{33}, all features exhibit low JSD values. This observation underscores the remarkable similarity between the synthetic and original data. To visually emphasise this high similarity due to the low JSD values, the y-axis in Fig. 4 has been scaled from zero to one, mirroring the range of JSD values. Specifically, while Liu et al. [29] reported an average JSD of 0.20 for evaluating synthetic data, our approach achieved a substantially lower average JSD of 0.105, demonstrating superior performance in preserving the distribution of continuous variables in the synthetic data.

4. AddCBR as a baseline for evaluating feature attribution methods

The AddCBR is introduced in this paper in Section 2.5, which was at first conceptually presented in a workshop paper by the same authors [32]. Here, it is extended and its value as a baseline for evaluating feature attribution methods is demonstrated. Particularly, AddCBR is developed with a weighted CBR regression model [90] where the feature weights come from the data model. AddCBR achieves the additive form by re-scaling the feature values after the prediction with the weighted CBR regression model. Considering the interpretable characteristics of CBR and its use as a proxy model for explaining other models' output [91,92], AddCBR becomes a potential benchmark for local interpretability, which is demonstrated with experimental results in Sections 4.3 and 4.4.

4.1. Implementation of data model

The data model is implemented with an AI algorithm. However, there remains a prerequisite for selecting the algorithm for implementing the data model since the next step of creating the baseline requires a set of feature weights. Therefore, any algorithm with the ability to produce feature importance values or weights, such as decision trees

Table 3
List of used hyperparameters in optimising XGBoost model for regression through a grid search over different combinations. The values of the hyperparameters used for the final training are highlighted in bold font.

Hyperparameters	List of values
learning_rate	[0.01, **0.1**]
max_depth	[3, 5, **7**, 10]
min_child_weight	[**1**, 3, 5]
subsample	[**0.5**, 0.7]
colsample_bytree	[**0.5**, 0.7]
n_estimators	[100, 200, **500**]

or tree-ensembles, can be adopted to implement the data model in this step.

In this study, the data model for regression was implemented with a Gradient Boosted Decision Trees (GBDT) ensemble method, namely XGBoost [108], and trained to predict the flight take-off time delay. We used XGBoost given its potential to be more accurate in prediction tasks for structured or tabular data [4,109] than the other widely used variants of GBDT, e.g., LightGBM [110], that was utilized in a previous work in air traffic delay prediction [37].

The XGBoost regression model was trained with the hyperparameter values selected through a grid search over 288 different combinations. Here, grid search is exploited in the process of finding the appropriate hyperparameter values following the works of Claesen and De Moor [111], who highlighted the importance of such methods in model optimization. Table 3 presents the list of values of the hyperparameters used for optimising the XGBoost regression model. The performances of the regression model for different combinations of the hyperparameter values were assessed using Mean Absolute Error (MAE). The trained model with the selected hyperparameter values performed with an MAE of 9.9 min, whereas the MAE was 10.02 min with the default values of the hyperparameters. The final hyperparameters used to train the regression model are: $learning\_rate = 0.1$, $max\_depth = 7$, $min\_child\_weight = 1$, $subsample = 0.5$, $colsample\_bytree = 0.5$, and $n\_estimators = 500$.

4.2. Creating additive CBR

CBR can generate example-based explanations. However, as discussed in Section 2.5, its global weights describe overall feature importance and do not directly support local explanations in the way additive models do. We refer to this standard form as Global CBR. To address this, we create AddCBR by re-scaling the values from the CBR regression model after prediction. This transformation allows AddCBR to offer local explanations and serve as a benchmark for evaluating local interpretability of additive feature attribution methods. In addition, AddCBR is designed to train using the feature importance values from the data model and produce additive feature attributions that are directly comparable to those of the methods to evaluate. Thus, it enables an objective assessment of the attribution quality of the additive feature attribution methods.

The AddCBR baseline is designed to transform the output of a CBR regression model into the additive feature attribution form defined in Eq. (1). Recalling the formal definitions from Section 2.1, $x_i = [x_{i1}, \ldots, x_{im}]$ be the feature values of ith instance, and $\omega = [\omega_1, \ldots, \omega_m]$ be the corresponding feature weights from the feature importance values learned by the data model. In the first step, $\hat{y}_i$ is predicted using the regression model Ω. Here, Ω is represented by a CBR regression model. Then, a scaling multiplier γ_i is obtained by dividing the prediction $\hat{y}_i$ by the sum of its factors, i.e., the dot product of the feature values of x_i and weights ω, using Eq. (2):

$$\gamma_i = \frac{\hat{y}_i}{\left(\sum_{j=1}^{m} x_{ij}\omega_j\right)} \tag{2}$$

Finally, the feature attribution values can be obtained by a dot product of the multiplier γ_i and the factor of the given instance, i.e., $(x_i \cdot \omega)$. Here, the multiplier γ_i rescales the contributions so that their sum matches the model output $\hat{y}_i$. Hence, for the given instance x_i, the final attribution vector $\phi_i = [\phi_{i1}, \ldots, \phi_{im}]$ is obtained as the additive feature attribution as per-feature contributions. For example, if a regression model predicts a delay of $\hat{y}_i$ min from the data instance x_i of a given flight, AddCBR decomposes value of $\hat{y}_i$ into per-feature contributions, i.e., $\phi_{i1}, \ldots, \phi_{im}$, whose sum equals to $\hat{y}_i$, providing a transparent, model-aligned baseline for evaluating other additive feature attribution methods. The whole process of creating AddCBR is summarised in Algorithm 1.

Algorithm 1: Additive CBR.

Input: x_i: data point, ω: feature weights.
Output: ϕ_i: contributions of the features to the prediction.
1 $\hat{y}_i \leftarrow$ predict using CBR for x_i with ω
2 $\gamma_i \leftarrow$ compute using Eq. (2)
4 $\phi_i \leftarrow \gamma_i \cdot (x_i \cdot \omega)$
5 **return** ϕ_i

In the proposed evaluation pipeline, while implementing AddCBR, the feature importance values from the data model (i.e., the XGBoost model trained in the previous step) were considered as the weights (ω). And, for the CBR model within AddCBR, three nearest neighbour instances (i.e., $k = 3$) were considered to predict the target variable. Particularly, the CBR model predicts by averaging the y_i of the three nearest neighbours retrieved using the Euclidean Distance weighted with the feature importance values from the XGBoost model. The choice of three nearest neighbours was made since anecdotal tests suggested that three neighbours perform better than one. Moreover, it is a commonly successful default that reflects the bias-variance trade-off sweet spot to capture important local structure for many practical situations [112].

The evaluation of AddCBR as a baseline was performed through experiments on prediction performance in terms of local accuracy, and with local and global assessment by feature ranking and the impact of the most and least important features on the prediction. In these experiments, the results from AddCBR are compared against the results from the XGBoost regression model.

4.3. Prediction performance of AddCBR

The prediction performance metric is the MAE and Standard Deviation of Absolute Error (σ_{AE}). MAE is the average difference between the actual observation y_i and the prediction $\hat{y}_i$ from the model. σ_{AE} signifies the dispersion of the absolute error around the MAE. The measures were calculated using Eqs. (3) and (4), respectively. As both the MAE and σ_{AE} are representations of errors done by the models while predicting, lower values indicate better results.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{3}$$

$$\sigma_{AE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left|y_i - \hat{y}_i\right| - MAE\right)^2} \tag{4}$$

To assess whether the proposed AddCBR model maintains predictive performance comparable to the reference model, we state the following hypothesis:

Table 4
MAE and standard deviation (σ_{AE}) of XGBoost and AddCBR predicting flight delay. The result is presented for all instances and the 1000 most accurate instances from the full test set (36,000 instances) and five test subsets (7200 instances). The differences between the predictions by XGBoost and AddCBR are below 1.22 min. The best values for each set are presented in bold font.

Set	Instances	All		Top 1000	
	Model	XGBoost	AddCBR	XGBoost	AddCBR
Test set	MAE	**0.152**	0.167	0.064	**0.003**
	σ_{AE}	**0.133**	0.158	0.068	**0.002**
Test Subset 1	MAE	**0.151**	0.168	0.068	**0.017**
	σ_{AE}	**0.133**	0.160	0.067	**0.009**
Test Subset 2	MAE	**0.153**	0.169	0.065	**0.017**
	σ_{AE}	**0.133**	0.157	0.062	**0.010**
Test Subset 3	MAE	**0.152**	0.167	0.072	**0.016**
	σ_{AE}	**0.133**	0.159	0.076	**0.009**
Test Subset 4	MAE	**0.150**	0.166	0.067	**0.017**
	σ_{AE}	**0.132**	0.157	0.063	**0.009**
Test Subset 5	MAE	**0.152**	0.168	0.068	**0.016**
	σ_{AE}	**0.133**	0.158	0.069	**0.010**

H2: The difference between model performance between AddCBR and XGBoost in MAE is negligible.

