BayLIME: Bayesian Local Interpretable Model-Agnostic Explanations

Xingyu Zhao[1,2], **Xiaowei Huang**[2], **Valentin Robu**[1], **David Flynn**[1]
[1]School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, U.K.
[2]Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, U.K.
Emails:{xingyu.zhao,xiaowei.huang}@liverpool.ac.uk,{v.robu,d.flynn}@hw.ac.uk

arXiv:2012.03058v1 [cs.AI] 5 Dec 2020

Abstract

A key impediment to the use of AI is the lacking of transparency, especially in safety/security critical applications. The black-box nature of AI systems prevents humans from direct explanations on how the AI makes predictions, which stimulated Explainable AI (XAI) – a research field that aims at improving the trust and transparency of AI systems. In this paper, we introduce a novel XAI technique, BayLIME, which is a Bayesian modification of the widely used XAI approach LIME (Ribeiro, Singh, and Guestrin 2016). BayLIME exploits prior knowledge to improve the consistency in repeated explanations of a single prediction and also the robustness to kernel settings. Both theoretical analysis and extensive experiments are conducted to support our conclusions.

1 Introduction

Recent years have seen increasing interest in building AI-enabled systems that not only accurate, but also trustworthy and capable of explaining their actions. Explainable AI (XAI) methods can be classified by various criteria (Molnar 2020, Chpt. 2.2), e.g., model-specific vs model-agnostic, local (instance-wise) vs global (entire model) or intrinsic vs post-hoc. While readers may refer to (Huang et al. 2020; Adadi and Berrada 2018) for a review, we confine ourselves in this paper on using local surrogate models for explaining individual predictions of black-box AI/machine-learning (ML) models, and present a novel Bayesian modification on the most popular method in this category – Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016). Despite the success of LIME, it has several disadvantages among which, in this work, we mainly aim to address the lack of *consistency in repeated explanations of a single prediction* and *robustness to kernel settings*. We argue these are among the most desirable properties in building trust in AI.

The inconsistency of LIME, where different explanations can be generated for the same prediction, has been identified as a critical issue by, e.g. (Rehman Zafar and Mefraz Khan 2019; Shankaranarayana and Runje 2019). It is caused by the randomness in the perturbed samples (of the instance under explanation) used for the training of local surrogate models, and naturally a smaller sample size leads to greater noise of such randomness. The inconsistency creates distrust, e.g., in medical domain where consistency is highly required (Rehman Zafar and Mefraz Khan 2019). Even worse, for applications with strict efficiency constraints (e.g., robotics missions), the perturbed sample size has to be restricted resulting in a more severe inconsistency issue.

As a quick example, in Fig. 1, the pretrained Convolutional Neural Network (CNN) InceptionV3 (Szegedy et al. 2016) predicts the instance of Fig. 1(A) as "Bernese mountain dog" (top label) with probability 0.6826. To interpret this prediction, we vary the size of perturbed samples (denoted as n) generated for the training of a local surrogate model, and record the time consumption in Fig. 1(B), from which we see the computational time has a *linear relation* to n. If, say, an application requires LIME to respond in 20s so that we have to limit n to around 100 (in our case), then we easily get three *inconsistent* explanations, as shown in Fig. 1(C)-(E), in three repeated runs of LIME.

The other problem of LIME motivating this work relates to defining the right *locality* – the "neighbourhood" on which a local surrogate is trained to approximate the ML model (Laugel et al. 2018). As noted by (Molnar 2020, Chpt. 5.7) as a "very big and unsolved problem", how to choose a kernel setting in LIME to properly define a neighbourhood is challenging. Even worse, explanations can be easily turned around by changing kernel settings (cf. the example in (Molnar 2020, Chpt. 5.7)). Thus, we prefer a more robust XAI technique to the kernel settings.

In this paper, we propose a novel Bayesian modification of LIME (BayLIME) to improve both the consistency and robustness properties mentioned above by incorporating prior knowledge. The leveraging of useful priors also allows the generation of *less* perturbed samples, thus BayLIME may benefit even more for applications with strict efficiency constraints. Our main contributions include: (1) A new XAI technique, BayLIME, that provides a Bayesian principled mechanism for the embedding of useful prior knowledge; (2) Two quantitative measures on the consistency in repeated explanations and robustness to kernel settings for feature-ranking based XAI techniques; (3) An open source prototype tool of BayLIME.

Next, we present preliminaries on LIME. Sec. 3 and 4 formally define the consistency and robustness properties studied in this paper. Sec. 5 introduces the BayLIME framework

Authors' preprint, new versions to appear.

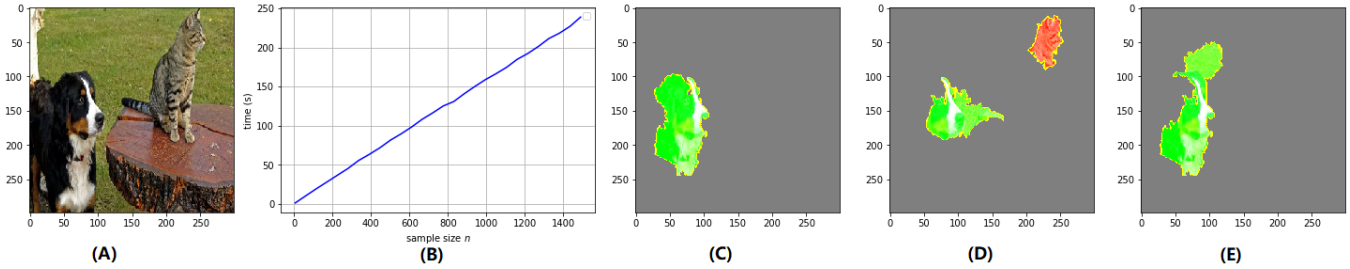

Figure 1: (A) The instance to be explained; (B) LIME time-consumption as a function of perturbed sample size n; (C)-(E) Three repeated LIME interpretations (showing the top-4 features) with the perturbed local samples size $n = 100$.

in detail with experimental results presented in Sec. 6, while relate work is summarised in Sec. 7. Finally, Sec. 8 concludes BayLIME and discusses future work.

