MASALA: Model-Agnostic Surrogate Explanations by Locality Adaptation

Saif Anwar[1,*], Nathan Griffiths[1], Abhir Bhalerao[1] and Thomas Popham[2]

[1]Department of Computer Science, University of Warwick, United Kingdom
[2]School of Engineering, University of Warwick, United Kingdom

Abstract

Existing local Explainable AI (XAI) methods, such as LIME, select a region of the input space in the vicinity of a given input instance, for which they approximate the behaviour of a model using a simpler and more interpretable surrogate model. The size of this region is often controlled by a user-defined locality hyperparameter. In this paper, we demonstrate the difficulties associated with defining a suitable locality size to capture impactful model behaviour, as well as the inadequacy of using a single locality size to explain all predictions. We propose a novel method, MASALA, for generating explanations, which automatically determines the appropriate local region of impactful model behaviour for each individual instance being explained. MASALA approximates the local behaviour used by a complex model to make a prediction by fitting a linear surrogate model to a set of points which experience similar model behaviour. These points are found by clustering the input space into regions of linear behavioural trends exhibited by the model. We compare the *fidelity* and *consistency* of explanations generated by our method with existing local XAI methods, namely LIME and CHILLI. Experiments on the PHM08 and MIDAS datasets show that our method produces more faithful and consistent explanations than existing methods, without the need to define any sensitive locality hyperparameters.

Keywords

Exlainable AI (XAI), Interpretable Machine Learning, Explanation, Model-Agnostic, Post-Hoc, Local Linear Modelling

1. Introduction

Many Machine Learning (ML) methods are treated as *black-boxes* because of their complex and often incomprehensible behaviour. As a result, there is tentative adoption in high-risk domains, such as healthcare, finance, and defence. There is a demand from stakeholders to establish trust in a model, since an incorrect decision may have serious consequences [1, 2, 3]. Explainable AI (XAI) methods aim to provide explanations for the predictions produced by a model and make transparent its behaviours [4]. In this paper, we define an *explanation* to be an interpretable representation of a *base model*'s decision-making process, such as in the form of feature importance scores or a set of decision rules.

Generally, XAI techniques can be divided into inherently-interpretable models and post-hoc methods. The former involves developing model architectures which are interpretable by

HI-AI@KDD, Human-Interpretable AI Workshop at the KDD 2024, 26th of August 2024, Barcelona, Spain

saif.anwar@warwick.ac.uk (S. Anwar)
0009-0000-5763-4416 (S. Anwar); 0000-0002-6406-8632 (N. Griffiths); 0000-0001-8830-329X (A. Bhalerao); 0000-0002-4567-9534 (T. Popham)

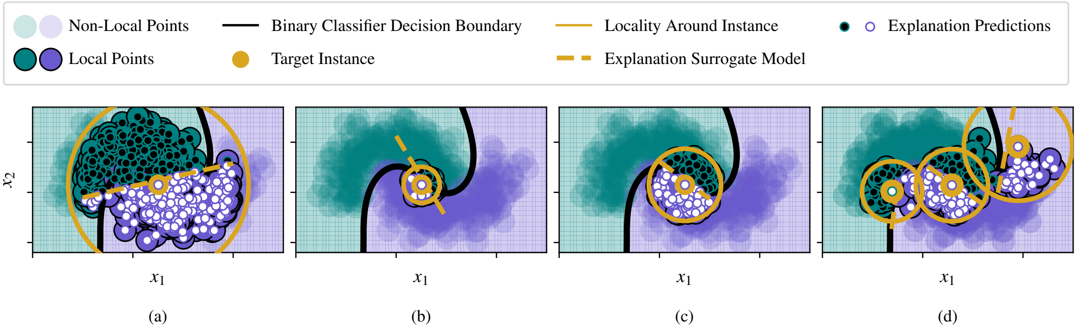


Figure 1: a) A surrogate fit within a locality which is too large leading to an inaccurate linear approximation of the non-linear decision boundary b) A surrogate fit within a locality which is too small and therefore, captures irregularities rather than the impactful linear trend. c) An explanation generated within an appropriate sized locality, which represents the true model behaviour in the immediate vicinity of the target instance. d) Appropriate explanations generated for three instances using localities of different sizes

design and do not require extensive effort to understand the reasoning for a given output [5, 6]. However, it is generally agreed that limiting complexity, for the sake of interpretability, may hinder performance when compared to more complex black-box models [7, 8].

Post-hoc XAI methods generate explanations for pre-trained black-box models, often in a model-agnostic manner [9]. Popular methods typically explain predictions through feature importance scores, either on a global or local scale [10, 11, 12]. Global methods explain general model behaviour for all datapoints, whereas local methods explain the model behaviour used to make a specific prediction. A local explanation is constrained to a *locality*, namely a region of the input space surrounding the instance for which the prediction is being explained, which we call the *target instance*. Some local XAI methods fit an inherently interpretable surrogate model in a locality around a target instance, where the interpretation of the surrogate model is the explanation, such as the coefficients of a regression model or rules established in a decision tree.

LIME [10] is a popular approach that fits a surrogate model to perturbations of a target instance that are generated by randomly sampling a Gaussian kernel centered around the target instance. The width of the kernel, which is manually defined by the user and is fixed for all explanations, controls the expanse of the perturbations and therefore the size of the explanation locality. It has been shown that a surrogate model fit to perturbations sampled from an inappropriately sized Gaussian kernel may not be representative of the base models training data and therefore does not represent the true base model behaviour [13]. CHILLI [14] is an adaptation of LIME which addresses some of these issues. Perturbations are generated according to the distribution of the model training data in the vicinity of the target instance. However, CHILLI also requires the user to define the locality of each explanation. The locality size must be defined in such a way that it is not too large, where the explanation ignores model intricacies, and not too small, where the explanation focuses on anomalous fluctuations rather than more dominant behavioural trends. This is illustrated in Figure 1, which shows how the locality size affects the behaviour of the surrogate model and how a fixed locality may not be appropriate for all instances. Since LIME and CHILLI use non-deterministic perturbation generation methods, explanations for the same prediction may differ, leading to a lack of consistency that undermines the trustworthiness of the explanation method [15, 16, 17].