The prediction performances of XGBoost and AddCBR in terms of MAE and σ_{AE} are given in Table 4. The result is presented separately for the whole test set and the top 1000 instances where both XGBoost and AddCBR predicted with minimal error, i.e., the average difference between y_i and $\hat{y}_i$ was close to zero. Moreover, a more granular analysis of the model performance was conducted by partitioning the test set into multiple subsets and conducting independent predictions across these subsets. For the subsets, the prediction performance remained similar to the whole test set except the 1000 most accurate instances by AddCBR, even though the difference remains negligible as presented in Table 4.

The variations between XGBoost and AddCBR are confirmed to be negligible. As regression models trained for the context of predicting flight delay, they can be considered as functionally equivalent. It is observed that, for each unseen testing instance, both produced nearly the same predictions, considering their differences were below a small error of 1.22 min, that is, the maximum difference between the predicted $\hat{y}_i$ by XGBoost and AddCBR. This result supports the use of AddCBR as the baseline.

4.4. Local and global assessment on AddCBR

The feature rankings derived from both XGBoost and AddCBR were scrutinised to establish the baseline. For XGBoost, the features were ranked based on their importance values, and for AddCBR, the features were ranked based on their contributions to the prediction. These ranks of the features were presented through global and local representations. The global representation corresponds to the rankings across all clusters, while the local representation focuses on rankings within individual clusters.

Furthermore, an analysis was conducted for the impact on predictions resulting from the changes in the feature values, supported by a statistical significance test. In this experiment, the top and bottom five important features were selected from both XGBoost and AddCBR. Each feature was perturbed five times with different multiples of the initial value. Particularly, if the initial value of the feature was f, the five perturbations were $2f$, $3f$, $4f$, $5f$, and $6f$. For each perturbation, the prediction was done with XGBoost while the other feature values were kept unchanged, and the change in prediction was measured in percentage with reference to the initial prediction. Finally, the average changes of the predicted values in portion were compared for the top and bottom features separately for both XGBoost and AddCBR.

4.4.1. Hypotheses

We consider two premises to determine whether AddCBR is adequate as a baseline. The first premise is that the baseline should have global and local rankings that are consistent. In other words, if features a, b, and c are among the top positions at the global ranking, then they should also appear at the top positions at the local ranks. The comparison between AddCBR and XGBoost with respect to consistency between local and global ranking is evaluated through the hypothesis:

H3: Local and global feature rankings produced by AddCBR are more consistent than those produced by XGBoost.

The second premise is that the baseline should be confirmed to have the best ranking. This can be done through the verification that the features ranked at the top positions are those that produce the highest impact on the prediction results, while those features ranked at the bottom produce no or minimal impact on the results. For this reason, we want to demonstrate that the baseline is the feature attribution method for which the *difference* between the impact produced by the top and bottom features is the highest.

To compare AddCBR and XGBoost with respect to the performance in ranking features, we utilize the top and bottom five ranked features by both methods, and formulate the hypothesis as follows:

H4: The difference between the impact produced by the top five and the bottom five features in the ranking obtained with AddCBR is higher than the difference obtained with XGBoost across all clusters with statistical significance.

We also performed a paired t-test considering a null hypothesis where the difference is not statistically significant.

Furthermore, the changes in prediction were examined particularly for the most important feature a_{34} and the least important feature a_9 based on the feature contributions provided by the AddCBR (see Fig. 5b) through the hypothesis:

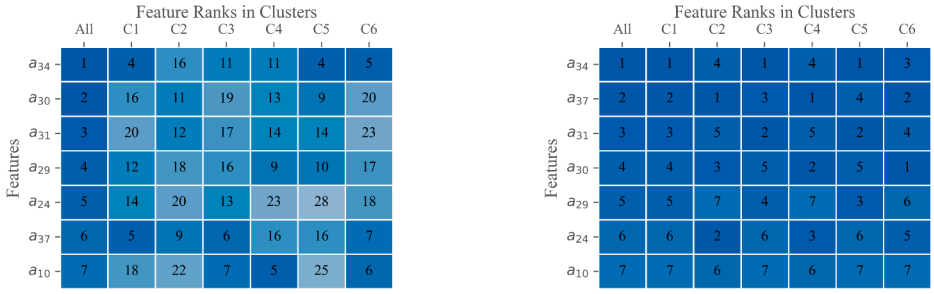
H5: The impact of the most important feature a_{34} is at least four times higher than the impact of the least important feature a_9 from AddCBR.

4.4.2. Results

The feature ranks extracted from XGBoost and AddCBR are presented in Fig. 5. For both methods, the top seven continuous features are shown. Notably, the features are the same for both methods at the top, but their ranks vary for global and local representations. However, more discrepancies are observed in the local representation from the XGBoost. On

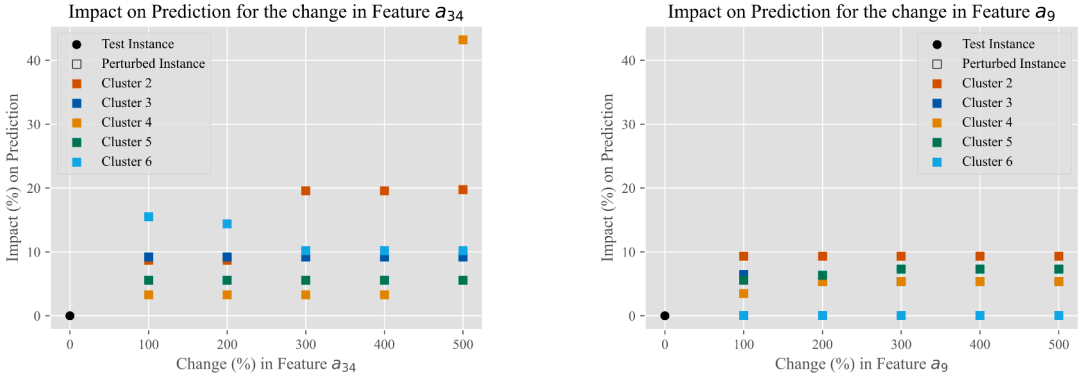
Table 5
Average impact on prediction measured in percentage for the change in values of top and bottom five features based on their importance from XGBoost and AddCBR. The higher values for the differences in impacts are better. Using a paired t-test, the impacts on the predictions were analysed and the test results with significant values i.e., $p < 0.05$, are marked with asterisks (*).

Cluster	Model	Average Impact (%) of Features			t-test	
		Top Five	Bottom Five	Difference	t	p
All	XGB	27.0	15.7	11.3	3.306	0.001*
	AddCBR	27.0	15.9	11.1	3.249	0.001*
1	XGB	15.9	20.7	4.8	−1.454	0.927
	AddCBR	27.0	15.7	11.3	3.306	0.001*
2	XGB	15.6	16.8	1.2	−0.451	0.674
	AddCBR	27.0	15.7	11.3	3.299	0.001*
3	XGB	15.9	16.8	0.9	−0.352	0.637
	AddCBR	27.0	15.7	11.3	3.299	0.001*
4	XGB	17.5	19.5	1.9	−0.710	0.761
	AddCBR	27.0	15.7	11.3	3.299	0.001*
5	XGB	17.8	19.5	1.8	−0.642	0.739
	AddCBR	27.0	15.7	11.3	3.299	0.001*
6	XGB	16.1	23.4	7.3	−2.202	0.986
	AddCBR	27.0	15.7	11.3	3.299	0.001*



(a) Feature ranks from XGBoost. (b) Feature ranks from AddCBR.

Fig. 5. Feature ranks of the most important features from (a) XGBoost and (b) AddCBR. Lower value and a darker shade of blue both correspond to the high importance of the features. The ranks of the seven out of 42 features from the selected dataset are illustrated since both the models considered them as the most important features, but with different rankings for global and local representations. For example, globally considering all the clusters, feature a_{30} is ranked *second* by XGBoost, whereas AddCBR ranked it *fourth*. In local representation, for individual clusters, the ranking of a_{30} by AddCBR remain similar to global, but discrepancies are observed for XGBoost.



(a) Most important feature. (b) Least important feature.

Fig. 6. Impact on the prediction by changing the values of the continuous features with (a) most and (b) least importance based on their contribution to feature attribution from AddCBR. To demonstrate a clear separation of the clusters, the data from Cluster 1 is excluded from the illustrations due to its impact on prediction.

the contrary, the ranking of AddCBR remains consistent. Specifically, the feature a_{34} is the most important feature from both XGBoost and AddCBR in global representation but it does not remain the same in any of the local representations from XGBoost. Unlike XGBoost, for AddCBR, the top rank of the feature a_{34} is preserved in the local representations as it stands out to be the most important feature for clusters one, three, and five. For other clusters, a_{34} became the third (in cluster six) and fourth (in clusters two and four) most important feature according to the contributions from AddCBR.