2 Preliminaries on LIME

LIME (Ribeiro, Singh, and Guestrin 2016) implements a local surrogate model that is used to explain individual predictions of black-box ML models. The intuition behind LIME is as follows. For a given black-box ML model (**model-agnostic**), we may probe it as many times as possible by perturbing some features (e.g., hiding superpixels of an image) of the input instance of interest (**locally**) and see how the prediction changes. Such changes in the predictions, as we vary the features of the instance, can help us understand why the ML model made a certain prediction over the instance. Then a new dataset consisting of the perturbed inputs and the corresponding predictions made by the black-box ML model can be generated, upon which LIME trains a surrogate model that is **interpretable** to humans (e.g., linear regressors, cf. (Molnar 2020, Chpt. 4) for more). The training of the interpretable surrogate model is weighted by the proximity of the perturbed samples to the original instance.

To be exact, the basic steps of how LIME works are:

1) Choose an instance $\boldsymbol{x}$ to interpret (e.g., an image or a row from a tabular dataset) in which there are m features[1] (e.g., the number of superpixels for images or columns for tabular data). Denote $\boldsymbol{x}$ as a $m \times 1$ column vector.

2) Do perturbation on $\boldsymbol{x}$ (e.g., switch on/off image superpixels) and generate a new input set $\boldsymbol{X} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_n}\}$ with n samples, i.e. $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{n \times m}$.

3) Probe the black-box ML model with $\boldsymbol{X}$ and record the predictions as a column vector $Y = [y_1, ..., y_n]^T$.

4) Weight the n perturbed samples in $\boldsymbol{X}$ according to their proximity to the original instance $\boldsymbol{x}$. Say the weights calculated by the kernel function (by default, an exponential kernel defined on some kernel width) are $\{w_1, ..., w_n\}$, then the new weighted dataset becomes $(\boldsymbol{X}', Y') = (\boldsymbol{WX}, \boldsymbol{W}Y)$ where $\boldsymbol{W} = diag(w_1, ..., w_n)$ is a diagonal matrix with diagonal elements equal to the w_i's.

5) On the dataset $(\boldsymbol{X}', Y')$, train a linear regressor

$$Y' = \boldsymbol{X}'\boldsymbol{\beta} + \epsilon \tag{1}$$

where $\boldsymbol{\beta}$ and ϵ are the coefficients and Gaussian white noise.

6) Then, the *absolute values* of the coefficient vector $\boldsymbol{\beta}$ represent the importance of the m features, based on which rankings can be done. By default, LIME uses the Ridge regressor with weighted samples (Holland 1973):

$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} + r\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{W}Y \tag{2}$$

where r is the Ridge regularisation parameter.

3 Consistency in Repeated Explanations

LIME is known to be unstable in terms of generating inconsistent explanations in repeated runs on a same prediction (Rehman Zafar and Mefraz Khan 2019; Shankaranarayana and Runje 2019). In general, it is caused by the randomly perturbed samples used for the training of local surrogate models. The presence of randomness means that the results of sampling might not be duplicated if the process were repeated, and naturally the more sparse samples generated the greater randomness presents in the dataset.

Consequently, such inconsistency is an even severer issue for applications with strict efficiency constraints (e.g., low response time and energy consumption) due to the fact that only a limited number of perturbed samples is allowed to be generated. Indeed, an effective way to improve LIME's efficiency is to reduce the number of perturbed samples n, e.g., as shown in Fig. 1(B). This is a corollary of what has been noted by the original LIME paper (Ribeiro, Singh, and Guestrin 2016): The time required to produce an explanation is *dominated* by the complexity of the black-box ML model. Thus, we cannot effectively tune other LIME arguments to get faster, rather the best option is to limit the *number of queries made* to the complex ML model (which, indeed, could be very costly when the ML model is deep (He and Sun 2015)) when generating labels Y for the n perturbed samples (step 3 of the LIME workflow).

While a quick example is shown by Fig. 1(C)-(E), we introduce a measure of such inconsistency in the following steps : *a)* In a single explanation, we first normalise the coefficient vector of m features (represented by $\boldsymbol{\beta}$, cf. step 6 in the LIME workflow) into an unit vector $\hat{\boldsymbol{\beta}} = \frac{\boldsymbol{\beta}}{||\boldsymbol{\beta}||}$. *b)* For repeated k explanations (denoted by the upper index $^{(k)}$ in the following notations), the normalised importance of a feature i (i.e. $|\hat{\beta}_i|$) and its ranking may form two probability distributions, denoted respectively as $g_i^{(k)} : [0, 1] \rightarrow [0, 1]$

[1] LIME allows the selection of top few features by K-LASSO, while we simply use all features in this paper for brevity.

and $f_i^{(k)} : [1, m] \rightarrow [0, 1]$. *c)* The *index of dispersion* (IoD) of $f_i^{(k)}$ essentially measures the relative variability of the ranks of the feature i in the k repeated explanations, denoted as $IoD(f_i^{(k)})$. To be consistent, we would expect a small $IoD(f_i^{(k)})$. *d)* The average importance of each feature $\mathbb{E}(g_i^{(k)})$ is quite different. Intuitively, we need to weight each $IoD(f_i^{(k)})$ accordingly. *e)* Finally, we define:

Definition 1 *The inconsistency of repeated k explanations for an instance with m features is measured by:*

$$\sum_{i=1}^{m} \frac{\mathbb{E}(g_i^{(k)})}{\sum_{j=1}^{m} \mathbb{E}(g_j^{(k)})} IoD(f_i^{(k)}) \tag{3}$$

where $g_i^{(k)} : [0, +\infty) \rightarrow [0, 1]$ and $f_i^{(k)} : [1, m] \rightarrow [0, 1]$ are probability distributions of the normalised importance of feature i and its ranks in k explanations respectively.

Intuitively, the inconsistency is a weighted sum of IoD of each feature's ranks in repeated explanations, so that the IoD of a more important feature is weighted higher. Upon establishing the inconsistency measure (3), our experimental results in Sec. 6 highlight the inconsistency issue of LIME and how our BayLIME improves on it.

4 Robustness to Kernel Settings

Another notorious problem of LIME (or any approach with a notion of localisation) is how to meaningfully define a "neighbourhood" of the instance of interest (as required by the step 4 of the LIME workflow depicted in Sec. 2), e.g., as noted by (Molnar 2020, Chpt. 5.7) as a "very big and unsolved problem". To be exact, by default, LIME uses an exponential smoothing kernel to define the neighbourhood with a given kernel width which determines how large the neighbourhood is: A small kernel width means that an instance must be very close to influence the local model, vice versa. However, there is no good way to find the best kernel settings. The best strategy for now is to try different kernel settings and see if the explanations make sense, which inevitably is subject to errors/bias. Moreover, in many scenarios, you can easily turn your explanation around by changing the kernel settings, as shown by an example in (Molnar 2020, Chpt. 5.7). This lack of robustness in XAI techniques to the bias in choosing kernel settings motivates our work.