In this paper, we propose **M**odel-**A**gnostic **S**urrogate expl**A**nations by **L**ocality **A**daptation (MASALA), a novel post-hoc XAI method that automatically finds the impactful local model behaviour surrounding a target instance. MASALA fits a Multivariate Linear Regression (MLR) surrogate model to a set of points that experience the same linear behaviour as the target instance, which it obtains by automatically detecting the linear regions of model behaviour across the input domain. The coefficients of the MLR represent feature relationships towards the target distribution. Since MASALA generates explanations using a deterministic clustering, explanations for the same instance are identical and therefore are consistent. Using the PHM08 and MIDAS datasets, we qualitatively and quantitatively demonstrate the ability of MASALA to generate explanations with higher fidelity and consistency than those produced by LIME and CHILLI. Our source code and data are available through the following repository: https://github.com/saiffanwar/MASALA

2. Related Work

Existing works have attempted to address locality issues by clustering similar points to fit linear surrogate models. Zafar et al. [18] propose DLIME, which uses agglomerative Hierarchical Clustering to divide the training data into groups of similar points according to their Euclidean distance across all features. Points that are clustered together may not experience the same model behaviour since they may be close in some feature dimensions, but distant in others and therefore may not experience similar model behaviour across all features.

It has been shown that LIME generates inconsistent explanations when using a locality size that is too small, since explanations may focus on irregularities introduced by the randomly sampled perturbations. Gaudel et al. propose s-LIME [19], which generates perturbations whose distance is proportional to the magnitude of the selected kernel width. However, this still requires the locality to be manually defined. Local Surrogate [17] avoids manual locality definition by generating perturbations around the decision boundary closest to the instance being explained, therefore approximating the model behaviour that led to the prediction. However, this may not be applicable to a regression problem where there is no decision boundary. In ALIME [20], an autoencoder is trained as a weighting function used to decide whether perturbations are local to the target instance. Although this leads to more consistent explanations, the threshold for discarding points must be manually defined and is effectively equivalent to the kernel width hyperparameter in LIME.

3. Methodology

The goal of MASALA is to fit a linear surrogate model to a set of points that experience similar model behaviour to a specified target instance. We first formalise the problem and then describe the details of the method.

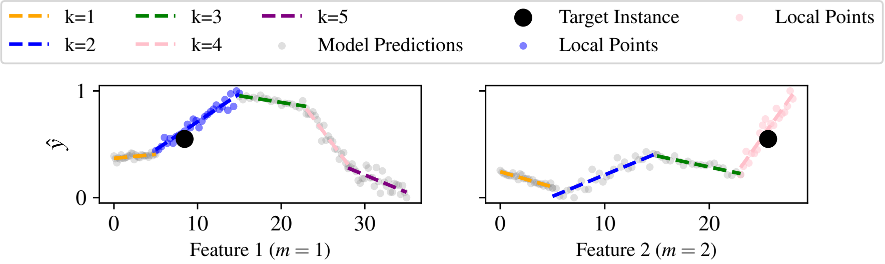


Figure 2: Distribution of 2 features from the same dataset against the model predictions and clustered into linear regions.

3.1. Problem Definition

Consider a black-box base model f, which maps an M-dimensional input vector $\mathbf{x} \in \mathcal{X}$, to a scalar output $y \in \mathcal{Y}$, and is trained on a dataset $\mathcal{D}$. The prediction for a given target instance $\mathbf{x}_i$ is explained by training a MLR surrogate model g_i on a subset of the training data $\mathcal{D}_i \in \mathcal{D}$, with target values being the predictions on $\mathcal{D}_i$ from the base model f. The linear coefficients of the MLR directly indicate the contribution of each feature towards the prediction. The selection of instances to include in $\mathcal{D}_i$ defines the locality of the explanation, since only the model behaviour used to make predictions for those instances will be approximated. Only instances that have similar feature values and experience similar model behaviour to the target instance should be included in $\mathcal{D}_i$. However, identifying a set $\mathcal{D}_i$ for a given $\mathbf{x}_i$ is non-trivial, since the model behaviour may vary across the feature space. MASALA identifies an appropriate subset $\mathcal{D}_i$ for training a surrogate MLR model, by finding all instances that share the same region of linearity in the distribution of model predictions as the target instance $\mathbf{x}_i$. Since we assume that the base model behaviour is locally linear for some region around all data instances, we propose exhaustively clustering the distribution of each input feature against the model predictions into regions of linearity. The set of K identified linear regions, or clusters, for a given feature dimension m, is denoted as $\mathcal{C}^m$, such that $\mathcal{C}^m = \{c_k^m | \forall k \in K\}$. Once a clustering has been obtained, the clusters within which a target instance $\mathbf{x}_i$ falls, in each feature dimension, can be identified. This is denoted as $c_k^m(\mathbf{x}_i)$ which indicates that in the distribution of feature m, the target instance falls within linear region k. The subset of instances used to train the surrogate g_i is then defined as the set of instances which share the same linear region in each feature dimension as the instance $\mathbf{x}_i$, as formulated in Equation 1.

$$\mathcal{D}_i = \{\mathbf{x}_j \in \mathcal{X} | c_k^m(\mathbf{x}_j) = c_k^m(\mathbf{x}_i), \forall m \in M\} \tag{1}$$

Figure 2 shows the distributions of 2 features from the same dataset, against the respective model predictions, along with a target instance to be explained. Each feature has been clustered into a different number of linear regions with the target instance falling within the blue and pink regions for features $m = 1$ and $m = 2$ respectively. The set $\mathcal{D}_i$ contains the points that also fall within both the blue and pink clusters. Since $\mathcal{D}_i$ does not change for a given instance, explanations generated for the same instance are identical, thus preserving consistency.

3.2. Local Linear K-Medoids Clustering

We now present our method for identifying regions of linear model behaviour in the distribution of an input feature and the predictions from the base model. We consider each feature dimension individually, and will therefore omit the superscript m in the remainder of this section. For example, $\mathcal{X}^m$ will be denoted as $\mathcal{X}$ and c_k^m will be denoted as c_k.

