Received XX XX XXXX; | Revised XX XX XXXX; | Accepted XX XX XXXX

DOI: xxx/xxxx

RESEARCH ARTICLE

Visual Diagnostics of an Explainer Model – Tools for the Assessment of LIME Explanations

Katherine Goode*[1] | Heike Hofmann[1,2]

[1]Department of Statistics, Iowa State University, Iowa, United States
[2]Center for Statistics and Applications in Forensic Evidence (CSAFE), Iowa State University, Iowa, United States

Correspondence
*Katherine Goode, Department of Statistics, Iowa State University, Ames, IA. Email: kgoode@iastate.edu

Abstract

The importance of providing explanations for predictions made by black-box models has led to the development of explainer model methods such as LIME (local interpretable model-agnostic explanations). LIME uses a surrogate model to explain the relationship between predictor variables and predictions from a black-box model in a local region around a prediction of interest. However, the quality of the resulting explanations relies on how well the explainer model captures the black-box model in a specified local region. Here we introduce three visual diagnostics to assess the quality of LIME explanations: (1) explanation scatterplots, (2) assessment metric plots, and (3) feature heatmaps. We apply the visual diagnostics to a forensics bullet matching dataset to show examples where LIME explanations depend on the tuning parameter values and the explainer model oversimplifies the black-box model. Our examples raise concerns about claims made of LIME that are similar to other criticisms in the literature.

KEYWORDS:
explainable machine learning, black-box models, interpretability, statistical graphics, data science

1 | INTRODUCTION

In the field of statistics, there are two main uses for models: inference and prediction. Machine learning models have proven to perform well in problems with the latter objective, but the accuracy of many machine learning models comes at the cost of interpretability due to their algorithmic complexity (hence the phrase "black-box models"). Model interpretability allows for the understanding and assessment of how a model produces predictions. The lack of the ability to understand and assess a model makes it difficult to trust the model, especially in areas with high stakes decisions such as the medical and forensics sciences. The increased use of machine learning models in applications and the introduction of the General Data Protection Regulation (GDPR) in 2018 [8] has resulted in a dramatic increase in explainable machine learning research, which focuses on developing ways to explain output from machine learning algorithms.

Throughout this paper, we distinguish between interpretability and explanability of models. We define *interpretability* as the ability to directly use model parameters to understand the relationships in the data captured by the model: e.g., a linear model coefficient associated with a predictor variable indicates the amount the response variable changes based on a change in the predictor. In contrast, we define *explanability* as the ability to use the model in an indirect manner to understand the relationships in the data captured by the model: e.g., partial dependence plots depict the marginal relationship between model predictions and predictor variables [6].

Numerous methods have been proposed to provide explanations for black-box model predictions [7, 10, 17, 18]. Some are specific to one type of model (e.g. [24, 28]), others are model-agnostic (e.g. [5, 25]). In this paper, we focus on the model-agnostic method of LIME [22].

LIME (local interpretable model-agnostic explanations) uses a surrogate model to relate predictor variables to black-box model predictions [22]. While some explainer models

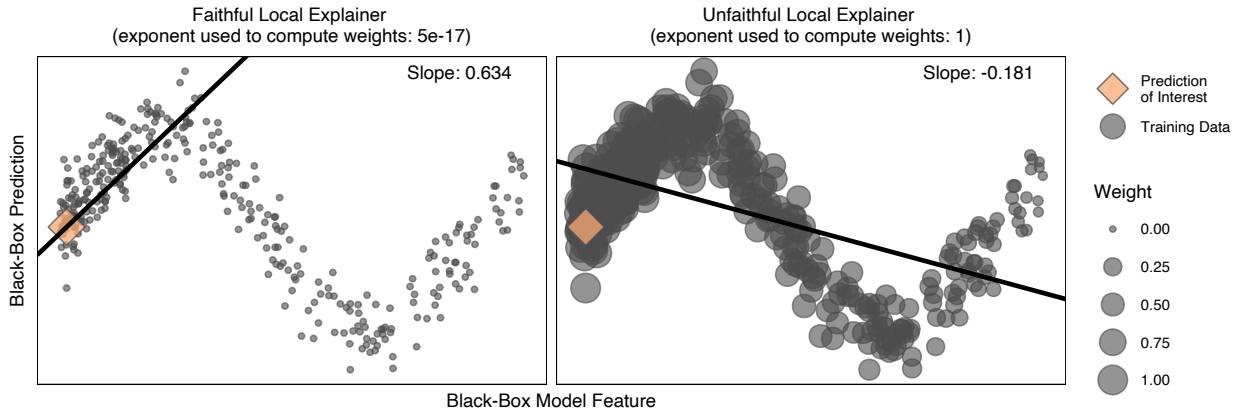

FIGURE 1 Hypothetical scenarios depicting the effect of different weights on LIME. (Left) An example of an explainer model with appropriate weights to provide a faithful approximation of the complex model. (Right) An example of an explainer model for the same prediction but with different weights that results in an explanation unfaithful to the complex model.

focus on understanding a model at the global level, LIME claims to provide explanations for individual predictions (local). Additionally, LIME is designed to work with any model (model-agnostic) and to produce easily understandable results (explanations) [22]. Conceptually, LIME fits a simple (interpretable) model, the explainer model, to approximate the complex model in a local region around a prediction of interest. The simple model is interpreted to identify the variables that most influence the complex model prediction.