Specifically, we explore the robustness of AI explainers to the kernel width parameter (denoted by l later) for a given kernel function. Without loss of generality, other kernel setting parameters[2] can be studied in a similar way.

Definition 2 *To explain a given instance i, we denote $h_i : (0, +\infty) \rightarrow \mathbb{R}^m$ as the importance vector of the m features taking l as the kernel width setting. For any pair of kernel width parameters l_1 and l_2, there exists a global Lipschitz value $L \in \mathbb{R}$ such that $||h_i(l_1) - h_i(l_2)|| \leq L||l_1 - l_2||$.*

Naturally, this global Lipschitz value quantifies the robustness of an explainer to the choices of kernel width.

[2]E.g., the distance parameter in a periodic kernel.

The calculation of L is an optimisation problem:

$$L = \underset{l_1, l_2 \in (0, +\infty)}{\arg\max} \frac{||h_i(l_1) - h_i(l_2)||}{||l_1 - l_2||} \tag{4}$$

which is very challenging to solve analytically or to estimate numerically by state-of-the-art approaches for black-box optimisation, due to the high complexity and non-linearity of the function h_i. Similar difficulties are also faced by (Alvarez-Melis and Jaakkola 2018) when studying the robustness to similar input instances.

To bypass those difficulties and still provide insights on the robustness of XAI techniques, we instead introduce a weaker empirical notion of the robustness to kernel settings:

Definition 3 *Assume L_1 and L_2 are both random variables of kernel width settings within $[l_{lo}, l_{up}]$, then R is the median[3] (denoted as $\mathbb{M}(\cdot)$) of the ratio between the perturbed distances of h_is and the pair (L_1, L_2):*

$$R = \mathbb{M}\left(\frac{||h_i(L_1) - h_i(L_2)||}{||L_1 - L_2||} \middle| l_{lo} \leq L_1 \leq l_{up}, l_{lo} \leq L_2 \leq l_{up} \right) \tag{5}$$

which represents the average robustness to the kernel settings when explaining the instance i.

The kernel width cannot be too large nor too small (cf. the general discussions on variance-bias trade-off in selecting hyper-parameters (Cawley and Talbot 2010)). In practice, after a few tries, e.g., by cross-validation, it is easy to obtain empirically a bound of $[l_{lo}, l_{up}]$ as the range of all possible kernel settings in which we perturb the parameter to check the robustness. In (5) we focus on the median value, rather than the optimised solution in (4), which is a much easier quantity to estimate. For non-safety-critical applications, we believe (5) provides better insights on the general robustness. On the other hand, for safety-critical applications where we concern more on the worst case, the global Lipschitz in (4) should be used and carefully estimated (e.g., by trustworthy black-box optimisation routines).

5 A Bayesian Retrofit of LIME

5.1 Bayesian Linear Regression

In Bayesian linear regression, a response y_i is assumed to be Gaussian distributed around $\boldsymbol{\beta}^T \boldsymbol{x_i}$:

$$Pr(y_i \mid \boldsymbol{\beta}, \boldsymbol{x_i}, \alpha) = \mathcal{N}(y_i \mid \boldsymbol{\beta}^T \boldsymbol{x_i}, \alpha^{-1}) \tag{6}$$

where α is the precision parameter (reciprocal of the variance) representing noise in the data. Then we may write down the likelihood function:

$$Pr(Y \mid \boldsymbol{\beta}, \boldsymbol{X}, \alpha) = \prod_{i=1}^{n} \mathcal{N}(y_i \mid \boldsymbol{\beta}^T \boldsymbol{x_i}, \alpha^{-1}). \tag{7}$$

For computational convenience, we choose a conjugate prior distribution for $\boldsymbol{\beta}$ – a (multivariate) Gaussian again:

$$Pr(\boldsymbol{\beta} \mid \boldsymbol{\mu_0}, \boldsymbol{S_0}) = \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu_0}, \boldsymbol{S_0}). \tag{8}$$

[3]Although other statistics may also suffice, we choose the median value to cope with the possible extreme outliers.

where $\boldsymbol{\mu_0}$ and $\boldsymbol{S_0}$ are the mean vector and covariance matrix respectively. Then, by the Bayes Theorem, we know the posterior $\boldsymbol{\beta}$ is also a Gaussian (thanks to the conjugacy):

$$\begin{aligned} Pr(\boldsymbol{\beta} \mid Y, \boldsymbol{X}, \alpha, \boldsymbol{\mu_0}, \boldsymbol{S_0}) &\propto Pr(Y \mid \boldsymbol{\beta}, \boldsymbol{X}, \alpha) Pr(\boldsymbol{\beta} \mid \boldsymbol{\mu_0}, \boldsymbol{S_0}) \\ &= \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu_n}, \boldsymbol{S_n}) \end{aligned} \tag{9}$$

where the posterior mean vector $\boldsymbol{\mu_n}$ and covariance matrix $\boldsymbol{S_n}$ are

$$\boldsymbol{\mu_n} = \boldsymbol{S_n}(\boldsymbol{S_0}^{-1}\boldsymbol{\mu_0} + \alpha \boldsymbol{X}^T Y) \tag{10}$$

$$\boldsymbol{S_n}^{-1} = \boldsymbol{S_0}^{-1} + \alpha \boldsymbol{X}^T \boldsymbol{X}. \tag{11}$$

For simplicity, if we consider the Gaussian prior is governed by a single precision parameter λ, i.e. $\boldsymbol{S_0} = \lambda^{-1}\boldsymbol{I}_m$ (where $\boldsymbol{I}_m$ is a $m \times m$ identity matrix), then:

$$\boldsymbol{S_n}^{-1} = \lambda \boldsymbol{I}_m + \alpha \boldsymbol{X}^T \boldsymbol{X}. \tag{12}$$

From Eq. (10) and (12), we know the prior knowledge is effectively encoded into the posteriors via the two precision parameters (α and λ) and the prior mean vector ($\boldsymbol{\mu_0}$).