3.2.1. Pairwise Dissimilarity

We apply an adapted K-medoids algorithm [21] which clusters datapoints based on their pairwise Euclidean distance. However, points that are close in the feature space, do not necessarily experience the same linear model behaviour. We introduce a custom distance measure in the form of pairwise dissimilarity $\Delta_{\mathcal{X}}$, which compares the local linearity, feature value, and local density of points. For each datapoint, x_i, a weighted Local Linear Regression (LLR) is performed on all points within its neighbourhood, $N(x_i)$, which is defined using a distance threshold in the feature space. Although this may be seen as defining a similar threshold to the kernel width locality parameter in LIME, the final explanation is much more robust to changes in the threshold since the weighted LLR automatically considers closer points with greater importance. The dissimilarity between two points x_i and x_j is calculated as

$$\Delta_{\mathcal{X}}(i,j) = ||\mathbf{w}_i - \mathbf{w}_j||_2 + D(x_i, x_j) + ||N(x_i)| - |N(x_j)||, \tag{2}$$

where $\mathbf{w}_i$ represents the vector of LLR model parameters for point x_i, and $|N(x_i)|$ denotes the number of points in the neighbourhood of x_i. The second term $D(i,j)$ is the difference in feature values of x_i and x_j, which may be a custom measure dependent on the type of feature. Including this distance ensures that points with similar neighbourhood trends that are in different regions of the feature space, are not clustered together. The final term allows for the inclusion of local data density since two points may share a similar linear neighbourhood trend, but one neighbourhood may exhibit more sparsity than the other. A linear model fit to a sparse neighbourhood may be skewed by anomalous datapoints, and therefore should be considered with more caution. It should be noted that all terms are normalised in the range [0,1] such that they have equal contribution towards the dissimilarity measure. Points with the smallest values in $\Delta_{\mathcal{X}}$ will be close together in the feature space, have similar LLR model parameters, and have similar neighbourhood density.

Points are clustered according to their pairwise dissimilarity using the K-medoids algorithm [21]. To allow for a deterministic clustering, medoids are initialised by evenly distributing K medoids across the sorted values from the data. A Linear Regression (LR) model is fit to the points within each cluster, where a_k and b_k are the LR parameters within cluster k. The cost, J, for a clustering C is calculated according to Equation 3.

$$J(C) = \sum_{k=1}^{K} \text{RMSE}(\{a_k \cdot x + b_k, f(x) | \forall x \in C_k\}) \tag{3}$$

Thus, the cost is the sum of the RMSEs between predictions from the LR model and base model predictions within each cluster. Rather than randomly assigning a new medoid for a cluster, the clustering cost is calculated when each point is the medoid. The medoid which gives the lowest clustering cost is selected. This is repeated for all clusters with the algorithm halting when the clustering cost no longer changes after optimising all clusters and is also a deterministic process. A lower clustering cost reflects an ensemble of linear proxy models that is more faithful to the behaviour of the base model.

3.2.2. Automatically defining K

It may seem that increasing the number of clusters K, would generate a more faithful ensemble of linear models. However, doing so leads to clusters with smaller coverage which may be overfit to erroneous behaviour, rather than to more impactful general linear trends. This effect can be worsened in sparse regions of the input space. The relationship between an input feature and model predictions will vary across features and datasets, and therefore the appropriate number of linear regions to cluster also varies. We propose an algorithm that automatically finds a suitable value for K, given a set of constraints, such that clusters must not overlap in the feature space, must contain a sufficient number of datapoints, and must cover a suitable range of values in the feature space.

If a cluster is wholly contained within another cluster, the larger cluster will adopt all datapoints of the smaller cluster. If two clusters overlap each other, two new clusters replace them by dividing the points at the midpoint of the overlapping clusters' intersection. Once all clusters have been separated, they each occupy a unique range of values in the feature space.

To avoid skewing to anomalous or non-significant behaviours, all clusters should contain sufficient datapoints, and therefore the data sparsity of each cluster is checked. The sparsity of a cluster is defined as the ratio between the number of points it contains and the number of points in the largest cluster, C_L, leading to a dynamic sparsity measure that is relative to the current clustering. If the current largest cluster contains a relatively small number of datapoints from the entire dataset, the sparsity measure considers that all clusters are generally small and there may be intricate relationships within the data. The criteria for deciding whether cluster C_k is sparse is shown in Equation 4,

$$\frac{|C_k|}{|C_L|} < \frac{1}{N^2} \sum_{i,j \in N} |x_i - x_j|, \tag{4}$$

where N is the total number of samples in the dataset. The right-hand side of the formulation is the average pairwise distance between all points in the dataset and is used as a sparsity threshold since it provides a measure of the average density of the data. If the sparsity of a cluster falls below the threshold, it is combined with a neighbouring cluster in the feature space. The combination that gives the lowest clustering cost when merging the sparse cluster with each of its neighbours, is selected for the new clustering. Similarly, the coverage of each cluster is also checked, where coverage is calculated as the percentage of the input space occupied. If the coverage falls below the same threshold used for sparsity, it is combined with a neighbouring cluster using the same protocol as for a sparse cluster.

To obtain a clustering, we start with some arbitrarily large K. An initial clustering is generated by selecting K random medoids and assigning points to the cluster for which the medoid is most similar according to $\Delta_{\mathcal{X}}$. The cost of this clustering is calculated using Equation 3. A new clustering is generated by randomly selecting a new medoid for a random cluster and reassigning all points. If the new clustering is of lower cost, it replaces the previous clustering. This is repeated until the cost of the clustering is unchanged by selecting a new medoid. The clustering is then checked against the constraints and modified if necessary, which may lead to a change in the number of clusters. If so, K is redefined as the new number of clusters and a new clustering is obtained in the same manner as outlined above, to find the lowest cost clustering for the new value of K. This process is repeated until the number of clusters does not change after satisfying the constraints. This algorithm is outlined in Appendix A.