While the concept of LIME is relatively simple, a practical implementation of LIME is not straightforward, and research is being done to improve the procedure [15]. The current implementations of LIME [19, 21] offer various tuning parameters (see Section 2) that affect the explainer model and ultimately, the explanation. Since the explainer model is an approximation of the complex model and not a direct interpretation, the explanations produced by an explainer model are subject to the quality of the approximation. In order to achieve reasonable explanations, the tuning parameter values selected should be assessed.

To demonstrate the effect of a tuning parameter on LIME, we consider two different explainer models applied to the same prediction. Figure 1 depicts two plots of the predictions from a hypothetical black-box model versus the feature used to train the model. The location of a prediction of interest is indicated by the diamond shaped points. Globally, there is a clear nonlinear relationship between the predictions and the feature, but the relationship could be approximated by a linear model in a local region around the prediction of interest. In both scenarios, a linear regression weighted by the proximity from an observation to the prediction of interest is used as the explainer model, but the proximities are computed differently as depicted by the varying sizes of the observations in the two plots.

On the left, the distance between an observation and the prediction of interest is computed using the Gower distance metric [9] and then raised to the power of 5×10^{-17} to emphasize a very local region. The distances are subtracted from 1 to obtain a proximity. Here, the explainer model is faithful to the complex model since it captures the linear relationship between the black-box predictions in an immediate neighborhood of the prediction of interest. The slope explains that the black-box predictions increase as the feature increases in the local region around the prediction of interest.

For the scenario on the right, the computation of the proximities is the same except the distances are raised to a power of 1 (the default exponent in the *lime* R package [19]). This causes the observations that are further away from the prediction of interest to be given larger weights than the scenario on the left. Here, the explainer model does not capture the linear trend near the prediction of interest, but instead, a more global trend. Without an assessment of this explainer model, the slope incorrectly suggests that the predictions decrease as the feature increases near the prediction of interest.

Figure 1 demonstrates a simple example in which the explanation quality varies drastically when different weights are used. Other concerns LIME have been raised in the literature: Laugel et al. [15] and Molnar [18] discuss the difficulty specifying a local region with LIME due to both an unclear definition of a "local region" and how to apply LIME to achieve an appropriate local region. Laugel et al. [15] also present examples of LIME explanations that are clearly global and not local. A different concern about LIME is raised by Alvarez-Melis and Jaakkola [1] pertaining to the robustness of explanations from LIME and other explainer models: they find that even small changes in predictor variables can lead to very different LIME explanations. Even the original authors of

LIME, Ribeiro et al. [22], acknowledge that if a linear model is used as the explainer, LIME relies on a linear approximation of the explainer model to the complex model and state "if the underlying model is highly non-linear even in the locality of the prediction, there may not be a faithful explanation".

Without an assessment of the LIME explainer model, the user is putting trust in another black-box algorithm to explain the complex black-box model. In this paper, we stress the importance of assessing the LIME explainer model. To do this, we lay out the set of claims about LIME made by Ribeiro et al. [22], and we provide a toolkit of visual diagnostics for the assessment of these claims: (1) *explanation scatterplots*, (2) *feature heatmaps*, (3) *assessment metric plots*. Using these visual diagnostics, we identify situations in which LIME does not meet its claims. In some simple examples, we demonstrate shortfalls in LIME explanations where the explanations are highly dependent on tuning parameter values without a clear way to identify a set of tuning parameter values that produce the "best" explanations.

While LIME is implemented for image, tabular, and text data, we only focus on tabular data. For additional simplicity, we only discuss classification prediction models with a dichotomous response and continuous predictors. However, the proposed diagnostics may be extended to other situations.

The remainder of the paper is structured as follows. Section 2 provides background and claims made by Ribeiro et al. [22] about LIME. We introduce the suggested diagnostic plots in Section 3. Then in Section 4, we demonstrate the use of the diagnostics to assess LIME explanations for a random forest fit to a forensics bullet matching dataset. Section 5 concludes with a discussion on extensions and limitations of the diagnostic plots and concerns about LIME in regards to the claims made by Ribeiro et al. [22] brought about by the visualization examples in this paper that agree with Alvarez-Melis and Jaakkola [1], Laugel et al. [15], and Molnar [18].