5.2 Embedding Prior Knowledge in LIME

The interpretable local surrogate model is trained on weighted samples perturbed around the instance of interest (cf. the step 4 in the LIME workflow). If we use a Bayesian linear regressor as our surrogate model, then, by substituting $(\boldsymbol{X}, Y)$ in Eq. (10) and (12) with the $(\boldsymbol{X}', Y')$ (defined earlier as weighted samples), the posterior estimates on $\boldsymbol{\beta}$ now are with Gaussian parameters:

$$\boldsymbol{\mu_n} = \boldsymbol{S_n}(\lambda \boldsymbol{I}_m \boldsymbol{\mu_0} + \alpha \boldsymbol{X}^T \boldsymbol{W} Y) \tag{13}$$

$$\boldsymbol{S_n}^{-1} = \lambda \boldsymbol{I}_m + \alpha \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}. \tag{14}$$

The result (13) can be rewritten as:

$$\begin{aligned} \boldsymbol{\mu_n} = &(\lambda \boldsymbol{I}_m + \alpha \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \lambda \boldsymbol{I}_m \cdot \boldsymbol{\mu_0} \\ &+ (\lambda \boldsymbol{I}_m + \alpha \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \alpha \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X} \cdot \boldsymbol{\beta}_{MLE} \end{aligned} \tag{15}$$

where $\boldsymbol{\beta}_{MLE} = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} Y$ is the Maximum Likelihood Estimates (MLE) for the linear regression model (Bishop 2006) on the weighted samples $(\boldsymbol{X}', Y')$.

The Eq. (15) is essentially a *weighted* sum of $\boldsymbol{\mu_0}$ and $\boldsymbol{\beta}_{MLE}$: A *Bayesian combination* of prior knowledge and the new observations. The weights are proportional to

- $\lambda \boldsymbol{I}_m$: the "pseudo-count" of prior sample size based on which we form our prior estimates of $\boldsymbol{\mu_0}$.
- $\alpha \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}$: the "accurate-actual-count" of observation sample size, i.e. the actual observation of the n perturbed samples $\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X}$ scaled by the precision α.

To see the above insight clearer, we present the special case of a single feature instance ($m = 1$) with a simplified kernel function that returns a same constant weight w_c (i.e. $w_i = w_c, \forall i = 1, .., n$), then μ_n becomes:

$$\frac{\lambda}{\lambda + \alpha w_c \sum_{i=1}^n x_i^2} \mu_0 + \frac{\alpha w_c \sum_{i=1}^n x_i^2}{\lambda + \alpha w_c \sum_{i=1}^n x_i^2} \beta_{MLE} \tag{16}$$

where $\beta_{MLE} = \frac{\alpha w_c \sum_{i=1}^n x_i y_i}{\alpha w_c \sum_{i=1}^n x_i^2} = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}$.

Moreover, denote X as the r.v. represents x_i, we know:

$$\sum_{i=1}^n x_i^2 = n\mathbb{E}(X^2) = n\left(Var(X) + \mathbb{E}(X)^2\right). \tag{17}$$

As implemented by LIME, x_is are n random samples from some distribution, depending on the type of the feature, e.g.: (i) For superpixels of images, $x_i \in \{0, 1\}$ are random samples from an uniform 0-1 two-point distribution; (ii) For numerical features of a tabular dataset, x_i are random samples from a standard Gaussian $\mathcal{N}(0, 1)$. Thus, say, for numerical features from tabular data, $\sum_{i=1}^n x_i^2 \approx n(1 + 0^2) = n$ via Eq. (17). Then (16) can be further simplified as:

$$\frac{\lambda}{\lambda + \alpha w_c n} \mu_0 + \frac{\alpha w_c n}{\lambda + \alpha w_c n} \beta_{MLE} \tag{18}$$

which neatly shows us how the prior knowledge has been embedded in our BayLIME in an intuitive story – "based on λ data-points prior to our new experiment, we form our prior estimate of μ_0. Now in the experiments, we collect n samples. After considering the precision (α) and weights (w_c) of the newly collected n samples, we form a MLE estimate β_{MLE}. Then we combine the two estimates – μ_0 and β_{MLE} – according to their proportions of the *effective* samples size used, i.e., λ and $\alpha w_c n$ respectively. Finally, the confidence in our new posterior estimate is captured by all effective samples used, i.e. $\lambda + \alpha w_c n$ (the posterior precision)". Now it is easy to see via Eq. (18) (which also holds for Eq. (15)):

Remark 1 *Smaller λ means lower confidence in the prior knowledge, thus the posteriors are mostly dominated by the new observations, and vice versa. That is, in extreme cases:*

- *When $\lambda \simeq 0$ the result* (15) *(and its simplified case* (18)*) reduces to MLE, i.e. "let the data speak for themselves".*
- *On the other hand, if $n \simeq 0$ (or equivalently $\alpha \simeq 0$, $w_c \simeq 0$), then the β_{MLE} estimate from the new data is negligible and the prior knowledge dominates the posteriors.*

It is worth noting that this kind of intuitive decomposition is generic to any Bayesian inference in the linearly updated canonical-exponential families (Bernardo and Smith 1994), which is found to be useful in many applications that exploit prior knowledge, e.g., in (Zhao et al. 2019; Walter and Augustin 2009; Filieri, Ghezzi, and Tamburrelli 2012).

5.3 The BayLIME Framework

BayLIME essentially uses Bayesian linear regressors as the local surrogate models, and considers three scenarios in which we have respectively: (i) no prior knowledge; (ii) knowledge on the prior distribution of the coefficients ($\boldsymbol{\mu_0}$ and λ) but not on the noise in data (α); (iii) full knowledge on both the priors of the coefficients and the noise in data.

BayLIME with non-informative priors. There is no prior knowledge in this scenario, thus we assume a zero mean vector for $\boldsymbol{\mu_0}$ and do Bayesian model selection (Bishop 2006, Chpt. 3.4) for λ and α. Specifically, BayLIME reuses the BAYESIANRIDGE linear regressor class from the SCIKIT-LEARN library (Pedregosa et al. 2011). It implements the algorithm described in (Tipping 2001) where the

fittings of the α and λ parameters from data are done as suggested in (MacKay 1992). Although estimating prior parameters from data is traditionally called "empirical Bayes", we name it in BayLIME as "with non-informative prior" to form a sharp contrast to the following BayLIME options.

BayLIME with partial informative priors. We assume there is some prior knowledge – a mean vector $\boldsymbol{\mu_0}$ and its precision λ (i.e. a complete prior distribution of the feature coefficients). We call this option in BayLIME "with partial priors" in the sense that we still don't know the parameter α, for which we fit from data like in the previous case.