3.3. Generating Explanations

We cluster each feature dimension in the input space and, as discussed in Section 3.1, use this as a foundation for generating explanations for any input instance. The computational cost of MASALA scales linearly with the number of feature dimensions. A specified target instance $\mathbf{x}_i$ will fall into a single linear region in each feature dimension. The set of instances $\mathcal{D}_i$ for training the surrogate g_i is defined using Equation 1. The linear coefficients of the MLR indicate the contribution of each feature towards the base model's prediction for the target instance.

4. Experiments

To evaluate MASALA we use the following two combinations of datasets and base models.

PHM08 Challenge [22] is a dataset used to predict the Remaining Useful Life (RUL) of a set of turbofan engines. A Gradient Boosting Regressor (GBR) [12] was trained using the lifetime operations of 218 engines, containing almost 46,000 samples. GBRs provide global feature importance scores, however these describe general model behaviour and are not sufficient for providing insights into the behaviour of the model at an instance level. The GBR achieved a RMSE of 59.5 on the test set. Appendix C.2 compares the predictions made by the base model for the PHM08 dataset to the true values. It can be noticed that the relationship between features and predictions is not always linear, and therefore a single linear surrogate model may not be able to capture the true model behaviour.

MIDAS [23] is a dataset provided by the UK Meteorological Office which contains hourly weather observations across the UK. We use data collected at Heathrow Airport over 3 years (Jan. 2019 - Dec. 2022), which contains 19138 observations of a number of weather related parameters. A Recurrent Neural Network (RNN) is trained to predict the air temperature at a given time. RNNs are complex deep learning models that lack inherent-interpretability. Many state-of-the-art temporal models incorporate RNN architectures, therefore, being able to explain the behaviour of such models is useful to ensure they can be trusted and applied safely. The RNN achieved a RMSE of 3.04 on the test set. Appendix C.1 compares the predictions made by the base model for the MIDAS dataset to the true values. Similar to the GBR, there is not a direct linear relationship between the distributions of input features and predictions. For both datasets, 75% of the data is used for training and 25% is reserved for testing.

Local fidelity is a common evaluation metric that measures how well a surrogate model approximates the behaviour of the base model, by comparing their respective predictions on the local data used to fit the surrogate [24, 25]. This local data, as is the case for LIME and CHILLI, may be perturbed samples of the instance being explained [14, 26], which may not be appropriate if the perturbations are not representative of the original training data [27]. Furthermore, locality is ill-defined in these existing methods and so local fidelity cannot be trusted. Instead, we measure *explanation fidelity* which is calculated as the absolute error between the predictions from the base model and surrogate model for the target instance.

$$\text{Average Explanation Fidelity} = \frac{1}{N}\sum_{i=1}^{N} |f(\mathbf{x}_i) - g_i(\mathbf{x}_i)| \tag{5}$$

The average explanation fidelity can be calculated over a number of N instances, as shown in Equation 5, to quantify the explanation methods performance. We calculate the average explanation fidelity for 20 instances selectedly uniformly at random from the test set.

Prior works have highlighted that when repeatedly explaining the same instance, random perturbation based methods, such as LIME, produce inconsistent and differing explanations [18, 28, 29]. To measure consistency of repeated explanations, existing works use Jaccard distance [18, 30]. Jaccard distance only considers explanations to be similar if their feature importance scores are identical. We instead propose calculating the average standard deviation of the normalised feature importance scores across 10 repeated explanations, which we subtract from 1 to measure *consistency*. We compare explanations generated by MASALA to those generated by LIME and CHILLI for a range of kernel width settings. Both the average consistency and fidelity of the explanations are calculated for all methods across 5 independent runs.

For illustration, the clustering used to generate explanations with MASALA for the MIDAS dataset is shown in Appendix B.

5. Results

Table 1 shows the average consistency and fidelity of all explanations obtained from the experiments along with the standard deviation across the 5 runs. It can be noticed that on average, higher kernel width settings for LIME and CHILLI (shown in parentheses in Table 1) produce explanations with lower error for target instances from the PHM08 dataset. However, as the explanation fidelity improves, the consistency decreases which makes it difficult to trust the more performant explanations. Figure 3 shows the absolute error of the explanations generated using CHILLI for the individual random instances at each kernel width setting. It can be noticed that the kernel width setting that produces the lowest error explanation varies. This highlights that the optimal locality parameter for each instance may vary and cannot be universally defined. For the PHM08 instances, explanations generated using MASALA always exhibited the lowest average error, and therefore greatest fidelity, compared to those generated using LIME and CHILLI across all kernel width values. This demonstrates the capability of MASALA to produce explanations that exceed the performance of LIME and CHILLI without the requirement of defining a kernel width value. Since the locality size of explanations generated using MASALA

	Consistency		Fidelity	
	PHM08	MIDAS	PHM08	MIDAS
LIME (0.1)	0.84 ± 0.134	0.54 ± 0.037	70.84 ± 15.199	0.0 ± 0.0
LIME (0.25)	0.63 ± 0.017	0.56 ± 0.029	70.84 ± 15.199	0.01 ± 0.004
LIME (0.5)	0.63 ± 0.02	0.84 ± 0.06	70.84 ± 15.199	1.16 ± 0.146
LIME (1.0)	0.73 ± 0.029	0.94 ± 0.055	66.95 ± 14.408	4.66 ± 0.39
CHILLI (0.01)	0.97 ± 0.012	0.90 ± 0.070	67.21 ± 12.66	0.0 ± 0.0
CHILLI (0.1)	0.96 ± 0.014	0.97 ± 0.011	66.28 ± 12.224	0.75 ± 0.092
CHILLI (0.25)	0.95 ± 0.036	0.97 ± 0.009	65.36 ± 12.481	0.97 ± 0.15
CHILLI (0.5)	0.92 ± 0.013	0.97 ± 0.008	62.60 ± 11.163	1.03 ± 0.177
CHILLI (1.0)	0.89 ± 0.01	0.97 ± 0.009	60.53 ± 9.573	1.09 ± 0.184
MASALA	**1.00 ± 0.000**	1.00 ± 0.000	35.94 ± 5.670	0.50 ± 0.163

Table 1
Average explanation consistency and fidelity achieved by LIME, CHILLI and MASALA with standard deviation across 5 independent runs.