All runs of LIME in this paper are executed in a forked version (https://github.com/goodekat/lime) of the R package *lime* (version 0.5.1) by Pedersen and Benesty [19]. The forked version is functionally indistinguishable from Pedersen and Benesty's implementation but allows us to export internal values relevant for an assessment of the explainer. The code for the diagnostic plots is stored in the R package *limeaid* available on GitHub (https://github.com/goodekat/limeaid).

2 | BACKGROUND ON LIME

The general form of the LIME algorithm can be divided into three steps [see also 15]:

1. *Data Simulation and Interpretable Transformation*: Simulate a dataset from the original data used to fit the black-box model. Apply a transformation to the simulated data and the prediction of interest that will allow for interpretable explanations.
2. *Explainer Model Fitting*: Apply the black-box model to the simulated data to obtain predictions. Compute the distance between each of the simulated data points and the prediction of interest. Perform feature selection. Fit an interpretable model with the black-box predictions from the simulated data as the response, the selected features from the transformed simulated data as the predictors, and the distances as weights. This model is the explainer model.
3. *Explainer Model Interpretation*: Interpret the explainer model to determine which features played the most important role in the prediction of interest.

Ribeiro et al. [22] provide an implementation of LIME in a Python package [21]. An adaption of the Python package in R has been implemented and made available by Pedersen and Benesty [19]. In both implementations, a ridge regression model is used as the explainer model, and the user selects various tuning parameter options: the number of features to return in the explanation, the simulation method, the feature selection method, and how the weights are computed. See Appendix A for an overview of the options available for the tuning parameters based on the R package.

In the original paper, Ribeiro et al. [22] make the following set of claims regarding the performance of LIME:

- *Interpretability*: The explainer model can be easily interpreted to provide meaningful explanations.
- *Faithfulness*: The explainer model sufficiently captures the relationship between the complex model predictions and the features in the local region around a prediction of interest to produce explanations that are faithful to the complex model.
- *Linearity*: By using a ridge regression model as the explainer model, it is assumed that there is a linear relationship between complex model predictions and the features in the local region around a prediction of interest.
- *Localness*: The explanations produced by LIME are local in regards to a prediction of interest.

The assumption of interpretability only depends on the complexity of the model used as explainer model. If the model is too complex to provide meaningful explanations (e.g. there are too many variables in the model), it is clear that the assumption of interpretability is violated. The other three assumptions are not as easy to assess, and for those, we suggest the use of diagnostic plots.

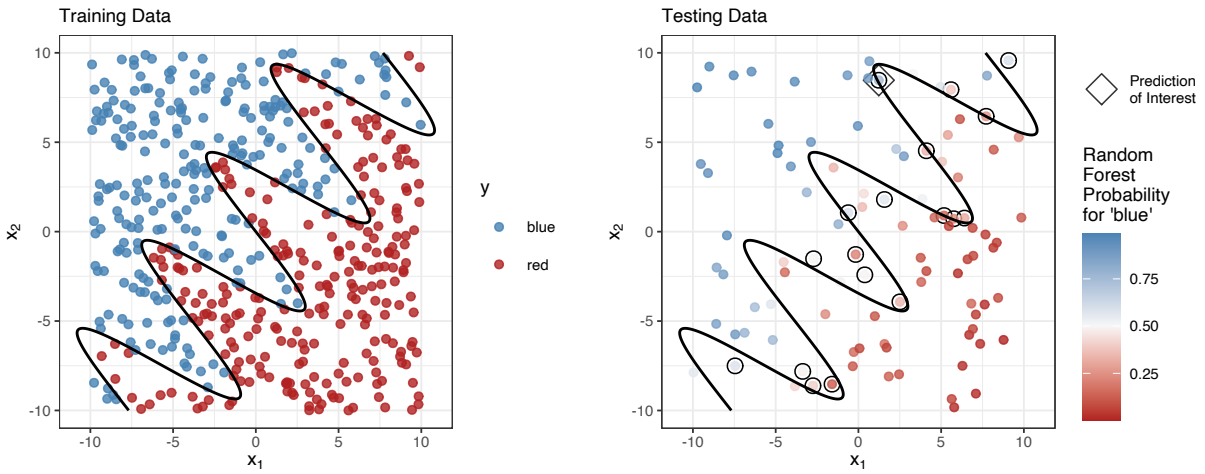

FIGURE 2 Plots of x_2 versus x_1 from the sine data training (left) and testing (right) sets introduced in Section 3 with the true classification boundary shown as a solid black line. In the testing set, the open black circles identify cases misclassified by the random forest, and the diamond indicates a prediction of interest.