The results of Hypothesis $H4$ are detailed in Table 5. The result is presented for the whole dataset and each cluster individually. For each selection of clusters and the models, i.e., XGBoost and AddCBR, the average impact on prediction in percentage is given for the five most (top) and least influential (bottom) features. Considering the role of the features for flight take-off delay prediction, the most influential features are found to be related to turnaround and scheduling processes (e.g., differences between scheduled and actual turnaround times, available turnaround durations, and remaining time until milestones such as target off-block time or estimated time over for the aerodrome of departure). On the other hand, the least influential ones are primarily secondary timing differences (e.g., gaps between planned and updated off-block time or overall flight durations).

Table 5 includes the outcome of the paired t-test between the impact of the top and bottom five features. The cases where the top features have significantly more impact on the prediction than the bottom features based on the p values are marked with asterisks (*). We reject the null hypothesis for the outcomes for which the resulting p values are less than the level of significance (i.e., 0.05). From the results, it can be observed that the impact is more consistent with the features ranked from AddCBR than those from XGBoost. Particularly, for both models, considering all the clusters, the top features significantly impact more than the bottom features. This condition is maintained only for AddCBR when individual clusters are considered. For XGBoost, it is quite the opposite, i.e., the top features impact the prediction less than the bottom features in individual clusters.

Fig. 6 illustrates the impact of the features on prediction supporting the Hypothesis $H5$. The changes in prediction were examined for the most important feature a_{34} and the least important feature a_9 based on the feature contributions provided by the baseline AddCBR. While computing the impact, the instance closest to the cluster centroid was selected as the actual instance (black dots in Fig. 6). Five perturbations were done from double to six times the feature value to assess the impact on prediction (coloured dots based on clusters in Fig. 6). The impact of these two features was assessed one at a time. The feature value was perturbed while keeping the values of other features unchanged. It can be observed from the subplots that the highest impact of the prediction was more than 40 % due to the change in the values of a_{34} that is the most important feature. On the contrary, the impact on prediction was within 0–10 % when the value of a_9 was changed.

4.4.3. Discussion

The baseline selection process contained two separate experiments with the feature ranks and feature impact on the prediction, where the features are ranked based on the importance values from XGBoost and

the contributions of features to the prediction from AddCBR. From the results of both experiments, it was observed that XGBoost and AddCBR perform similarly in the case of global representation of the data or considering the whole dataset. However, in the case of local representations or individual behaviours presented within distinct clusters, the performance of AddCBR remains consistent with reference to the global representation, which is not preserved by XGBoost. This consistent global and local representation of feature ranks and impacts on prediction strengthens the choice of AddCBR as the baseline. Therefore, the AddCBR was chosen as the baseline for the experiments that evaluate the additive feature attribution methods.

5. Evaluation of feature attribution methods with CQV and domain-specific aspects

This section presents the details of the proposed evaluation criteria for the feature attribution methods, preceded by a brief description of the implementation of feature attribution methods.

5.1. Implementation of feature attribution methods

To generate explanations for the predictions from XGBoost, the two previously introduced additive feature attribution methods, namely SHAP [10] and LIME [11] were implemented. These methods were selected because of their vast popularity in the recent XAI research [16]. SHAP was implemented using *TreeExplainer* [10] with default settings. LIME was implemented with 1000 perturbations and 1000 samples. We note that there are other additive feature attribution methods for XAI, such as DeepLIFT [113] and Layer-wise Relevance Propagation [47]. We did not include those because they are not model-agnostic; they are specifically designed for neural networks and are therefore not applicable to this study focusing particularly on regression tasks with decision trees and CBR.

Given that the additive feature attribution methods are implemented, a series of experiments was conducted to evaluate the performance of the additive feature attribution methods (i.e., SHAP and LIME), which are described in this section. For the evaluation, we assess the quantitative quality of the feature attributions of SHAP and LIME by comparing them with the baseline AddCBR using the three evaluation metrics, namely, *feature ranking*, *feature attribution* and *feature impact*. Feature ranking compares the ranks of the important features from the additive feature attribution methods in comparison to the baseline AddCBR. In the second procedure, the quantitative quality of the feature attributions by the additive feature attribution methods is assessed in terms of different behaviours in the data. The last procedure includes the assessment of the impacts of the top and bottom-ranked features from the additive feature attribution methods on the data model's prediction. Each experiment is presented with a description of the methodology, metrics, and hypotheses applied. The results are presented initially for the dataset with six clusters, followed by a sensitivity analysis where the results are compared against the dataset with two and eight clusters.

5.2. Evaluation on feature ranking

The first evaluation experiment was conducted to assess how the additive feature attribution methods rank the important features compared to the baseline AddCBR. Particularly, the Normalised Discounted Cumulative Gain (nDCG) was used to evaluate the feature ranks produced by SHAP and LIME by comparing the feature ranks from the baseline. nDCG [114] is a widely used evaluation metric in information retrieval and recommendation systems. It measures the quality of a ranked list by considering both the relevance and the position of items. In terms of implementation, studies [115,116] show that varied results can be obtained while using different libraries. In this experiment, the nDCG metric is calculated using the sklearn library [117]. In practice, nDCG normalises the cumulative gain of the ranked list by dividing it by the ideal cumulative gain, resulting in a score in the range $[0, 1]$. A higher nDCG value indicates a better-ranked list that effectively captures the relevance of items in a specific context.

Table 6
The maximum ($\max_{nDCG}$), average (μ_{nDCG}), and standard deviation (σ_{nDCG}) of nDCG scores for the feature ranking from SHAP and LIME for all the test instances. The nDCG scores were calculated considering the feature order from the AddCBR as the baseline. For $\max_{nDCG}$ and μ_{nDCG}, the higher values are better, and for σ_{nDCG}, the lower values are better, which are presented in bold font.

Method	SHAP			LIME		
Cluster	$\max_{nDCG}$	μ_{nDCG}	σ_{nDCG}	$\max_{nDCG}$	μ_{nDCG}	σ_{nDCG}
All	**0.968**	**0.852**	**0.038**	0.960	0.844	0.039
1	**0.963**	**0.855**	**0.038**	0.950	0.843	0.039
2	0.948	**0.844**	**0.038**	**0.952**	0.843	0.040
3	**0.968**	**0.854**	**0.037**	0.946	0.844	0.040
4	**0.955**	**0.849**	**0.037**	0.951	0.844	0.039
5	**0.964**	**0.855**	**0.037**	0.942	0.843	0.039
6	**0.960**	**0.854**	**0.038**	**0.960**	0.844	0.039

To evaluate the quality of feature rankings using the nDCG metric, we propose the following hypothesis:

H6: The ranking of the feature contributions produced by SHAP results in higher nDCG values than those from LIME.

Table 6 presents the maximum ($\max_{nDCG}$), average (μ_{nDCG}), and standard deviation (σ_{nDCG}) of nDCG scores for all the clusters together and for individual clusters. From the table, it is evident that the feature ranks from SHAP produced better results in terms of nDCG score across all the clusters. However, for cluster two, LIME achieved a higher $\max_{nDCG}$ than SHAP, and for cluster six, the values of $\max_{nDCG}$ were equal for both methods. Overall, both $\max_{nDCG}$ and μ_{nDCG} values for the feature ranking by SHAP are higher than LIME, which advocates for a better feature ranking by SHAP. However, the μ_{nDCG} values are closer yet SHAP stands out to produce better feature ranks based on their contribution to the prediction. This is also observed with a lower σ_{nDCG} for SHAP emphasising less variation in the feature ranks compared to the baseline AddCBR. Evidently, the nDCG scores across individual clusters are consistent with the overall value, which indicates the balance between the global and local representation produced by SHAP.

The overlaps in the feature rankings by SHAP and LIME are also compared with the baseline AddCBR using the illustration presented in Fig. 7. The illustration shows that the highest-importance feature (a_{34}) is common to both methods and the baseline. Among the top-5 features ranked by the baseline AddCBR, SHAP shares all features with different rankings, while LIME shares only three, resulting in a greater overlap for SHAP with the AddCBR baseline. These observations quantitatively confirm that all methods agree on the most critical feature and that SHAP's ranking aligns more closely with the AddCBR baseline than LIME's does. Thus, the illustration of overlaps in feature ranking also resembles the quantitative evaluation with the nDCG metric presented in Table 6.