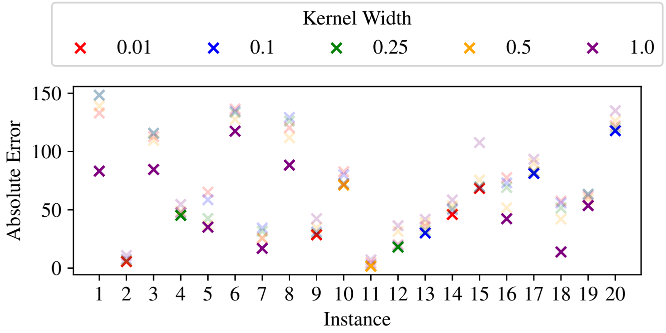


Figure 3: Error of explanations generated using CHILLI with different kernel widths for random instances from PHM08.

is dependent on local trends, it is able to capture the appropriate locality for individual instances without the need for a kernel width setting.

For the MIDAS dataset, CHILLI and LIME at low kernel width settings achieved significantly lower average error compared to MASALA. This can be attributed to the fact that at such small localities, the perturbations used to train the surrogate model only occupy a minuscule region of the input space, which the surrogate can model very accurately. It can also be noticed however, that at particularly small localities, LIME and CHILLI experience a lower consistency than MASALA. An example of repeated explanations for a single instance generated by each method is shown in Figure 4. There is significant variation in explanations generated at low kernel width values which may be attributed to LIME and CHILLI randomly sampling perturbations. We see that for a kernel width of 0.1, CHILLI produces much more consistent explanations, since the perturbation generation method considers the distribution of the original data and this locality size may be appropriate in capturing the relevant trends. However, this comes at the cost of fidelity, indicated by the lower average error, since the produced explanations may be describing more general trends rather than local ones. Consistency cannot be sacrificed for fidelity, since

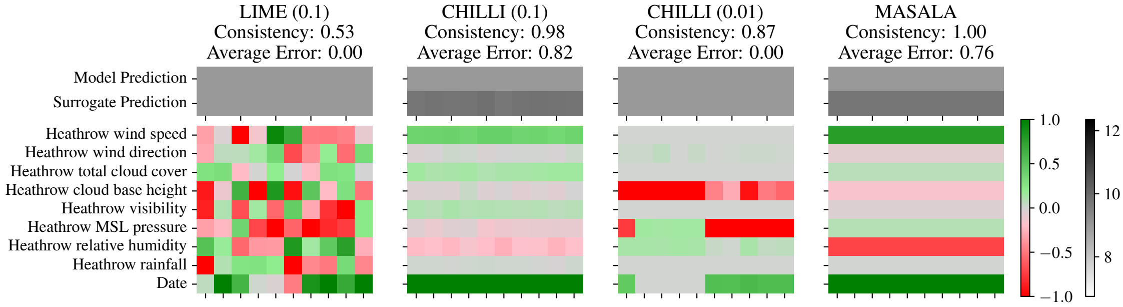


Figure 4: Explanations generated using LIME, CHILLI and MASALA, 10 times for the same instance from the MIDAS dataset. Each column is a single explanation with the colour of the square indicating the linear relationship each feature has towards the target variable. The kernel width setting used for LIME and CHILLI is shown in parentheses.

if multiple explanations are presented for a single prediction, there is uncertainty regarding their correctness and trustworthiness. Explanations generated using MASALA achieve perfect consistency since they are generated using a predetermined clustering, leading to identical repeated explanations.

6. Conclusion

In this paper we propose MASALA, a novel method for generating explanations for black-box model predictions using linear local surrogate models. We proposed a clustering technique that identifies a set of points which are similar to an instance for which an explanation is being generated, and use this to fit a linear surrogate model to approximate the base model behaviour. As a result, MASALA automatically detects the relevant and impactful model behaviour in an appropriately sized region of the input space. We find that explanations generated using our method produce more faithful and consistent explanations than those generated using LIME and CHILLI, without the need to manually define a locality hyperparameter that may differ for each instance being explained. Although a deterministic clustering ensures consistency, it is not clear how a non-deterministic clustering method which generates explaantions of equal fidelity would compare. There may be explanations that are equally faithful yet present different feature contributions so the question arises as to which of the explanations, or both, are correct. Future work would explore the possibilities of this and investigate whether such explanations are equally valid and how they can be compared.

Acknowledgments

We gratefully acknowledge the funding provided by the UK Engineering and Physical Sciences Research Council (grant ref. EP/T517641/1) and TRL Ltd to support this iCASE with project code B.CSAA.0001.

References

[1] K. Devitt, M. Gan, J. Scholz, R. Bolia, A method for ethical AI in Defence (2021). URL: https://apo.org.au/node/311150, publisher: Department of Defence (Australia).

[2] R. P. Singh, G. L. Hom, M. D. Abramoff, J. P. Campbell, M. F. Chiang, on behalf of the AAO Task Force on Artificial Intelligence, Current Challenges and Barriers to Real-World Artificial Intelligence Adoption for the Healthcare System, Provider, and the Patient, Translational Vision Science & Technology 9 (2020) 45. URL: https://doi.org/10.1167/tvst.9.2.45. doi:10.1167/tvst.9.2.45.

[3] A. Preece, D. Harborne, D. Braines, R. Tomsett, S. Chakraborty, Stakeholders in Explainable AI, 2018. URL: http://arxiv.org/abs/1810.00184, arXiv:1810.00184 [cs].

[4] F. K. Dosilovic, M. Brcic, N. Hlupic, Explainable artificial intelligence: A survey, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, Opatija, 2018, pp. 0210–0215. URL: https://ieeexplore.ieee.org/document/8400040/. doi:10.23919/MIPRO.2018.8400040.

[5] J. Tang, L. Xia, C. Huang, Explainable Spatio-Temporal Graph Neural Networks, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, ACM, Birmingham United Kingdom, 2023, pp. 2432–2441. URL: https://dl.acm.org/doi/10.1145/3583780.3614871. doi:10.1145/3583780.3614871.

[6] M. Goerigk, M. Hartisch, A framework for inherently interpretable optimization models, European Journal of Operational Research 310 (2023) 1312–1324. URL: https://www.sciencedirect.com/science/article/pii/S0377221723002953. doi:10.1016/j.ejor.2023.04.013.