BayLIME with full informative priors. By assuming an ideal scenario in which we have full prior knowledge of all the $\boldsymbol{\mu_0}$, λ and α parameters, BayLIME may directly implement the results of (13) and (14) in this case.

Ways of obtaining prior knowledge. Although we explicitly implement the three options mentioned above in BayLIME exploiting the availability of prior knowledge, how to rigorously elicit accurate priors remains an open question and is out of the scope of this paper (and forms important future work). However, we believe such challenge is neither unique to BayLIME, but generic to any approach with a Bayesian flavour, nor intractable in practice, e.g.:

a) It is a trend to embed human knowledge in the training of AI/ML models (Ilyas et al. 2019; Lamb et al. 2020; Huang, Zhao, and Huang 2020), and human-in-the-loop has been widely adopted in many ML applications, e.g., (Branson, Perona, and Belongie 2011; Branson et al. 2010). Such explicitly embedded knowledge reflecting the true AI/ML model behaviours naturally forms the prior knowledge required by BayLIME.

b) There are emerging XAI techniques (cf. (Huang et al. 2020; Molnar 2020) for a review), a collection of explanations by several *diverse* AI explainers based on fundamentally different theories (e.g., gradient-based vs perturbation-based, global vs local) may provide useful prior knowledge that can be combined by BayLIME, presuming the drawbacks of individual XAI methods can be covered by such "hybrid" solution provided by BayLIME.

c) Validation & verification techniques that directly analyse the behaviour of an AI/ML model may also indicate the importance of certain features, forming prior knowledge.

d) In our experiments in the following section, we use explanations of *a set of similar instances* (to the instance under explanation) to form our prior knowledge. To be exact, the mean of the importance of each feature in that set together forms our prior mean vector $\boldsymbol{\mu_0}$, and the number of similar instances implies λ (leaving the precision parameter of the likelihood α either as unknown or assigned empirically). Although it is still not rigorous enough (e.g., how to decide what is a "similar" instance), we believe our way of obtaining priors serves as a first illustrative example.

6 Evaluation

6.1 Research Questions and Evaluation Methods

We evaluate our BayLIME by performing extensive experiments to address the following research questions.

RQ1: What is the inconsistency issue of LIME (especially when considering efficiency) that has been dealt with by BayLIME? We carried out experiments to reveal the issue by quantifying the inconsistency via both (3) and the well-established statistics Kendall's W (Kendall and Smith 1939).

RQ2: How effectively BayLIME improves the consistency when explaining an instance in different scenarios of available prior knowledge? By varying the parameters λ and α with different BayLIME options, experiments were conducted across a range of settings.

RQ3: How effectively BayLIME improves the robustness to kernel settings when explaining an instance in different scenarios of available prior knowledge? Given the lack of robustness to kernel settings is a well-known issue of LIME (Molnar 2020; Laugel et al. 2018), we examined the robustness to kernel settings of both LIME and BayLIME by approximating the measure defined in (5), and compute statistics for insights on how effectively BayLIME improves on it.

Methods LIME is one of the few XAI techniques that works for all tabular data, text and images (Molnar 2020). Our experiments were also conducted on different types of datasets that derived from the original LIME tutorials, e.g., the Boston house-price dataset, the breast cancer Wisconsin dataset (with both numerical and categorical features), and a standard InceptionV3 CNN pretrained on the ImageNet dataset. Due to the page limit, we only present typical results in this paper where the general observations hold for all experiments (cf. our BayLIME project website[4] for more).

To answer RQ1 and RQ2, in addition to LIME, we select a set of BayLIME explainers with different options and prior parameters. For each explainer, we iterate the explanation of the given instance k times, and quantify the inconsistency according to (3) and Kendall's W as well. While, we treat the perturbed sample size n as an independent variable in these experiments to confirm our earlier observation that inconsistency is an even severer problem when n has to be limited (e.g., due to efficiency constraints, cf. Fig. 1(B)).

For RQ3, we firstly define an interval $[l_{lo}, l_{up}]$ as the empirical bounds of all possible kernel width settings for the given application. We randomly sample pairs of kernel width parameters from that interval. Then, for each pair, we can calculate the "distance" of the two explanations[5]. Finally, we obtain a sample set of ratios between the perturbed distances of explanations and the kernel width pair, on which statistics provides insights on the general robustness, e.g., the median value as used by (5).

To recap, the prior knowledge used in our experiments is obtained from previous LIME explanations of a set of *similar* instances, cf. the discussion at the end of Section 5.3.

All experiments were run on a Windows 10 Pro 64-bit machine with Intel 1.80GHz 4 cores i7-8550U CPU and 16GB

[4] https://github.com/x-y-zhao/BayLime.

[5] In RQ3, we fix the perturbed sample size to a sufficiently large number, e.g., $n = 1000$ (as shown in RQ2), to minimise the impact of inconsistency.

RAM. The source code, data used and the full experimental results are publicly available at our BayLIME website.

6.2 Results and Discussion

RQ1. The red curves in the 1st row of Fig. 2 present the inconsistency measurements as a function of the perturbed sample size n. Although it decreases quickly, we observe very high inconsistency when the perturbed samples size n is relatively small (e.g., $n < 200$). Then the inconsistency continually drops and shows a converging trend. These results clearly support our earlier conjecture on the inconsistency issue of LIME, especially when the perturbed samples size n has to be limited by an upper-bound due to considerations on, e.g., timing restriction, energy consumption etc.

Moreover, apart from BayLIME with non-informative priors, BayLIME with partial/full informative priors with different prior parameters all significantly ease the inconsistency issue in general, although their effectiveness varies in detail (for which we will discuss in RQ2).

To further confirm the revealed inconsistency issue that has been dealt with by BayLIME is not unique to our new measure (3), rather indeed a real and fundamental problem, we also plot a similar (but more established) statistics – Kendall's W which ranges from 0 (no agreement) to 1 (complete agreement). As expected, the plots of Kendall's W in the 2nd row in Fig. 2 can be interpreted as the same as above. But Kendall's W only considers the (discrete) ranks of each feature and treats all features equally. In contrast, our measure (3) *weights* the IoD of the discrete ranks of each feature by its (continuous) importance. With the extra information considered (compared to Kendall's W), our measure (3) thus is more superior in terms of: (i) discriminating explanations with the same ranks of features but the importance vectors are different; (ii) minimising the effect from the normal fluctuation of the ranks of less-important/irrelevant features.