3 | VISUAL DIAGNOSTICS FOR LIME

In this section, we introduce three visual diagnostic plots that assess the LIME claims from different perspectives:

1. *Explanation Scatterplot* (Section 3.1): Comparison of the explainer and complex models for an individual prediction of interest.
2. *Feature Heatmap* (Section 3.2): Comparison of features selected by LIME across applications of LIME with different tuning parameter values.
3. *Assessment Metric Plot* (Section 3.3): Comparison of performance metrics for LIME across ~~applications of LIME with~~ different tuning parameter values.

The sine data

To demonstrate the visual diagnostics, we generate an example dataset that will be referred to as the sine data. The sine data contains 600 observations with features of x_1 and x_2 independently sampled from Unif(-10, 10) distributions and x_3 sampled from a N(0, 1) distribution. A binary response variable y is created using a sine curve such that

$$y = \begin{cases} \text{blue} & \text{if } x_2' > \sin\left(x_1'\right) \\ \text{red} & \text{if } x_2' \leq \sin\left(x_1'\right) \end{cases}. \quad (1)$$

where $x_1' = x_1\cos(\theta) - x_2\sin(\theta)$, $x_2' = x_1\sin(\theta) + x_2\cos(\theta)$, and $\theta = -0.9$. Note that due to the creation of y in this manner, y is dependent on x_1 and x_2 and independent of x_3. The dataset is randomly divided into training and testing sets of 500 and 100 observations, respectively. A random forest is fit to the training set using the R package *randomForest* (version 4.6.14) [16] with the default settings and is applied to the test set to obtain predictions.

Figure 2 shows scatterplots of x_2 versus x_1 from the training data (left) and the testing data (right). Both plots include the true classification boundary as the solid black line. The training data are colored by the observed response variable (y), and the testing data are colored by random forest prediction probabilities. The random forest misclassifies 18 points, which are all located near the classification boundary and identified by open circles.

From these scatterplots, we see that the global relationship between response y and features x_1 and x_2 is linear: the probability for label blue increases with the difference between features x_2 and x_1. Locally, the relationship between y and features x_2 and x_1 varies a lot more around the line of identity. Here, the relationship is determined by the sine wave. However, the sine is a good-natured function that can be approximated well linearly in local regions as shown in Figure 1 .

We apply LIME using six sets of tuning parameter values to all observations in the sine data test set to observe explanation variability. Five of the LIME applications use a quantile bin based simulation method (samples are simulated uniformly from a feature using a specified number of quantile bins) with the number of bins varying from 2 to 6 by application. We use 6 bins as the maximum, because the complexity of the explanations increases with the number of bins. Note that 4-quantile-bins is the default method in the *lime* R package. The sixth application of LIME uses a kernel density simulation method (samples are drawn from kernel density approximations of the feature distributions). The default methods for feature selection (forward selection) and the computation of

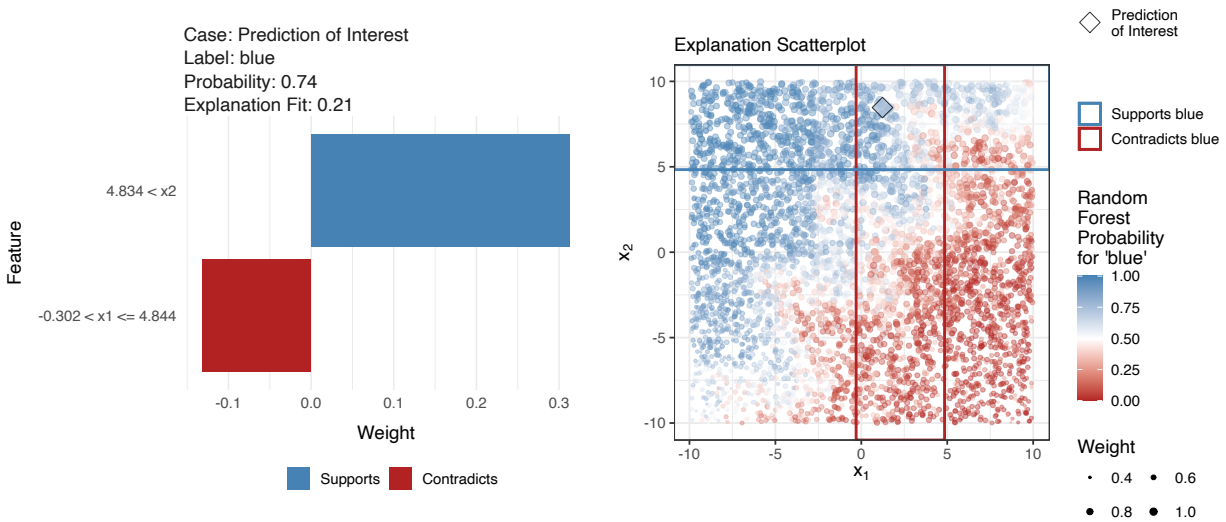

FIGURE 3 (Left) Visualization from the *lime* R package of a LIME explanation for the sine data prediction of interest identified in Figure 2 . (Right) *Explanation Scatterplot*. An explanation scatterplot associated with the explanation on the left. The 4-quantile-bins over-simplify the relationship between the random forest predictions and the x_1 and x_2 values near the prediction of interest.

the weights (Gower distance raised to an exponent of 1) are used for all applications. See Appendix A for more detailed descriptions of the tuning parameters.