[7] O. Loyola-Gonzalez, Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view, IEEE Access 7 (2019) 154096–154113. doi:10.1109/ACCESS.2019.2949286.

[8] J. Wanner, L.-V. Herm, K. Heinrich, C. Janiesch, A social evaluation of the perceived goodness of explainability in machine learning, Journal of Business Analytics 5 (2022) 29–50. doi:10.1080/2573234X.2021.1952913.

[9] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed., https://christophm.github.io/interpretable-ml-book, 2022. URL: https://christophm.github.io/interpretable-ml-book.

[10] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, 2016. URL: http://arxiv.org/abs/1602.04938, arXiv:1602.04938 [cs, stat].

[11] S. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, 2017. URL: http://arxiv.org/abs/1705.07874, arXiv:1705.07874 [cs, stat].

[12] J. Friedman, R. Tibshirani, Hastie, Trevor, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics, 2nd ed., Springer International Publishing, 2017.

[13] J. Dieber, S. Kirrane, Why model why? Assessing the strengths and limitations of LIME, 2020. URL: http://arxiv.org/abs/2012.00093, arXiv:2012.00093 [cs].

[14] S. Anwar, N. Griffiths, A. Bhalerao, T. Popham, M. Bell, S. Hellman, CHILLI: A data context-aware perturbation method for XAI, ICML 2023 Workshop on AI & Human Computer

Interaction (2023).

[15] X. Zhao, W. Huang, X. Huang, V. Robu, D. Flynn, BayLIME: Bayesian local interpretable model-agnostic explanations, in: Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, PMLR, 2021, pp. 887–896. URL: https://proceedings.mlr.press/v161/zhao21a.html, iSSN: 2640-3498.

[16] Z. Tan, Y. Tian, J. Li, GLIME: General, Stable and Local LIME Explanation, 2023. URL: http://arxiv.org/abs/2311.15722, arXiv:2311.15722 [cs, stat].

[17] T. Laugel, X. Renard, M.-J. Lesot, C. Marsala, M. Detyniecki, Defining Locality for Surrogates in Post-hoc Interpretablity, 2018. URL: http://arxiv.org/abs/1806.07498, arXiv:1806.07498 [cs, stat].

[18] M. R. Zafar, N. M. Khan, DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems, 2019. URL: http://arxiv.org/abs/1906.10263, arXiv:1906.10263 [cs, stat].

[19] R. Gaudel, L. Galárraga, J. Delaunay, L. Rozé, V. Bhargava, s-LIME: Reconciling Locality and Fidelity in Linear Explanations, 2022. URL: http://arxiv.org/abs/2208.01510, arXiv:2208.01510 [cs].

[20] S. M. Shankaranarayana, D. Runje, ALIME: Autoencoder Based Approach for Local Interpretability, in: H. Yin, D. Camacho, P. Tino, A. J. Tallón-Ballesteros, R. Menezes, R. Allmendinger (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2019, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2019, pp. 454–463. doi:10.1007/978-3-030-33607-3_49.

[21] H.-S. Park, C.-H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Systems with Applications 36 (2009) 3336–3341. URL: https://linkinghub.elsevier.com/retrieve/pii/S095741740800081X. doi:10.1016/j.eswa.2008.01.039.

[22] A. Saxena, D. Simon, N. Eklund, Damage Propagation Modeling for Aircraft Engine Prognostics (2008).

[23] Met Office, Met Office MIDAS Open: UK Land Surface Stations Data (1853-current), Centre for Environmental Data Analysis, 2019.

[24] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A Survey of Methods for Explaining Black Box Models, ACM Computing Surveys 51 (2019) 1–42. URL: https://dl.acm.org/doi/10.1145/3236009. doi:10.1145/3236009.

[25] A. A. Freitas, Comprehensible classification models: a position paper, ACM SIGKDD Explorations Newsletter 15 (2014) 1–10. URL: https://dl.acm.org/doi/10.1145/2594473.2594475. doi:10.1145/2594473.2594475.

[26] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local Rule-Based Explanations of Black Box Decision Systems, 2018. URL: http://arxiv.org/abs/1805.10820, arXiv:1805.10820 [cs].

[27] C. Molnar, G. König, J. Herbinger, T. Freiesleben, S. Dandl, C. A. Scholbeck, G. Casalicchio, M. Grosse-Wentrup, B. Bischl, General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models, Technical Report arXiv:2007.04131, arXiv, 2021. URL: http://arxiv.org/abs/2007.04131, arXiv:2007.04131 [cs, stat] type: article.

[28] Y. Zhang, K. Song, Y. Sun, S. Tan, M. Udell, "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations, 2019. URL: http://arxiv.org/abs/1904.

12991, arXiv:1904.12991 [cs, stat].

[29] Z. C. Lipton, The Mythos of Model Interpretability, 2017. URL: http://arxiv.org/abs/1606.03490, arXiv:1606.03490 [cs, stat].

[30] E. Amparore, A. Perotti, P. Bajardi, To trust or not to trust an explanation: using LEAF to evaluate local linear XAI methods, PeerJ Computer Science 7 (2021) e479. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8056245/. doi:10.7717/peerj-cs.479.