RQ2. As discussed in RQ1, compared to LIME, both BayLIME with partial and full informative priors improve remarkably on the consistency, while BayLIME with non-informative priors does not. This is not surprising, since both LIME and BayLIME with non-informative priors are *only* exploiting the information from the new data that generated randomly – The presence of randomness means that the results of sampling cannot be duplicated if the process were repeated, Naturally, the more sparse the samples are, the greater randomness presents in the dataset.

By contrast, BayLIME with partial/full informative priors can "average out" the sampling noise in the new generated data by combining the information from the priors, in a Bayesian principled way, cf. Eq. (15) (or its simplified case (18)). Combining informative prior knowledge may reduce the inconsistency – imagining an extreme case in which the posteriors are fully determined by the given priors, then repeated explanations would be perfectly consistent.

To closely inspect how effectively BayLIME with different priors affects the consistency, we first need the auxiliary of the factor λ/α. It is a known result that λ/α can be treated as a *regularization coefficient* in Bayesian linear regressors (Bishop 2006, Chpt. 3.3), meaning a larger λ/α penalises more on the training data (to control over-fitting). Indeed, this aligns well with our Remark 1: (i) when $\alpha \simeq 0$, the factor $\lambda/\alpha \to +\infty$ meaning a huge penalty on the data, thus the prior knowledge effectively dominates the posteriors; (ii) when $\lambda \simeq 0$, $\lambda/\alpha \to 0$ meaning no penalty on the data, thus the posteriors is dominated by the new observations.

Both the plots of BayLIME with full informative priors (yellow curves) in Fig. 2(A) and (C) have a regularization factor $\lambda/\alpha = 20$, and are basically identical. This is because, once $\lambda/\alpha = 20$ is fixed, the proportion of contributions to the posteriors by the priors and the new data is fixed. In other words, given n samples, the ability of "averaging out" sampling noise by the priors is fixed. When λ/α increases to 200, as shown by the yellow curves in Fig. 2(B) and (D), such ability of averaging out sampling noise is even stronger, which explains why the inconsistency measurements in this case are even lower than the case of $\lambda/\alpha = 20$.

For BayLIME with partial informative priors (blue curves in Fig. 2), we observe that smaller λ results in worse consistency, e.g., Fig. 2(C) vs (D). Again, Remark 1 applies here – smaller λ implies less contributions from the priors to the posteriors, meaning with less ability to average out the randomness in the new data.

Starting from a non-zero small number, as n increases, we can see: (i) LIME and BayLIME with non-informative priors have *monotonic trends*; and (ii) BayLIME with partial/full informative priors might exhibit an uni-modal pattern in general, e.g., the Fig. 2(G)[6] with a minimum point. The former is simply because that the more data collected the less sampling noise in repeated runs. While the latter represents a tension between the consistent prior knowledge and consistent MLE based on large samples. There must be a "balance-point" in-between that compromises both ends of the tension, yielding a minimised consistency. Finally, when $n \to +\infty$, it is trivial to see (e.g., by taking the limit of (18) as a function of n), all plots of inconsistency (and Kendall's W) will eventually converge (to the measurement based on MLE using infinitely large samples).

RQ3. Fig. 3 are box-and-whisker charts providing insights on the general robustness of eight AI explainers to kernel width settings, in which the median values defined by (5) are also marked, as usual, by horizontal bars inside the boxes.

Again, LIME and BayLIME with non-informative priors exhibit similar robustness, since there is no prior knowledge being considered rather the data solely determines the explanations of both. In stark contrast, when either partial or full prior knowledge is taken into account, we observe an obvious improvement on the robustness to kernel settings.

The regularisation factor λ/α and Remark 1 are still handy here in the discussions on how varying the λ and α affects the robustness – It all boils down to how much contribution from the priors (that is independent from kernel setting), compared with the contribution from the new data (that is sensitive to kernel settings), to the posteriors.

[6]Other plots may not show the pattern clearly due to the range/scale of the axes and inevitable noise in the experiments.

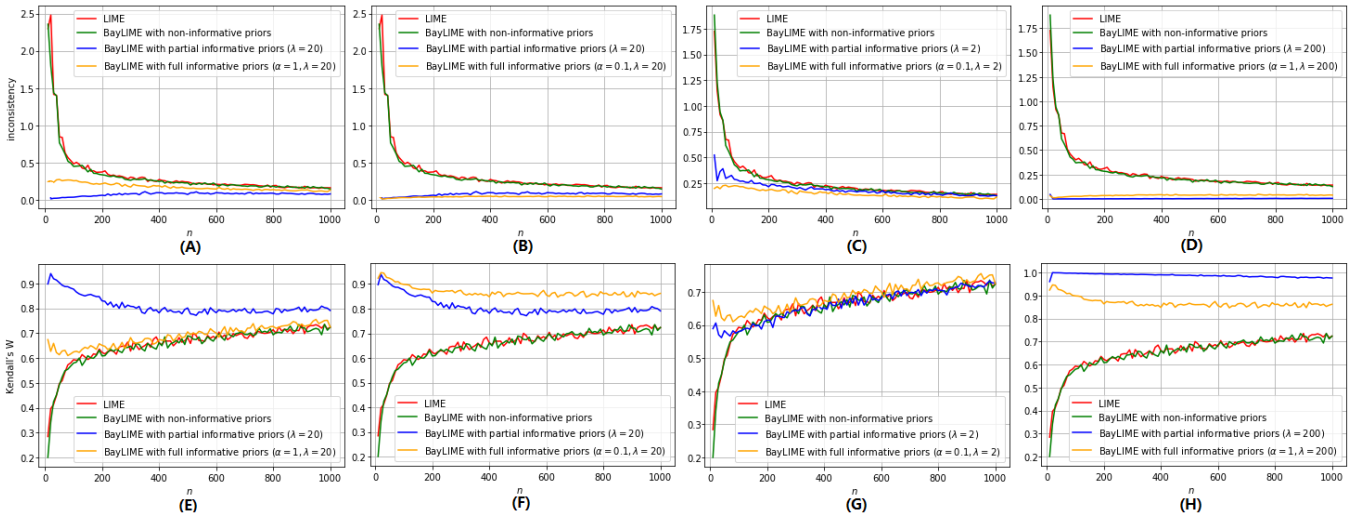

Figure 2: The inconsistency (3) (1st row) and Kendall's W (2nd row) in $k = 200$ repeated explanations (of an instance from the Boston house-price dataset) by LIME and BayLIME. Each column shows a representative combination of the α and λ.