5.3. Evaluation on feature attribution

The proposed evaluation approach with synthetic data is based on the concept of constraining the data generation around cluster centroids to capture the behaviour of each cluster. This concept enables assessing whether feature attribution methods can recover this same behaviour. In other words, we expect the additive feature attribution methods to attribute features in a way that reflects the data distribution of each cluster. We quantitatively evaluate explanation quality by comparing the variability of the feature attributions to the variability of the feature values in the synthetic data. In this experiment, the Coefficient of Quartile Variation (CQV) [33], a robust statistical measure of relative dispersion, is used as a metric to evaluate additive feature attribution methods. Using CQV, we demonstrate that the better-performing additive feature attribution method will produce explanations showing fea-

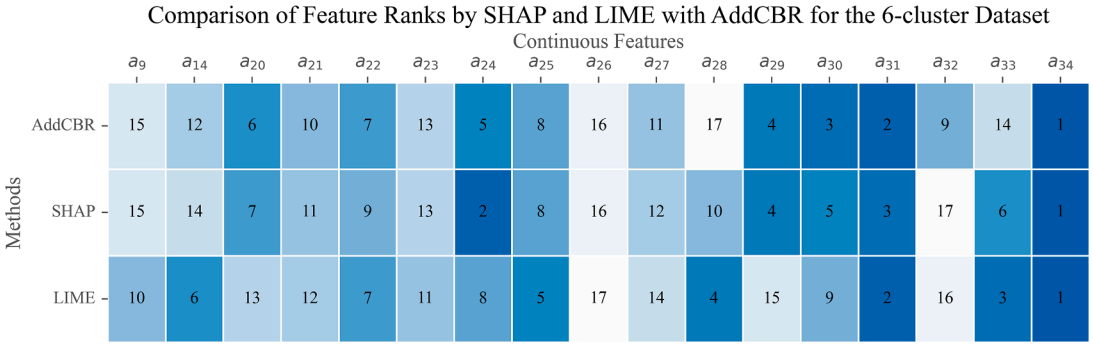


Fig. 7. Comparison of feature ranks from SHAP and LIME with the feature ranks from AddCBR as the baseline. Lower value and darker shade of blue both correspond to the high importance and rank of the features. The ranks of the 17 continuous features from the selected dataset with six clusters are illustrated with their relative ranks. Note that the feature a_{34} is ranked first by all the methods, and between SHAP and LIME, the ranking by SHAP is closer to the ranking by the baseline AddCBR.

ture contributions with similar variability to the feature values in the synthetic dataset. The value of CQV ranges from zero to infinity, where values close to zero indicate less variability in the data and vice versa. The value of CQV is computed using the Eq. (5) [33], where Q_1 is the population 25th percentile and Q_3 is the population 75th percentile.

$$CQV = \frac{Q_3 - Q_1}{Q_3 + Q_1} \tag{5}$$

Intuitively, a better-performing additive feature attribution method will produce output whose variability closely mirrors the variability of the actual values of the features. This intuition is rooted in the stability property of explanations: *similar instances should have similar explanations* [118]. Due to the difference in nature (e.g., unit, value ranges, etc.), the similarity between the inputs (feature values) and explanations (feature contributions) is measured using variability. This works only because it is for regression, where the patterns in the data are preserved and transferred to the contributions in the output. Here, feature values and contributions are not necessarily the same, but they preserve similar patterns. Thus, if the feature values do not vary much within a cluster, reliable feature attribution values should contain little variation for those instances. Again, if a feature's values vary widely, a method responding more accurately to those differences may show a wider spread in feature attributions. Botta-Dukát [119] demonstrated that the CQV of two sets of values can be best compared using scatter plots, and the points closer to the reference 1:1 line indicate similar variability between the two sets of values. In this experiment of evaluating the feature attribution of the additive feature attribution methods, CQV was computed for the feature values and the contributions of the features from the additive feature attribution methods, thus comparing their variability. The CQV of feature values and contributions is compared using scatter plots and the reference 1:1 line. The plots of CQV values closer to the 1:1 line indicate similar variability of the feature value and contribution presented through the axes of the plots. Particularly, the plots of CQV are generated for SHAP and LIME to compare with the CQV plots of the baseline AddCBR.

To compare SHAP and LIME in terms of the variability in their feature attributions with respect to the feature values, quantified by CQV, we propose the following hypothesis:

H7: Feature contributions from SHAP produce smaller CQV values than those from LIME.

Fig. 8 presents the CQV for all the feature values and the contributions from SHAP and LIME compared to the baseline AddCBR. In the figure, the axes of each subplot refer to the data and feature attribution for individual clusters. It was observed from the illustrations that the CQV of the feature values and the feature contributions from SHAP are closer to the 1:1 line than those of LIME. The plots closer to 1:1 refer to identical variability in the data and the feature contributions. The data points for the baseline AddCBR are closer to the 1:1 line and accumulated near the lower left segment of the subplots, signifying that the variations in the data and feature contributions from AddCBR are identical and low. On the other hand, the feature attribution produced by SHAP and LIME both have more variability than the data, as their data points are scattered in the subplots.

In the tasks of regression, the feature values are responsible for the prediction, while the contributions from the additive feature attribution methods sum up to the prediction. Though the feature values and the contributions are different measures, the variability among these two measures should follow the same pattern as they regard a single prediction. From Fig. 8, it is prominent that the CQV of the feature values and contributions (yellow dots) from AddCBR follows the 1:1 line with outliers in clusters two, three, and six. The CQV values for both SHAP and LIME are sparsely distributed along the x-axis, signifying the fact that the variability in their contributions is not following the variability in the data. The illustration in Fig. 8 aligns with the claim of Krishna et al. [120], which states that the XAI methods often disagree in terms of the explanations they produce and the behaviours of the corresponding data. A similar conclusion can be drawn from the presented analysis on the variability of data and feature attributions from the additive feature attribution methods. However, the variability of the contributions produced by SHAP is more similar to the data than the same for LIME.

5.4. Evaluation on feature impact

In this experiment, the level of impact on the prediction is assessed when we change the values of the top and bottom ranked features. The features are ranked according to the feature contributions from different additive feature attribution methods, i.e., SHAP and LIME. We conducted this experiment with a similar procedure to the experiment discussed in Section 4.4. The value of the selected feature was perturbed while keeping the values of other features unchanged. Then, the prediction was done by XGBoost, and the impact of the change of feature values on the prediction was calculated. Both the changes in prediction and feature values are measured in percentages. A paired t-test was also performed to assess the significant difference in the feature impacts.

To evaluate how well the feature attribution methods distinguish the most impactful features from the less impactful ones, we formulate the following hypothesis:

H8: The difference between the impact produced by the top five and the bottom five features in the ranking obtained with SHAP is higher than the difference obtained with LIME across all clusters with statistical significance.

The impacts on prediction for the changes in the most and least important features from SHAP and LIME were assessed in global and local representations. In addition, the differences in impacts were statistically tested considering all the clusters and individual clusters.

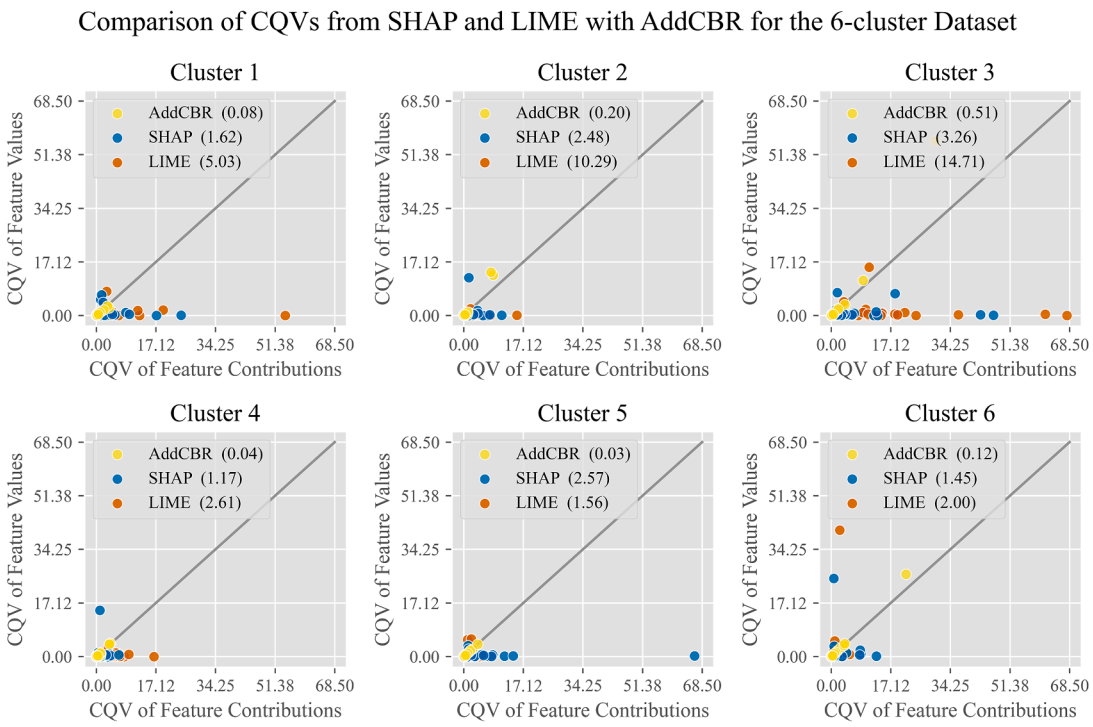


Fig. 8. Evaluation of feature attribution from SHAP (blue dots) and LIME (red dots) considering AddCBR (yellow dots) as the baseline with scatter plots for CQV of the feature values and contributions, where each subplot represents individual clusters. In each subplot, the data points correspond to individual features, and the grey diagonal line is the reference 1:1 line. Points closer to the 1:1 line refer to similar variability between feature values and contributions. In the legends, the values in parentheses refer to the average distance to the 1:1 line from the scattered points for the respective methods.