A. Local Linear K-Medoids Clustering Algorithm

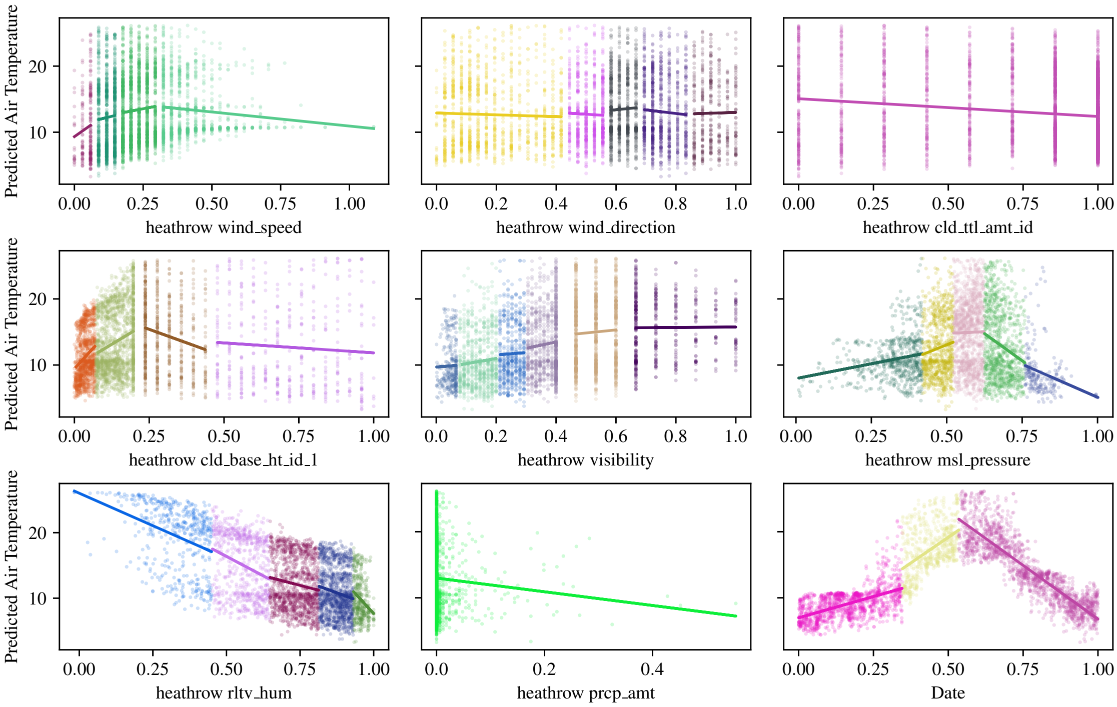
Algorithm 1 Local Linear K-Medoids Clustering

```
Require: $K > 0$, $\Delta_{\mathcal{X}}$, Previous $K = 0$, Clustering Cost = $\infty$
  while Previous $K \neq K$ do
    Previous $K = K$
    Select $K$ medoids which are evenly distributed across the distribution of $\mathcal{X}$
    for each non-medoid $x \in \mathcal{X}$ do
      Find the least dissimilar medoid according to $\Delta_{\mathcal{X}}$ and assign $x$ to
      the corresponding cluster to generate clustering $C$
    end for
    Fit LR model within each cluster $C_k$
    repeat
      Calculate Clustering Cost $J(C)$, using Equation 3
      Lowest Cost = $J(C)$
      for  each cluster $C_k \in C$  do
        for each $x \in C_k$ do
          Change medoid for $C_k$ to $x$
          Generate new clustering $C$ with new medoids
          Fit LR models within each cluster $C_k$
          Calculate $J'(C)$ using Equation 3
          if $J'(C) <$ Lowest Cost then
            Lowest Cost = $J(C)$
            Accept $x$ as new medoid
          else
            Reject $x$ as new medoid
          end if
        end for
      end for
      Cost Difference = Lowest Cost - $J(C)$
      if Cost Difference < 0 then
        Accept new clustering
        Clustering Cost = $J(C)$
      end if
    until Cost Difference = 0
    $C'$ = Clusters after satisfying all constraints for clustering $C$
    $K$ = Number of clusters in $C'$
  end while
```

B. MIDAS Clustering

Here we show the clustering obtained by applying Algorithm 1 to the MIDAS data containing features describing weather characteristics at Heathrow Airport between 2019 and 2022. The distribution of each feature is shown against the predictions obtained by the base model being explained, in this case a Recurrent Neural Network. The clustered regions of linear behaviour are shown as different colours with the linear regression model fit to the points within each cluster also shown as a line of the same colour.

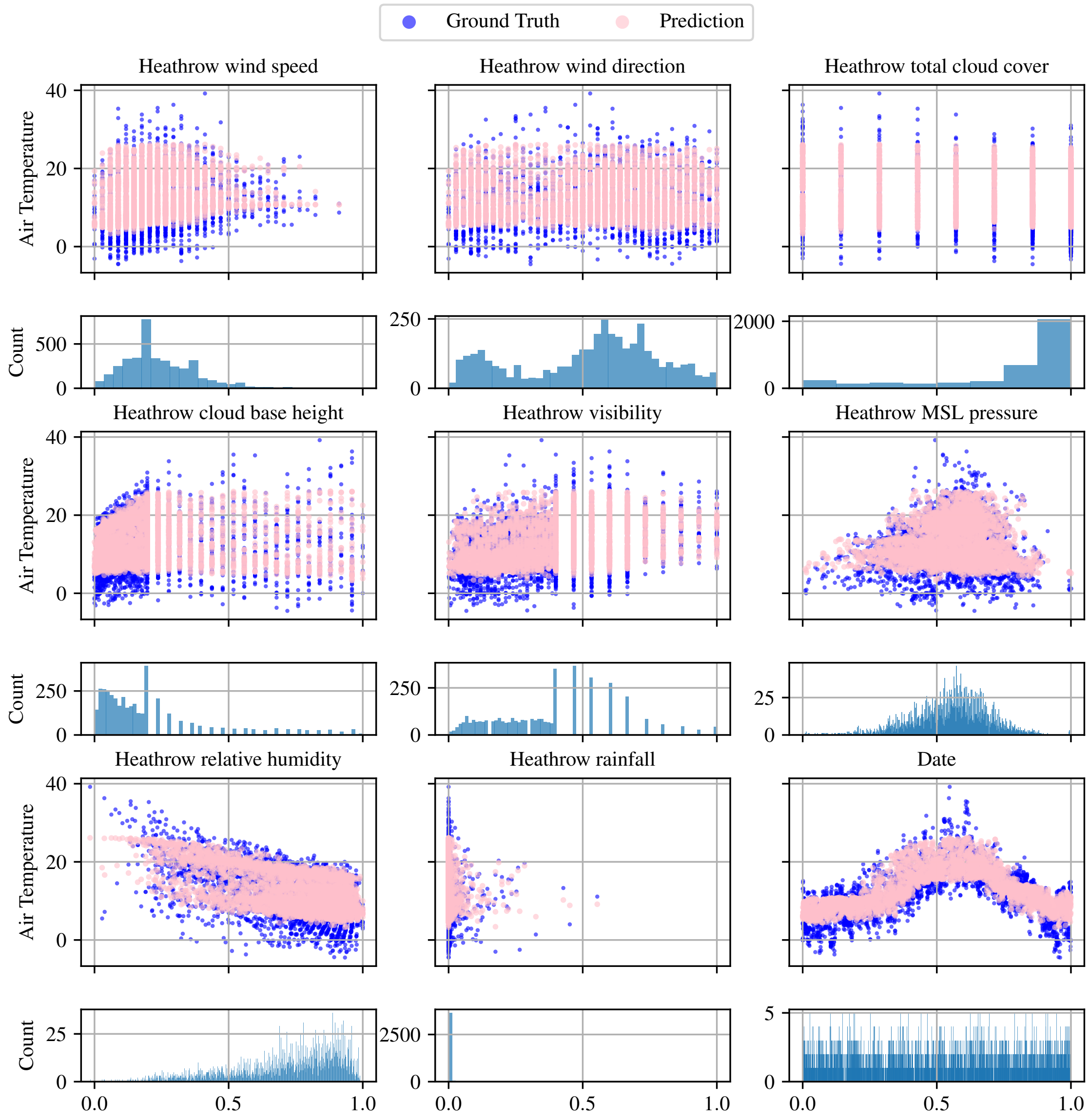
We have not shown the clustering obtained for the PHM08 dataset because there are too many features to be able to visualise the individual clusters effectively.