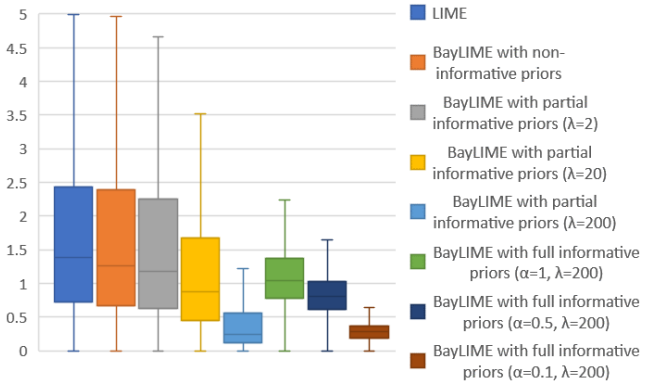

Figure 3: The general (un-)robustness of eight AI explainers to kernel settings (box-and-whisker plots without outliers).

7 Related Work

Many works have been done on improving aspects of LIME: Authors of (Shi, Zhang, and Fan 2020) modify the perturbed sampling method of LIME to cope with the correlation between features. RL-LIM (Yoon, Arik, and Pfister 2019) employs reinforcement learning to select a subset of perturbed samples to train the local surrogate model. MAPLE (Plumb, Molitor, and Talwalkar 2018) uses local linear regressors along with random forests that determines weights of perturbed instances to optimise the local linear model. Both DLIME (Rehman Zafar and Mefraz Khan 2019) and ALIME (Shankaranarayana and Runje 2019) share the similar concern with us on the inconsistency of repeated explanations. DLIME replaces the random perturbation with deterministic methods and ALIME employs an auto-encoder as a better weighting function for the local surrogate model. While, BayLIME seeks inherently different solutions in a Bayesian way which has the unique advantage of embedding prior knowledge (thus direct experimental comparisons between them are not sensible – prior knowledge plays an important role in BayLIME that cannot be represented by other two methods). Moreover, BayLIME can deal with strict efficiency constraints, while DLIME and ALIME cannot.

To the best of our knowledge, the only model-agnostic XAI technique with a Bayesian flavour is in (Guo et al. 2018), which aims at deriving generalised insights for a ML model through a *global* approximation (by a Bayesian nonparametric regression mixture model with multiple elastic nets). That approach can only fit conjugate priors empirically from data, and normally requires a large sample size for the inference of the large number of model parameters. By contrast, BayLIME is the first to exploit informative priors for better consistency and robustness to kernel settings.

8 Discussion, Conclusion & Future Work

BayLIME is a Bayesian modification of LIME that provides a principled mechanism to combine useful knowledge (e.g., from other diverse XAI methods, embedded human knowledge in the training of the AI/ML model under explanation or simply previous explanations of similar instances), which is a clear trend in AI (Ilyas et al. 2019; Lamb et al. 2020; Huang, Zhao, and Huang 2020). Such combination benefits the consistency in repeated explanations of a single prediction, robustness to kernel settings and may also improve the efficiency by requiring less queries made to the AI/ML model. That said, we discuss the following questions to highlight the practical usefulness of BayLIME.

Where to get the priors? In this very first Bayesian framework paper, we on purpose discuss four general ways of obtaining priors (cf. the end of Sec. 5.3), all of which we believe are feasible in practice, and implemented one of them in our experiments. For practical instantiation of our framework, the answer to this question varies case by

case, depending on the application-specific context. To give a more concrete example we plan to do in future: (i) Say the application is to explain several frames in a short video of a running dog; (ii) The 1st frame is explained by LIME with n_1 perturbed samples, yielding an importance vector of features μ; (iii) Then to explain the 2nd frame, we use BayLIME with partial-informative-priors by setting $\lambda = n_1$ and $\mu_0 = \mu$, and fit α from new n_2 perturbed samples (note, thanks to the priors, n_2 may be substantially smaller than n_1 for a better efficiency); (iv) The posterior distribution of the feature importance vector of the 2nd frame can be used as the priors of the 3rd frame and so on. We need to take extra care here, e.g., tracking the same features among frames to allow the "flow" of prior/posterior knowledge.

How do we know that the prior is good? To answer this, first let us be clear on what do we mean by "good" priors – is it the prior that truly/accurately reflects the unknown behaviour of the AI/ML model under explanation? Or the prior gives an explanation that looks good to human users? The latter definition is "cheating" in a Bayesian sense (you cannot rig a prior to get a result you like). While the former is more sensible, but we would never know the prior is good or not due to the lack of *ground truth* behaviours of the black-box AI/ML model. Thus, as a Bayesian, the best we can do is to combine all existing evidence, including both the prior knowledge (either good or bad that is unknown to us) and the new observations/samples, to yield an estimation.

What if we used a bad prior? Reusing the sensible definition of good priors above, indeed, there could be the case we are using a bad prior that does not describe the AI/ML-model's ground truth well and introduces bias. In this case, BayLIME may end up an explanation that is "consistently and robustly bad", which seems to imply that BayLIME is only useful when we are *certain* the priors are good. However, in practice, we can never be certain a prior is good or bad, rather the prior is simply a piece of evidence to us. It is not only against the Bayesian spirit but also unwise to discard any evidence without sound reasons. If there is a proof that the evidence (either the priors or the new observations) is not trustworthy, then certainly we should not consider it in our Bayesian inference – in this sense, all Bayesian methods depends on good-quality evidence (not just BayLIME). Moreover, how to collect unbiased evidence is clearly out of the scope of the Bayesian model itself. A simple example to emphasise the point – Bayesian inference has been widely and successfully used in the reliability assessment of safety-critical systems when seeing failure data (Zhao et al. 2020), while how to guarantee the system's failure recording component is logging data correctly is not the problem of the Bayesian assessment model.

Why not just use the good priors as the explanation results? Again, we cannot be sure if the priors are good or not due to the black-box nature of the underlying AI/ML model. From a Bayesian point of view, if we do have the budget to collect new evidence, then it should be done to update our beliefs. In our case, depending on how the priors are obtained, we can imagine situations in which priors alone might be not as good as the posteriors of BayLIME – for instance, when the priors are obtained from a gradient-based or global-interpretable XAI method, we may want to use BayLIME to incorporate some local/perturbation-based information to cover its theoretical limitations. We plan to do concrete case studies here to confirm this conjecture.