Table 7
Average impact on prediction measured in percentage for the change in values of top and bottom five features based on their contributions from SHAP and LIME. The higher values for the differences in impacts refer to better feature attributions. Using a paired *t*-test, the impacts on the predictions were analysed and the test results with significant values i.e., $p < 0.05$, are marked with asterisks (*).

Cluster	Method	Average Impact (%) of Features			*t*-test	
		Top Five	Bottom Five	Difference	*t*	*p*
All	SHAP	27.0	16.0	11.0	3.222	0.001*
	LIME	20.8	18.2	2.6	0.782	0.217
1	SHAP	27.0	15.6	11.4	3.336	0.000*
	LIME	26.0	15.8	10.2	3.007	0.001*
2	SHAP	27.0	15.6	11.4	3.346	0.000*
	LIME	27.0	15.7	11.3	3.303	0.001*
3	SHAP	27.0	15.7	11.3	3.330	0.000*
	LIME	26.0	15.7	10.3	3.043	0.001*
4	SHAP	27.0	15.6	11.4	3.346	0.000*
	LIME	27.0	15.9	11.1	3.253	0.001*
5	SHAP	27.0	15.8	11.2	3.295	0.001*
	LIME	26.4	15.7	10.7	3.146	0.001*
6	SHAP	23.1	15.6	7.4	2.741	0.003*
	LIME	26.0	15.8	10.1	3.002	0.001*

We performed a paired *t*-test considering a null hypothesis where the impact would be no different, with a level of significance of 0.05. The results of the tests are detailed in Table 7. The average impact on prediction in percentage is given for the top and bottom five important features based on the ranks produced by SHAP and LIME. The result of the paired *t*-test between the impacts of top and bottom five features is also presented, and the cases where the top features have significantly ($p < 0.05$) more impact on the prediction than the bottom features are emphasised. Evidently, for both methods, while individual clusters are considered, the top features had a significantly higher impact than the bottom features. However, the top features from LIME didn't have a significantly higher impact on the prediction, whereas the top features from SHAP had a higher impact while all the clusters were considered together.

The assumption behind the experiment on feature impact is that the features with the highest contribution require small changes to impact the prediction result. On the other hand, features with low contributions would require large changes to impact the prediction result. However, this assumption was proved by investigating the impact on prediction by changing the values uniformly for high and low contributing features. From the results, it was found that for a uniform change in the feature values, the impact is more from the high contributing features. Specifically, for the ranking from SHAP, the differences in the impact of the high and low contributing features are more significant than LIME based on the corresponding *p* values of the statistical significance test presented in Table 7. Most importantly, the difference in impact on prediction between the high and low contributing features from LIME is not significant globally, i.e., considering all the clusters. This result can be justified by the fact that LIME is designed to generate local explanations [11], thus it is unable to differentiate the features based on their importance values at a global level.

Throughout the presented experiments in Sections 5.2–5.4, the results demonstrate that the feature ranking, attribution, and impact from SHAP are better than those from LIME. Consequently, these findings are also aligned with the claim from the literature that a method employing standardization on input, such as LIME, does not produce feature attributions of the same quality as the method that does not use standardization, like SHAP, in a regression task.

6. Conclusion and future works

This article advances the evaluation of feature attribution methods for regression. The framework encompasses an evaluation strategy grounded in the intrinsic characteristics of the data and preserving its privacy, offering a comprehensive assessment of the feature attribution methods within the context of regression problems. First, the article contributes an approach to generate synthetic regression data that replicates the behaviour of a given data set, and shows how to use the synthetic data to evaluate the additive feature attribution methods applied to the original data set. This proposed methodology can be reused by those who want to conduct a thorough analysis of real-world applications, provided the authorities share the centroids of the clusters that contain the intrinsic characteristics of the original data, even if the data is proprietary or confidential. Notably, this approach provides a solution to data privacy concerns that restrict dataset distribution complying with different policies and regulations such as the General Data Protection Regulation (GDPR) [89]. Second, we demonstrate how the additive representation of cases, AddCBR, aligns global and local feature attributions, making it possible to use it as a benchmark for evaluation. The AddCBR is created as a functionally equivalent model to XGBoost by utilising the feature importance values as weights for CBR. However, the process of creating AddCBR is not defined for the data models that learn an abstract representation of the data (e.g., neural networks), which is a limitation of this study. Third, we proposed and demonstrated the use of a statistical metric, CQV, in evaluating feature attribution methods alongside other metrics from the literature. Given the extensive use of CQV as a stability metric in different domains, the approach addresses the lack of consensus in the literature on the evaluation approaches for XAI methods. As a whole, we proposed an evaluation pipeline for feature attribution methods and effectively evaluated two such methods, namely SHAP and LIME, against the AddCBR benchmark. On a different note, the outcomes of the evaluation experiments confirmed that LIME, a representative of methods incorporating a standardization process, does not yield feature attributions of satisfactory quality, which aligns with the current XAI literature [27,77].

This work evaluates the proposed pipeline on a single aviation dataset, providing extensive functional validation, limiting generalizability to other domains. Functional evaluation is an essential prerequisite to user validation in XAI, ensuring that methods are rigorously tested from an algorithmic standpoint before involving domain experts. Moreover, the development and assessment of XAI methods are inherently domain-specific, and expert evaluations are often difficult to conduct due to scarcity of experts, subjectivity, and high opportunity costs [75]. For these reasons, we focused on functional evaluation in this study. We acknowledge this as a limitation and note that future work will extend the evaluation to additional domains, such as healthcare, finance, and manufacturing, to broaden applicability.

As the research progresses, the exploration will be extended for classification tasks and other types of data models (e.g., neural networks). AddCBR is currently limited to regression models with feature importance values, though extensions to neural networks (e.g., deep CBR [121,122], and other variants reviewed by Leake et al. [123]) are a prospective future direction. Exploration of a suitable variant of CBR for other XAI methods (e.g., saliency maps or gradient-based methods) can be done with further research. These will contribute to the refinement of XAI methods across different application domains. Another possible research direction is to investigate different methods, such as generative modelling methods like generative adversarial networks, to generate synthetic data other than the presented clustering-based approach to evaluate the performance of the additive feature attribution methods.

CRediT authorship contribution statement

Mir Riyanul Islam: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; **Rosina O. Weber:** Conceptualization, Investigation, Methodology, Resources, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing; **Mobyen Uddin Ahmed:** Funding acquisition, Project administration, Supervision; **Shahina Begum:** Funding acquisition, Project administration, Supervision.

Data availability

All the synthetic datasets[6] and implementation scripts[7] are made available in Zenodo.

Declaration of competing interest

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

Acknowledgements

The following projects supported this study; i) TRUSTY, financed by the SESAR JU under the EU's Horizon 2022 Research and Innovation programme (Grant Agreement No. 101114838), ii) CPMXai, funded by the VINNOVA (Diary No. 2021-03679), iii) ARTIMATION, funded by the SESAR JU under the EU's Horizon 2020 Research and Innovation programme (Grant Agreement No. 894238), and iv) xApp, funded by the VINNOVA (Diary No. 2021-03971).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.knosys.2025.114599.

References

[1] D. Gunning, D.W. Aha, DARPA's explainable artificial intelligence program, AI Mag. 40 (2) (2019) 44–58. https://doi.org/10.1609/aimag.v40i2.2850
[2] T. Miller, Explanation in artificial intelligence: insights from the social sciences, Artif. Intell. 267 (2019) 1–38. https://doi.org/10.1016/j.artint.2018.07.007
[3] R. Hoffman, G. Klein, S.T. Mueller, M. Jalaeian, C. Tate, The Stakeholder Playbook for Explaining AI Systems, Technical Report, DARPA Explainable AI Program, 2021.
[4] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, C. Zhong, Interpretable machine learning: fundamental principles and 10 grand challenges, Stat. Surv. 16 (none) (2022) 1–85. https://doi.org/10.1214/21-SS133
[5] U. Ehsan, M.O. Riedl, Explainability pitfalls: beyond dark patterns in explainable AI, Patterns 5 (6) (2024) 100971. https://doi.org/10.1016/j.patter.2024.100971
[6] R.O. Weber, A.J. Johs, P. Goel, J.M. Silva, XAI is in trouble, AI Mag. 45 (3) (2024) 300–316. https://doi.org/10.1002/aaai.12184
[7] Y. Izza, A. Ignatiev, J. Marques-Silva, On tackling explanation redundancy in decision trees, J. Artif. Intell. Res. 75 (2022) 261–321. https://doi.org/10.1613/jair.1.13575
[8] B. Kim, C. Rudin, J. Shah, The bayesian case model: a generative approach for case-based reasoning and prototype classification, in: Neural Information Processing Systems, 2014, pp. 1–9. https://doi.org/10.48550/arXiv.1503.01161
[9] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115. https://doi.org/10.1016/j.inffus.2019.12.012
[10] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), NIPS'17, 2017, pp. 4768–4777. https://doi.org/10.48550/arXiv:1705.07874
[11] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), ACM, San Francisco, CA, USA, 2016, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778
[12] D. Slack, S. Hilgard, E. Jia, S. Singh, H. Lakkaraju, Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods, arXiv preprint (arXiv:1911.02508v2 [cs.LG]) (2020). https://doi.org/10.48550/arXiv.1911.02508