For the presentation of the explanation scatterplot, we focus on the misclassified point indicated by a diamond in Figure 2 . Misclassified points are often of interest to explain since they may provide information about ways to improve the model. For the introduction of the other two plots, we consider the LIME explanations for all observations in the sine data test set.

A Visual Representation of a LIME Explanation

Before introducing the visual diagnostics, let us consider a commonly used visualization of a LIME explanation. Figure 3 (left) depicts the explanation for the prediction of interest indicated in Figure 2 obtained using 4-quantile-bins. In the text at the top, the "Probability" is the random forest probability of 0.74 that the observation of interest belongs to the "Label" category of blue. The "Explanation Fit" is the deviance ratio of 0.21 associated with the ridge regression explainer model (often interpreted as an R^2), which suggests that the explainer model is not a good linear fit.

When quantile bins are used to simulate data, LIME converts continuous predictor variables to indicator variables identifying whether the variable value falls in the same quantile bin as the prediction of interest or not. The indicator variables are used as the features in the ridge regression explainer model. The features included in the visualization are the ones selected by feature selection, the lengths of the bars represent the coefficients from the ridge regression associated with the indicator variables, while the color of a bar denotes the sign of the coefficient. This explainer model suggests that a random forest prediction of 'blue' for this observation is mostly supported by the prediction of interest having a value of x_2 that is greater than 4.834 but somewhat contradicted by the prediction of interest having a value of x_1 that is greater than -0.302 and less than or equal to 4.844.

3.1 | Explanation Scatterplots

As a first check to determine whether the explanation depicted in Figure 3 (left) is trustworthy, we note that since the support for blue by x_2 outweighs the support for red by x_1, the explainer model overall favors a label of blue for the prediction of interest agreeing with the random forest probability. In fact, both random forest and explainer model classify the observation incorrectly. However, based on the information in Figure 3 (left) alone, it is not possible to make an informed assessment of the explanation.

For a further assessment of the explainer model, we turn to an *explanation scatterplot*: a visual diagnostic for assessing the LIME claims of locality and fidelity for an individual explanation by juxtaposing the complex and explainer models in one plot. The format of an explanation scatterplot depends on the LIME simulation method. We introduce the explanation scatterplot here under the *lime* R package default method of 4-quantile-bins. See the supporting information for explanation scatterplot formats under other LIME simulation scenarios.

The explanation scatterplot applies the concept of plotting the model in the data space discussed in Wickham et al. [30]. The visualization is built by plotting the LIME simulated data

for the top two features identified by the explanation in a scatterplot and coloring these points by the predictions from the complex model. The point size represents the weight assigned by LIME. In order to show the LIME results for the observation of interest, lines are drawn on top of the points representing the boundaries of the indicator variables used to fit the explainer model. The line color denotes whether LIME indicates that a feature supports or contradicts a class prediction.

An explanation scatterplot corresponding to the LIME explanation depicted on the left in Figure 3 is shown on the right hand side of Figure 3 . By juxtaposing the random forest predictions and the explainer model boundaries, we are able to assess the faithfulness and localness of the explainer model.

First, consider the claim of localness. The weights decay relatively slowly outside of the intersection of the two quantile bins suggesting that the LIME explanation is highly influenced by points outside of the bins containing the prediction of interest. However, it is difficult to say if the claim of localness has been violated, because a definition of a local region is not specified. Depending on what region a viewer considers to be local, an argument could be made in favor of or against a violation of localness. The explanation scatterplot raises awareness of the unclear definition of a local region with LIME.

Now, consider the claim of faithfulness: It can be said that the majority of the points in the x_2 quantile bin support a prediction of blue, which is captured by the bar supporting a prediction of blue in the LIME explanation, and a similar statement can be made about the x_1 quantile-bin. These statements validate the explanation produced by LIME. However, the explanation scatterplot plot shows that the random forest performs well at capturing the sine curve classification boundary by creating various sized rectangles consisting of predictions with similar probabilities, which the LIME explanation does not pick up on since the bins are created without information about the random forest predictions. Thus, LIME does provide an explanation for the prediction, but it is a very poor explanation in terms of faithfulness to the random forest prediction regions.

It is difficult to assess linearity from this explanation scatterplot, but a residual plot of the explainer model could be used to check the linearity claim. See Appendix B for the residual plot associated with the explanation considered in Figure 3 , which shows a violation of the linearity claim.