In summary, in this first BayLIME framework paper, we leave the problems regarding how to obtain good priors for BayLIME as open questions. As the emerging of techniques peeking inside AI/ML models (not necessarily from the XAI community), it will lead to many novel ways of obtaining knowledge of the black-box behaviour. Yet, the key challenge is how to combine the diverse knowledge obtained. This is precisely what BayLIME addresses.

9 Acknowledgements

This work is partially supported by the UK EPSRC through the Offshore Robotics for Certification of Assets [EP/R026173/1].

References

Adadi, A.; and Berrada, M. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6: 52138–52160. doi:10.1109/ACCESS.2018.2870052.

Alvarez-Melis, D.; and Jaakkola, T. S. 2018. On the Robustness of Interpretability Methods. *arXiv e-prints* arXiv:1806.08049.

Bernardo, J. M.; and Smith, A. F. M. 1994. *Bayesian theory*. Wiley. ISBN 0-471-92416-4.

Bishop, C. M. 2006. *Pattern recognition and machine learning*. Information science and statistics. Springer.

Branson, S.; Perona, P.; and Belongie, S. 2011. Strong supervision from weak annotation: Interactive training of deformable part models. In *2011 International Conference on Computer Vision*, 1832–1839. Barcelona, Spain: IEEE. doi:10.1109/ICCV.2011.6126450.

Branson, S.; Wah, C.; Schroff, F.; Babenko, B.; Welinder, P.; Perona, P.; and Belongie, S. 2010. Visual Recognition with Humans in the Loop. In Daniilidis, K.; Maragos, P.; and Paragios, N., eds., *Computer Vision – ECCV 2010*, volume 6314 of *LNCS*, 438–451. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-15561-1. doi:10.1007/978-3-642-15561-1_32.

Cawley, G. C.; and Talbot, N. L. 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research* 11: 2079–2107.

Filieri, A.; Ghezzi, C.; and Tamburrelli, G. 2012. A formal approach to adaptive software: Continuous assurance of non-functional requirements. *Formal Aspects of Computing* 24(2): 163–186. ISSN 1433-299X. doi:10.1007/s00165-011-0207-2.

Guo, W.; Huang, S.; Tao, Y.; Xing, X.; and Lin, L. 2018. Explaining Deep Learning Models – A Bayesian Non-parametric Approach. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 4514–4524. Curran Associates, Inc.

He, K.; and Sun, J. 2015. Convolutional Neural Networks at Constrained Time Cost. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'15, 5353–5360.

Holland, P. W. 1973. Weighted Ridge Regression: Combining Ridge and Robust Regression Methods. Working Paper 11, National Bureau of Economic Research. doi: 10.3386/w0011.

Huang, W.; Zhao, X.; and Huang, X. 2020. Embedding and Extraction of Knowledge in Tree Ensemble Classifiers. *arXiv e-prints* arXiv:2010.08281.

Huang, X.; Kroening, D.; Ruan, W.; Sharp, J.; Sun, Y.; Thamo, E.; Wu, M.; and Yi, X. 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review* 37: 100270. ISSN 1574-0137. doi:https://doi.org/10.1016/j.cosrev.2020.100270.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems 32*, NIPS'19, 125–136. Curran Associates, Inc.

Kendall, M. G.; and Smith, B. B. 1939. The problem of m rankings. *The Annals of Mathematical Statistics* 10(3): 275–287. ISSN 00034851.

Lamb, L. C.; Garcez, A.; Gori, M.; Prates, M.; Avelar, P.; and Vardi, M. 2020. Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective. *arXiv e-prints* arXiv:2003.00330.

Laugel, T.; Renard, X.; Lesot, M.-J.; Marsala, C.; and Detyniecki, M. 2018. Defining locality for surrogates in post-hoc interpretability. In *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 47–53. Stockholm, Sweden.

MacKay, D. J. C. 1992. Bayesian Interpolation. *Neural Computation* 4(3): 415–447. doi:10.1162/neco.1992.4.3.415.

Molnar, C. 2020. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* E-book on leanpub.com.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.

Plumb, G.; Molitor, D.; and Talwalkar, A. S. 2018. Model Agnostic Supervised Local Explanations. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 2515–2524. Curran Associates, Inc.

Rehman Zafar, M.; and Mefraz Khan, N. 2019. DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems. In *Proc. of SIGKDD Workshop on Explainable AI/ML (XAI) for Accountability, Fairness and Transparency*, 6. ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1135–1144. New York, NY, USA: Association for Computing Machinery. doi:10.1145/2939672.2939778.

Shankaranarayana, S. M.; and Runje, D. 2019. ALIME: Autoencoder Based Approach for Local Interpretability. In Yin, H.; Camacho, D.; Tino, P.; Tallón-Ballesteros, A. J.; Menezes, R.; and Allmendinger, R., eds., *Intelligent Data Engineering and Automated Learning – IDEAL 2019*, volume 11871 of *LNCS*, 454–463. Cham: Springer International Publishing. doi:10.1007/978-3-030-33607-3_49.

Shi, S.; Zhang, X.; and Fan, W. 2020. A Modified Perturbed Sampling Method for Local Interpretable Model-agnostic Explanation. *arXiv e-prints* arXiv:2002.07434.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proc. of the IEEE conference on computer vision and pattern recognition*, CVPR'16, 2818–2826.

Tipping, M. E. 2001. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research* 1(Jun): 211–244.

Walter, G.; and Augustin, T. 2009. Imprecision and prior-data conflict in generalized Bayesian inference. *Journal of Statistical Theory and Practice* 3(1): 255–271. doi:10.1080/15598608.2009.10411924.

Yoon, J.; Arik, S. O.; and Pfister, T. 2019. RL-LIM: Reinforcement Learning-based Locally Interpretable Modeling. *arXiv e-prints* arXiv:1909.12367.

Zhao, X.; Robu, V.; Flynn, D.; Dinmohammadi, F.; Fisher, M.; and Webster, M. 2019. Probabilistic model checking of robots deployed in extreme environments. In *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, volume 33, 8076–8084. Honolulu, Hawaii, USA. doi:https://doi.org/10.1609/aaai.v33i01.33018066.

Zhao, X.; Salako, K.; Strigini, L.; Robu, V.; and Flynn, D. 2020. Assessing safety-critical systems from operational testing: A study on autonomous vehicles. *Information and Software Technology* 128: 106393. ISSN 0950-5849. doi:10.1016/j.infsof.2020.106393.