[6] https://zenodo.org/records/10115807
[7] https://zenodo.org/records/10152705

[13] J. Marques-Silva, X. Huang, Explainability is *not* a game, Commun. ACM 67 (7) (2024) 66–75. https://doi.org/10.1145/3635301

[14] D. Nguyen, Comparing automatic and human evaluation of local explanations for text classification, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 1069–1078. https://doi.org/10.18653/v1/N18-1097

[15] W.J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, Proc. Natl. Acad. Sci. 116 (44) (2019) 22071–22080. https://doi.org/10.1073/pnas.1900654116

[16] M. Mainali, R.O. Weber, What's meant by explainable model: a scoping review, in: Proceedings of the Workshop on XAI co-located with the 32nd International Joint Conference on Artificial Intelligence (IJCAI), 2023, pp. 1–8. https://doi.org/10.48550/arXiv.2307.09673

[17] J. Oramas, K. Wang, T. Tuytelaars, Visual explanation by interpretation: improving visual feedback capabilities of deep neural networks, in: Proceedings of the Seventh International Conference on Learning Representations (ICLR), 2019, pp. 1–29. https://doi.org/10.48550/arXiv.1712.06302

[18] M. Yang, B. Kim, Benchmarking Attribution Methods with Relative Feature Importance, arXiv preprint (arXiv:1907.09701 [cs.LG]) (2019). https://doi.org/10.48550/arXiv.1907.09701

[19] B. Barr, K. Xu, C. Silva, E. Bertini, R. Reilly, C.B. Bruss, J.D. Wittenbach, Towards Ground Truth Explainability on Tabular Data, arXiv preprint (arXiv:2007.10532v1 [cs.LG]) (2020). https://doi.org/10.48550/arXiv.2007.10532

[20] D. Mahajan, C. Tan, A. Sharma, Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers, arXiv preprint (arXiv:1912.03277v3 [cs.LG]) (2020). https://doi.org/10.48550/arXiv.1912.03277

[21] Y. Zhou, S. Booth, M.T. Ribeiro, J. Shah, Do feature attribution methods correctly attribute features?, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 36(9), 2022, pp. 9623–9633. https://doi.org/10.1609/aaai.v36i9.21196

[22] F. Doshi-Velez, B. Kim, Towards A Rigorous Science of Interpretable Machine Learning, arXiv preprint (arXiv:1702.08608v2 [stat.ML]) (2017). https://doi.org/10.48550/arXiv.1702.08608

[23] D. Gunning, Explainable artificial intelligence (XAI), Defense Adv. Res. Projects Agency (DARPA) 2 (2) (2017).

[24] R.R. Hoffman, S.T. Mueller, G. Klein, Explaining explanation, part 2: empirical foundations, IEEE Intell. Syst. 32 (4) (2017) 78–86. https://doi.org/10.1109/MIS.2017.3121544

[25] S.T. Mueller, R.R. Hoffman, W.J. Clancey, A.K. Emery, G. Klein, Explanation in Human-AI Systems: A Literature Meta-Review Synopsis of Key Ideas and Publications and Bibliography for Explainable AI, Technical Report, Defense Advanced Research Projects Agency (DARPA), Arlington, VA, USA, 2019.

[26] J. Zhou, A.H. Gandomi, F. Chen, A. Holzinger, Evaluating the quality of machine learning explanations: a survey on methods and metrics, Electronics 10 (5) (2021) 593. https://doi.org/10.3390/electronics10050593

[27] S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Muller, G. Montavon, Toward explainable artificial intelligence for regression models: a methodological perspective, IEEE Signal Process. Mag. 39 (4) (2022) 40–58. https://doi.org/10.1109/MSP.2022.3153277

[28] A.M. Salih, Z. Raisi-Estabragh, I.B. Galazzo, P. Radeva, S.E. Petersen, K. Lekadir, G. Menegaz, A perspective on explainable artificial intelligence methods: SHAP and LIME, Adv. Intell. Syst. 7 (1) (2025) 2400304. https://doi.org/10.1002/aisy.202400304

[29] Y. Liu, S. Khandagale, S. Khandagale, C. White, W. Neiswanger, Synthetic benchmarks for scientific research in explainable machine learning, in: J. Vanschoren, S. Yeung (Eds.), Proceedings of the Neural Information Processing Systems - Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021), 1, 2021, pp. 1–14. https://doi.org/10.48550/arXiv.2106.12543

[30] Z. Qian, T. Callender, B. Cebere, S.M. Janes, N. Navani, M. van der Schaar, Synthetic data for privacy-preserving clinical risk prediction, Sci. Rep. 14 (1) (2024) 25676. https://doi.org/10.1038/s41598-024-72894-y

[31] D.R. Padariya, A privacy-preserving framework for generative model-driven synthetic datasets, AAAI 39 (28) (2025) 29289–29290. https://doi.org/10.1609/aaai.v39i28.35222

[32] M.U. Ahmed, S. Barua, S. Begum, M.R. Islam, R.O. Weber, When a CBR in hand better than twins in the bush, in: P. Reuss, J. Schönborn (Eds.), Proceedings of the 4th Workshop on XCBR: Case-based Reasoning for the Explanation of Intelligent Systems, 3389 of CEUR Workshop Proceedings, CEUR-WS.org, Nancy, France, 2022, pp. 141–152. https://ceur-ws.org/Vol-3389/#XCBR99.

[33] D.G. Bonett, Confidence interval for a coefficient of quartile variation, Comput. Stat. Data Anal. 50 (11) (2006) 2953–2957. https://doi.org/10.1016/j.csda.2005.05.007

[34] E. Štrumbelj, I. Kononenko, An efficient explanation of individual classifications using game theory, J. Mach. Learn. Res. 11 (1) (2010) 1–18.

[35] A.J. Cook, G. Tanner, European Airline Delay Cost Reference Values, Technical Report, University of Westminster, London, UK, 2015.

[36] M. Lukacs, Cost of Delay Estimates, Technical Report, Federal Aviation Administration, Washington, DC, USA, 2020.

[37] R. Dalmau, F. Ballerini, H. Naessens, S. Belkoura, S. Wangnick, An explainable machine learning approach to improve take-off time predictions, J. Air Transp. Manag. 95 (2021) 102090. https://doi.org/10.1016/j.jairtraman.2021.102090

[38] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS), NIPS'18, 2018, pp. 9525–9536. https://doi.org/10.48550/arXiv.1810.03292

[39] J. Marques-Silva, A. Ignatiev, Delivering trustworthy AI through formal XAI, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 36(11), 2022, pp. 12342–12350. https://doi.org/10.1609/aaai.v36i11.21499

[40] X. Huang, J. Marques-Silva, The Inadequacy of Shapley Values for Explainability, arXiv preprint (2023). https://doi.org/10.48550/arXiv.2302.08160

[41] Y.-S. Lin, W.-C. Lee, Z.B. Celik, What do you see? Evaluation of explainable artificial intelligence (XAI) interpretability through neural backdoors, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1027–1035. https://doi.org/10.1145/3447548.3467213

[42] A. Rosenfeld, Better metrics for evaluating explainable artificial intelligence, in: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2021), AAMAS '21, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2021, pp. 45–50.

[43] J. van der Waa, E. Nieuwburg, A. Cremers, M. Neerincx, Evaluating XAI: a comparison of rule-based and example-based explanations, Artif. Intell. 291 (2021) 103404. https://doi.org/10.1016/j.artint.2020.103404

[44] Y. Rong, T. Leemann, T.-T. Nguyen, L. Fiedler, P. Qian, V. Unhelkar, T. Seidel, G. Kasneci, E. Kasneci, Towards human-centered explainable AI: a survey of user studies for model explanations, IEEE Trans. Pattern Anal. Mach. Intell. 46 (4) (2024) 2104–2122. https://doi.org/10.1109/TPAMI.2023.3331846

[45] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, in: Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS), NIPS'16, 2016, pp. 3395–3403. https://doi.org/10.48550/arXiv.1605.09304

[46] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking clever hans predictors and assessing what machines really learn, Nat. Commun. 10 (1) (2019) 1096. https://doi.org/10.1038/s41467-019-08987-4

[47] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (7) (2015) e0130140. https://doi.org/10.1371/journal.pone.0130140

[48] L.S. Shapley, A value for n-person games, in: H.W. Kuhn, A.W. Tucker (Eds.), Contributions to the Theory of Games, II, Princeton University Press, 1953, pp. 307–318.