This example explanation scatterplot only includes two features. In situations where more than two features are included in a LIME explanation, the explanation scatterplots can be extended to a generalized pairs plot [4] that includes all pairwise combinations of features. Generalized pairs plots (and scatterplot matrices in general) have diminishing value when the number of features increase [13] [26]. Machine learning models are commonly fit using a large number of features, and therefore, a generalized pairs plot of explanation scatterplots for all features would be ineffective. However, when applying LIME, the user selects the number of features to return in the explanation. In the *lime* R package, Pedersen and Benesty [19] encourage users to select less than 10 features. As long as a small number of features are returned in the LIME explanation, it is feasible to use a generalized pairs plot of explanation scatterplots. An example is shown in Section 4.3.

3.2 | Feature Heatmap

Explanations produced by LIME are likely to be affected by the choice of tuning parameter values. A hypothetical example of this is shown by Figure 1 where the method used to weight the observations influenced the explanation. As of the time of writing this manuscript, we have encountered no recommendations for how to specify the parameter values besides for the default settings in the *lime* R package. In order to compare the explanations produced by LIME using different tuning parameter values and provide another perspective for assessing localness, we visualize an overview of the explanations with the *feature heatmap* diagnostic plot.

The feature heatmap uses colors to identify the features selected by LIME across multiple predictions (referred to as cases here) and tuning parameter values organized by the feature importance assigned by LIME. That is, for LIME applied with t sets of tuning parameter values to n cases to select the f top features, create f heatmaps (one for each of the positions of importance determined by the magnitude of the explainer model coefficients) with the cases on the y-axis, the tuning parameter values on the x-axis, and the cells colored by the feature chosen for the corresponding case and tuning parameter value. Additional tuning parameters may also be included in the plot via facets.

When interpreting feature heatmaps, horizontal lines of the same color represent the ideal situation where the explanations are consistent and do not depend on the tuning parameter values but do depend on the location of the observation in the feature space. Vertical lines of color represent explanations that are dependent on the tuning parameter values and may not be local since the same feature is chosen regardless of the location of the observation in the feature space. See the supporting information for hypothetical examples depicting the extreme cases of feature heatmaps.

Figure 4 shows a feature heatmap for the LIME applications to the 100 observations in the sine data test set. The most important and second most important features selected by LIME are shown in in the top and bottom facets, respectively. For the quantile bins, the original features prior to being converted to indicator variables are included since it is obvious that different features would be selected when the sizes of

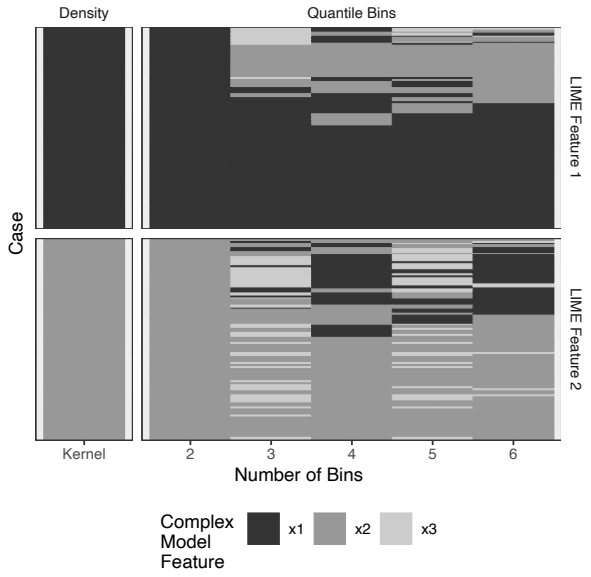

FIGURE 4 *Feature Heatmap*. An example feature heatmap from applying LIME to the sine data test set using different tuning parameter values. The vertical striping indicates that the LIME explanations are not consistent across tuning parameter values.

the bins change. This figure shows that for kernel density and 2-quantile-bins, LIME produces global explanations since x_1 is selected as the most important feature and x_2 is selected as the second most important feature across all cases in the test set. There is variability in the features selected by LIME for 3- to 6- quantile-bins suggesting more local explanations. There are signs of vertical striping, which suggests a dependence on tuning parameters. Note that the explanations from 3- and 5-quantile-bins include the selection of the random noise variable (x_3) as an important variable in many predictions~~, which would not be expected to be important to a random forest~~. It is unclear whether the random forest is using a variable that would not be expected to be important or LIME is incorrectly identifying the important variable when 3- and 5- quantile bins are used.

3.3 | Assessment Metric Plot

The feature heatmap for the sine data in the previous section shows an example of inconsistent LIME explanations across tuning parameter values. In this situation, the user must determine which set of explanations to trust. One way to do this is to compute assessment metrics for each set of explanations to identify the optimal tuning parameter values. We discuss three metrics for this purpose and present a visual comparison in an *assessment metric plot*.