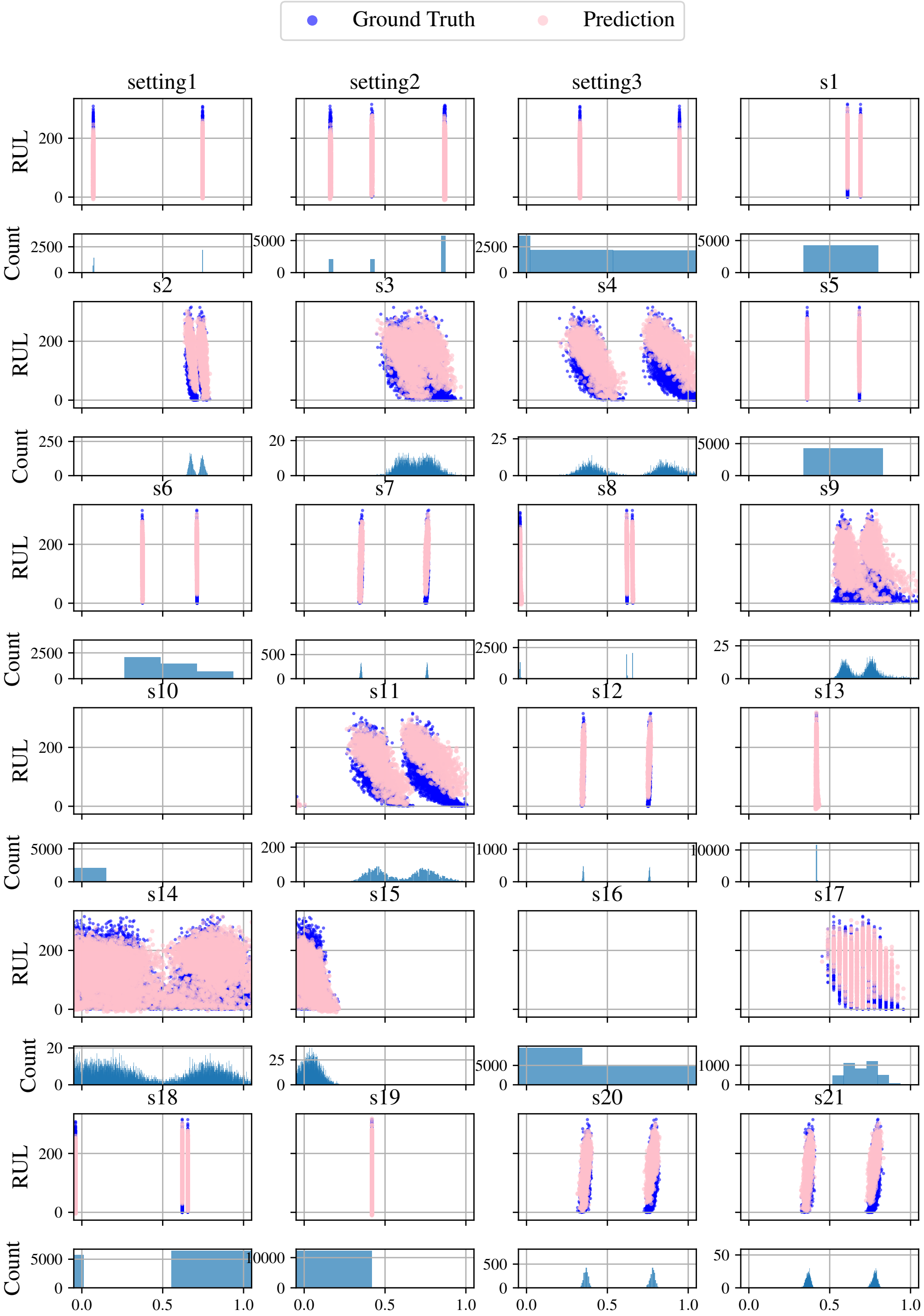
C. Model Predictions

The distribution of each feature from the MIDAS and PHM08 datasets against the true target values and the RNN and GBR models respectively. It can be seen that there is not a clear linear relationship present between many features and the target variable. Therefore, a single linear model would not be appropriate for explaining the behaviour of the base model for all instances.

C.1. MIDAS Recurrent Neural Network



C.2. PHM08 Gradient Boosting Regressor


Ground Truth
Prediction
setting1
setting2
setting3
s1
s2
s3
s4
s5
s6
s7
s8
s9
s10
s11
s12
s13
s14
s15
s16
s17
s18
s19
s20
s21
RUL
Count