[49] L. Arras, A. Osman, W. Samek, CLEVR-XAI: a benchmark dataset for the ground truth evaluation of neural network explanations, Inf. Fusion 81 (2022) 14–40. https://doi.org/10.1016/j.inffus.2021.11.008

[50] S. Hooker, D. Erhan, P.-J. Kindermans, B. Kim, A benchmark for interpretability methods in deep neural networks, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 9737–9748.

[51] G. Montavon, Gradient-based vs. propagation-based explanations: an axiomatic comparison, in: W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, K.-R. Müller (Eds.), Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, 11700 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2019, pp. 253–265. https://doi.org/10.1007/978-3-030-28954-6_13

[52] F. Yang, M. Du, X. Hu, Evaluating Explanation without Ground Truth in Interpretable Machine Learning, arXiv preprint (arXiv:1907.06831v2 [cs.LG]) (2019). https://doi.org/10.48550/arXiv.1907.06831

[53] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, A. Preece, Sanity checks for saliency metrics, in: Proceedings of the AAAI Conference on Artificial Intelligence, 34, 2020, pp. 6021–6029. https://doi.org/10.1609/aaai.v34i04.6064

[54] A. Ignatiev, F. Pereira, N. Narodytska, J. Marques-Silva, A SAT-based approach to learn explainable decision sets, in: D. Galmiche, S. Schulz, R. Sebastiani (Eds.), Automated Reasoning, 10900 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2018, pp. 627–645. https://doi.org/10.1007/978-3-319-94205-6_41

[55] A. Ignatiev, N. Narodytska, J. Marques-Silva, On Validating, Repairing and Refining Heuristic ML Explanations, arXiv preprint (arXiv:1907.02509v1 [cs.LG]) (2019). https://doi.org/10.48550/arXiv.1907.02509

[56] N. Narodytska, A. Shrotri, K.S. Meel, A. Ignatiev, J. Marques-Silva, M. Janota, I. Lynce, Assessing heuristic machine learning explanations with model counting, in: Theory and Applications of Satisfiability Testing - SAT 2019, 11628 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2019, pp. 267–278. https://doi.org/10.1007/978-3-030-24258-9_19

[57] X. Cui, J.M. Lee, J.P. Hsieh, An integrative 3C evaluation framework for explainable artificial intelligence, in: Proceedings of the Americas Conference on Information Systems (AMCIS), 10, 2019, pp. 1–10.

[58] B. Kim, M. Wattenberg, J. Gilmer, C.J. Cai, J. Wexler, F. Viégas, R. Sayres, Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV), in: Proceedings of the 35th International Conference on Machine Learning (ICML), PMLR, 2018, pp. 1–18. https://doi.org/10.48550/arXiv.1711.11279

[59] D. Alvarez-Melis, T.S. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 7786–7795.

[60] A. Balagopalan, H. Zhang, K. Hamidieh, T. Hartvigsen, F. Rudzicz, M. Ghassemi, The road to explainability is paved with bias: measuring the fairness of explanations, in: Proceedings of the 2022 Conference on Fairness, Accountability, and Transparency (FAT*), FAccT '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1194–1206. https://doi.org/10.1145/3531146.3533179

[61] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. Van Keulen, C. Seifert, From anecdotal evidence to quantitative evaluation meth-

ods: a systematic review on evaluating explainable AI, ACM Comput. Surv. 55 (13s) (2023) 1–42. https://doi.org/10.1145/3583558
[62] Z. Chen, V. Subhash, M. Havasi, W. Pan, F. Doshi-Velez, What Makes a Good Explanation?: A Harmonized View of Properties of Explanations, arXiv preprint (arXiv:2211.05667v3 [cs]) (2024). https://doi.org/10.48550/arXiv.2211.05667
[63] A.-p. Nguyen, M.R. Martínez, On Quantitative Aspects of Model Interpretability, arXiv preprint (arXiv:2007.07584v1 [cs.LG]) (2020). https://doi.org/10.48550/arXiv.2007.07584
[64] J. Chen, L. Song, M.J. Wainwright, M.I. Jordan, Learning to explain: an information-theoretic perspective on model interpretation, in: J.G. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning (ICML), 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 882–891. https://doi.org/10.48550/arXiv.1802.07814
[65] U. Bhatt, P. Ravikumar, J.M.F. Moura, Building human-machine trust via interpretability, Proc. AAAI Conf. Artif. Intell. 33 (01) (2019) 9919–9920. https://doi.org/10.1609/aaai.v33i01.33019919
[66] S.M. Xie, S. Ermon, Reparameterizable subset sampling via continuous relaxations, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Macao, China, 2019, pp. 3919–3925. https://doi.org/10.24963/ijcai.2019/544
[67] B. Bai, J. Liang, G. Zhang, H. Li, K. Bai, F. Wang, Why Attentions May Not Be Interpretable?, arXiv preprint (arXiv:2006.05656v4 [stat.ML]) (2021). https://doi.org/10.48550/arXiv.2006.05656
[68] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision - ECCV 2014, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2014, pp. 818–833. https://doi.org/10.1007/978-3-319-10590-1_53
[69] Y. Rong, T. Leemann, V. Borisov, G. Kasneci, E. Kasneci, A consistent and efficient evaluation strategy for attribution methods, in: Proceedings of the 39th International Conference on Machine Learning (ICML), 2022, pp. 1–26. https://doi.org/10.48550/arXiv.2202.00449
[70] L. Sixt, M. Granz, T. Landgraf, When explanations lie: why many modified BP attributions fail, in: Proceedings of the 37th International Conference on Machine Learning (ICML), PMLR, 2020, pp. 9046–9057. ISSN: 2640–3498. https://doi.org/10.48550/arXiv.1912.09818
[71] A. Hedström, L. Weber, D. Krakowczyk, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, M.M.C. Höhne, Quantus: an explainable AI toolkit for responsible evaluation of neural network explanations and beyond, J. Mach. Learn. Res. 24 (34) (2023) 1–11. https://doi.org/10.48550/arXiv.2202.06861
[72] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, D.A. Keim, Towards a rigorous evaluation of XAI methods on time series, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE, Seoul, Korea (South), 2019, pp. 4197–4201. https://doi.org/10.1109/ICCVW.2019.00516
[73] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning, 70 of *ICML'17*, JMLR.org, Sydney, NSW, Australia, 2017, pp. 3319–3328.
[74] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K.T. Schütt, S. Dähne, D. Erhan, B. Kim, The (un)reliability of saliency methods, in: W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, K.-R. Müller (Eds.), Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, 11700 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2019, pp. 267–280. https://doi.org/10.1007/978-3-030-28954-6_14
[75] R.R. Hoffman, S.T. Mueller, G. Klein, J. Litman, Metrics for Explainable AI: Challenges and Prospects, arXiv preprint (arXiv:1812.04608v2 [cs.AI]) (2019). https://doi.org/10.48550/arXiv.1812.04608
[76] A.R. Troncoso-García, M. Martínez-Ballesteros, F. Martínez-Álvarez, A. Troncoso, A new approach based on association rules to add explainability to time series forecasting models, Inf. Fusion 94 (2023) 169–180. https://doi.org/10.1016/j.inffus.2023.01.021
[77] S. Deng, C. Aldrich, X. Liu, F. Zhang, Explainability in reservoir well-logging evaluation: comparison of variable importance analysis with shapley value regression, SHAP and LIME, IFAC-PapersOnLine 58 (22) (2024) 66–71. https://doi.org/10.1016/j.ifacol.2024.09.292
[78] R. Roscher, B. Bohn, M.F. Duarte, J. Garcke, Explainable machine learning for scientific insights and discoveries, IEEE Access 8 (2020) 42200–42216. https://doi.org/10.1109/ACCESS.2020.2976199
[79] A. Binder, M. Bockmayr, M. Hägele, S. Wienert, D. Heim, K. Hellweg, M. Ishii, A. Stenzinger, A. Hocke, C. Denkert, K.-R. Müller, F. Klauschen, Morphological and molecular breast cancer profiling through explainable machine learning, Nat. Mach. Intell. 3 (4) (2021) 355–366. https://doi.org/10.1038/s42256-021-00303-4
[80] J.V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, M. Srivastava, How can I explain this to you? An empirical study of deep neural network explanation methods, in: Advances in Neural Information Processing Systems (NeurIPS 2020), 33, 2020, pp. 4211–4222.
[81] K. Fauvel, V. Masson, É. Fromont, A performance-explainability framework to benchmark machine learning methods: application to multivariate time series classifiers, in: Proceedings of the Workshop on XAI co-located with the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI), 2020, pp. 1–8. https://doi.org/10.48550/arXiv.2005.14501
[82] A.A. Ismail, M. Gunady, H.C. Bravo, S. Feizi, Benchmarking deep learning interpretability in time series predictions, in: Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS), 2020, pp. 1–32. https://doi.org/10.48550/arXiv.2010.13924
[83] J. DeYoung, S. Jain, N.F. Rajani, E. Lehman, C. Xiong, R. Socher, B.C. Wallace, ERASER: a benchmark to evaluate rationalized NLP models, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 4443–4458. https://doi.org/10.18653/v1/2020.acl-main.408
[84] E. Amparore, A. Perotti, P. Bajardi, To trust or not to trust an explanation: using LEAF to evaluate local linear XAI methods, PeerJ Comput. Sci. 7 (2021) e479. https://doi.org/10.7717/peerj-cs.479
[85] C. Agarwal, S. Krishna, E. Saxena, M. Pawelczyk, N. Johnson, I. Puri, M. Zitnik, H. Lakkaraju, OpenXAI: towards a transparent evaluation of model explanations, Adv. Neural Inf. Process. Syst. 35 (2022) 15784–15799.
[86] C. Agarwal, O. Queen, H. Lakkaraju, M. Zitnik, Evaluating explainability for graph neural networks, Sci. Data 10 (1) (2023) 144. Publisher: Nature Publishing Group. https://doi.org/10.1038/s41597-023-01974-x
[87] X. Li, M. Du, J. Chen, Y. Chai, H. Lakkaraju, H. Xiong, M4: a unified XAI benchmark for faithfulness evaluation of feature attribution methods across metrics, modalities and models, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Curran Associates Inc., Red Hook, NY, USA, 2023, pp. 1630–1643.
[88] J.R. Lee, S. Emami, M.D. Hollins, T.C.H. Wong, C.I. Villalobos Sánchez, F. Toni, D. Zhang, A. Dejl, XAI-Units: benchmarking explainability methods with unit tests, in: Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency, FAccT '25, Association for Computing Machinery, New York, NY, USA, 2025, pp. 2892–2905. https://doi.org/10.1145/3715275.3732186
[89] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: automated decisions and the GDPR, Harv. J. Law Technol. 31 (2) (2018) 841–887. https://doi.org/10.2139/ssrn.3063289
[90] M.M. Richter, R.O. Weber, Case-Based Reasoning: A Textbook, Springer, Berlin, Heidelberg, Berlin, Heidelberg, 2013. https://doi.org/10.1007/978-3-642-40167-1
[91] E.M. Kenny, M.T. Keane, Explaining deep learning using examples: optimal feature weighting methods for twin systems using post-hoc, explanation-by-example in XAI, Knowledge-Based Syst. 233 (2021) 107530. https://doi.org/10.1016/j.knosys.2021.107530
[92] C. Nugent, P. Cunningham, A case-based explanation system for black-box systems, Artif. Intell. Rev. 24 (2) (2005) 163–178. https://doi.org/10.1007/s10462-005-4609-5
[93] H. Koolen, I. Coliban, Flight Progress Messages Document, Technical Report, EUROCONTROL, Brussels, Belgium, 2020.
[94] D. Rankin, M. Black, R. Bond, J. Wallace, M. Mulvenna, G. Epelde, Reliability of supervised machine learning using synthetic data in health care: model to preserve privacy for data sharing, JMIR Med. Inform. 8 (7) (2020) e18910. https://doi.org/10.2196/18910
[95] A.F. Kalay, Generating Synthetic Data with Locally Estimated Distributions for Disclosure Control, arXiv preprint (arXiv:2210.00884v2 [stat.CO]) (2025). https://doi.org/10.48550/arXiv.2210.00884
[96] Y. Zhang, J.L. Dong, B. Xue, Y. Xiong, S. Gupta, M.V. Segbroeck, N. Shara, P. McGarvey, Exploring the utilization of synthetic data in unsupervised clustering for opioid misuse analysis, AMIA Annu. Symp. Proc. 2024 (2025) 1313–1322.
[97] B. Li, P. Wang, P. Sun, R. Meng, J. Zeng, G. Liu, A model for determining the optimal decommissioning interval of energy equipment based on the whole life cycle cost, Sustainability 15 (6) (2023) 5569. https://doi.org/10.3390/su15065569
[98] J. Paparrizos, F. Yang, H. Li, Bridging the Gap: A Decade Review of Time-Series Clustering Methods, arXiv preprint (arXiv:2412.20582v1 [cs.LG]) (2024). https://doi.org/10.48550/arXiv.2412.20582
[99] S. Aghabozorgi, A. Seyed Shirkhorshidi, T. Ying Wah, et al., Time-series clustering—A decade review, Inf. Syst. 53 (2015) 16–38. https://doi.org/10.1016/j.is.2015.04.007
[100] A. Gonsek, M. Jeschke, S. Rönnau, O.J.N. Bertrand, From paths to routes: a method for path classification, Front. Behav. Neurosci. 14 (2021). Publisher: Frontiers. https://doi.org/10.3389/fnbeh.2020.610560
[101] T.S. Madhulatha, An overview on clustering methods, IOSR J. Eng. 02 (04) (2012) 719–725. https://doi.org/10.9790/3021-0204719725
[102] C. Yuan, H. Yang, Research on k-value selection method of k-means clustering algorithm, J. — Multidiscip. Sci. J. 2 (2) (2019) 226–235. https://doi.org/10.3390/j2020016
[103] E. Schubert, Stop using the elbow criterion for k-means and how to choose the number of clusters instead, ACM SIGKDD Explor. Newsl. 25 (1) (2023) 36–42. https://doi.org/10.1145/3606274.3606278
[104] T. Calinski, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat. - Theory Methods 3 (1) (1974) 1–27. https://doi.org/10.1080/03610927408827101
[105] C.A. Sugar, G.M. James, Finding the number of clusters in a dataset: an information-theoretic approach, J. Am. Stat. Assoc. 98 (463) (2003) 750–763. https://doi.org/10.1198/016214503000000666
[106] I.J. Good, Explicativity: a mathematical theory of explanation with statistical applications, Proc. R. Soc. Lond. A 354 (1678) (1977) 303–330. https://doi.org/10.1098/rspa.1977.0069
[107] J. Lin, Divergence measures based on the shannon entropy, IEEE Trans. Inf. Theory 37 (1) (1991) 145–151. https://doi.org/10.1109/18.61115
[108] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Francisco California USA, 2016, pp. 785–794. https://doi.org/10.1145/2939672.2939785