Each metric presented below is computed on LIME explanations for a set of predictions obtained using the same tuning parameter values. Here we provide a high level description of the metrics. Notation and formulas for these metrics are included in Appendix C.

- *Average* R^2: Assesses the model fit and linearity claim by computing the average of the explainer model R^2 values (deviance ratios from the R package *glmnet*).
- *Average Fidelity*: Measures the faithfulness of the explainer model to the complex model by comparing their predictions. Computed as the average of the explainer model fidelity metrics: a metric presented in Ribeiro et al. [22] (the weighted distance between explainer and complex model predictions for all observations in the LIME simulated data associated with an individual prediction of interest).
- *Mean Squared Explanation Error (MSEE)*: Also measures the faithfulness of the explainer model to the complex model by comparing their predictions, but only the prediction of interest is used to compute an average squared deviation between explainer and complex model predictions.

Figure 5 shows an example assessment metric plot. The three metrics are computed for each of the LIME applications to the sine data test set. The simulation methods are listed on the x-axis. The plot is faceted by metric, and the metric values are plotted on the y-axis. The point color represent the rank of

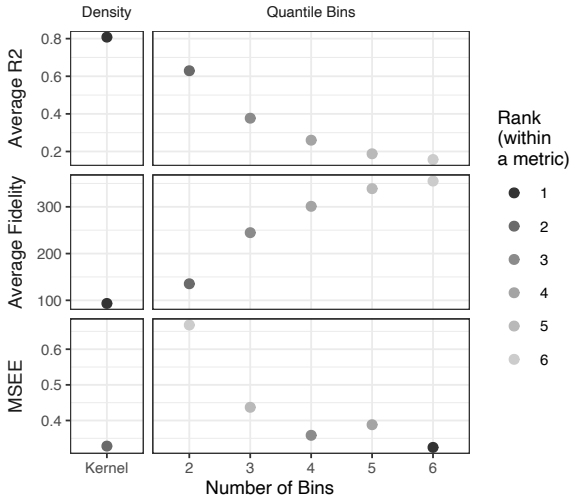

FIGURE 5 *Assessment metric plot*. This example assessment metric plot compares different applications of LIME to the sine data test set. The kernel density simulation method performs well across all three metrics.

the simulation methods performance based on a particular metric (darker indicates a better metric value and lighter indicates a worse metric value). Higher average R^2 values are better, and lower average fidelity and MSEE values are better. This example only includes one tuning parameter: the simulation method. If more than one tuning parameter is considered, the assessment metric plot is extended by adding additional facets, point shapes, or levels to the x-axis.

All three metrics suggest that the kernel density method performed well, but the metrics disagree for the quantile bins methods. Average R^2 and average fidelity rank the performance of the number of quantile bins the same (2-quantile-bins perform the best and 6-quantile-bins perform the worst). In fact, these two metrics appear to have a mirrored relationship in this example. MSEE provides almost the exact opposite results with 6-quantile-bins performing the best and 2-quantile-bins performing the worst. Average fidelity and MSEE are similar metrics, but MSEE only takes the prediction of interest into account and not the full simulated dataset as average fidelity does. This suggests that 6-quantile-bins produce an explainer model more faithful for the prediction of interest while 2-quantile-bins produce an explainer model more faithful over all simulated data observations (weighted by proximity to the prediction of interest).

Recall, Figure 4 shows the kernel density method returns the same explanations across all cases in the test set suggesting a global trend may be the best explanation for this example. This may be reasonable considering that we would expect x_1 and x_2 to be the two features used by the random forest to distinguish between response categories. However, further exploration of individual explanations using explanation scatterplots may help to identify the tuning parameter values that produce the most trustworthy explanations.

4 | APPLICATION TO BULLET MATCHING DATA

In this section, we apply the visual diagnostics for LIME explanations to a practical data problem investigating the similarity of marks on fired bullets.

4.1 | Bullet Matching Data

In current practice, forensic firearm examiners evaluate whether two bullets are from the same source (fired from the same gun) or from different sources based on microscopic comparison of the striation patterns engraved on bullets during the firing process (see Figure 6). The process is based on a visual and therefore subjective assessment of the evidence. The lack of objective evaluation and the associated absence of established error rates has first been criticized by the National Research Council [3] and later by the President's Council of Advisors on Science and Technology [20].

In response, Hare et al. [12] proposed an automated machine learning method for bullet matching to complement a visual inspection by firearm examiners. Based on high-resolution topological scans of land engraved areas, Hare et al. [12] obtain signatures of striations from two bullet lands (Figure 7). Nine features quantifying the similarity of signatures, such as the cross-correlation function, the distance between signatures, and the number of matching striae, are extracted and used to train a random forest to determine the probability of a comparison resulting from the same source (matching signatures) or from different sources (non-matching signatures). The model in Hare et al. [12] was trained on a set of scans of bullets from the James Hamby Consecutively Rifled Ruger Barrel Study [11], which included 10,384 land-to-land comparisons [12]. See the supporting information for additional information on the signature similarity features.