[109] C. Huertas, Gradient Boosting Trees and Large Language Models for Tabular Data Few-Shot Learning, arXiv preprint (arXiv:2411.04324v1 [cs.LG]) (2024). https://doi.org/10.48550/arxiv.2411.04324
[110] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, in: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), 30, 2017, pp. 1–9.
[111] M. Claesen, B. De Moor, Hyperparameter Search in Machine Learning, arXiv preprint (arXiv:1502.02127v2 [cs.LG]) (2015). https://doi.org/10.48550/arxiv.1502.02127
[112] R.K. Halder, M.N. Uddin, M.A. Uddin, S. Aryal, A. Khraisat, Enhancing k-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications, J. Big Data 11 (1) (2024) 113. https://doi.org/10.1186/s40537-024-00973-y
[113] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proceedings of the 34th International Conference on Machine Learning (ICML), ICML'17, JMLR.org, Sydney, NSW, Australia, 2017, pp. 3145–3153. https://doi.org/10.48550/arXiv.1704.02685
[114] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (4) (2002) 422–446. https://doi.org/10.1145/582415.582418
[115] R. Busa-Fekete, G. Szarvas, T. Élteto, B. Kégl, An apple-to-apple comparison of learning-to-rank algorithms in terms of normalized discounted cumulative gain, in: Proceedings of the Workshop on Preference Learning: Problems and Applications in AI Co-located with the 20th European Conference on Artificial Intelligence (ECAI), 242, Ios Press, Montpellier, France, 2012, pp. 1–7.
[116] Y. Wang, L. Wang, Y. Li, D. He, T.-Y. Liu, A theoretical analysis of NDCG type ranking measures, in: Proceedings of the 26th Annual Conference on Learning Theory (COLT), 30, PMLR, 2013, pp. 25–54.
[117] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830.
[118] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, second ed., Munich, Germany, 2022.
[119] Z. Botta-Dukát, Quartile coefficient of variation is more robust than CV for traits calculated as a ratio, Sci. Rep. 13 (1) (2023) 4671. https://doi.org/10.1038/s41598-023-31711-8
[120] S. Krishna, T. Han, A. Gu, J. Pombra, S. Jabbari, S. Wu, H. Lakkaraju, The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective, arXiv preprint (arXiv:2202.01602v3 [cs.LG]) (2022). https://doi.org/10.48550/arXiv.2202.01602
[121] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, J.K. Su, This looks like that: deep learning for interpretable image recognition, in: Advances in Neural Information Processing Systems, 32, Curran Associates, Inc., 2019, pp. 1–12.
[122] M.T. Keane, E.M. Kenny, The Twin-System Approach as One Generic Solution for XAI: An Overview of ANN-CBR Twins for Explaining Deep Learning, arXiv preprint (arXiv:1905.08069v1 [cs.AI]) (2019). https://doi.org/10.48550/arXiv.1905.08069
[123] D. Leake, Z. Wilkerson, D.J. Crandall, Combining case-based reasoning with deep learning: context and ongoing case feature learning research, in: Neuro-Symbolic Learning and Reasoning in the Era of Large Language Models, 2023, pp. 1–5.