4.2 | Application of LIME to Bullet Matching Data

Since firearm identification is commonly used as evidence for convictions in court cases, it is important to be able to understand and assess a model used to quantify the probability that a bullet is fired from a gun. LIME explanations would provide a local explanation for an individual prediction, but just as it is important to assess the model for this high-stakes application, it is also important to assess the LIME explanations. We will demonstrate an assessment of LIME explanations using the visual diagnostics introduced in this paper.

A random forest model fit to an expanded dataset of 83,028 land-to-land comparisons from two sets of bullets in the James Hamby Consecutively Rifled Ruger Barrel Study [11] was validated in Vanderplas et al. [29]. Here, we train a random forest model to mimic the model validated in Vanderplas et al. [29] using the same data, model structure, and features (the nine similarity measures developed in Hare et al. [12]). We apply the trained random forest model to 6 bullets from another set of the Hamby study with 364 rows of land comparisons. See the supporting information for further descriptions of the data.

The newly trained random forest has an out-of-bag accuracy of 1 and out-of-bag false positive and false negative rates of 0.3 and 2×10^{-4}, respectively. On the test data, the random forest performance decreases with an accuracy of 0.85 and false positive and false negative rates of 0.55 and 0.011, respectively. This is an example where explanations of the model predictions could provide insight to the cause of the decrease in model performance on the test data.

FIGURE 6 (Top left) Traditionally rifled gun barrel. The grooves and lands alternate to give bullets a spin during the firing process, which create markings (striations) on a bullet when fired. (Top right) Image of a fired bullet. The vertical stripes along the lower half of the bullet show groove and land engraved areas. The land engraved areas contain the microscopic striations created when the bullet passed through the barrel of the gun. (Bottom) Close up of a land engraved area showing striations (vertical lines).

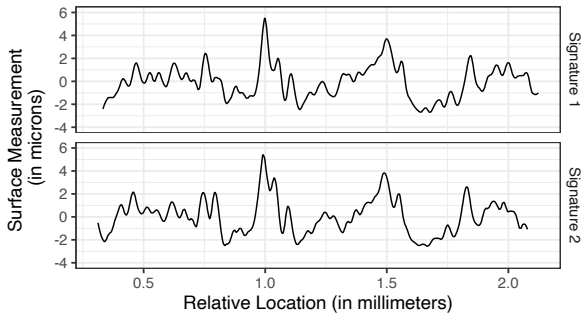

FIGURE 7 Example bullet signatures. The signatures are from the same land and therefore have very similar patterns.

In Figure 8 , we consider a global visualization of the relationship between the random forest predictions and the model features with a parallel coordinate plot of the training data (top row) and testing data (bottom row). In the training data we observe a clear difference in feature values based on whether the corresponding random forest scores are close to 1 or 0. High values of rough correlation, cross correlation function, and number of matches are indicative of random forest scores close to 1; similarly, low values of mismatches, distance, and non-consecutively matching striae are also associated with random forest scores close to 1. Observations in the test data that are classified incorrectly by the random forest tend to have feature values similar to observations in the training data with similar random forest scores.

LIME is applied to all test set observations using different tuning parameter values: 12 sampling methods (2- to 6-equally-spaced bins, 2- to 6- quantile-bins, kernel density estimation, and normal approximation) and 3 Gower exponents (0.5, 1, and 10). Thus, a total of $12 \times 3 = 36$ different applications of LIME are performed. We specify that each LIME explanation return 3 features, and feature selection is performed using the default option in LIME, which selects the features with the highest weights in a ridge regression model.

4.3 | LIME Assessment Visualizations

To get an overview of the LIME explanations from the 36 applications, we consider a feature heatmap (Figure 9). In addition to facets for simulation method and LIME feature importance, this plot includes a vertical facet for Gower power and a horizontal facet for whether the observation is a known match or non-match. This plot highlights several key features of the LIME explanations from the bullet matching dataset.

First, applications of 2-quantile-bins, 2-equal-bins, and somewhat for the density based simulations produce the same explanations for almost all cases and LIME tuning parameter values, suggesting these explanations are global and not local. Second, within a simulation method, the features selected by LIME for an observation do not appear to vary by the Gower power. However, the LIME explanations for an observation often vary across simulation methods. With the equal bins, there are vertical stripes that suggest a dependence of the LIME explanations on the number of bins. The vertical stripes are not as apparent with the quantile bins. Lastly, there are clear differences between the LIME explanations produced by the bin based simulation methods for the matches and non-matches. This suggests that different features are of importance in the random forest, depending on whether the observation corresponds to match or non-match.

To try to identify a set of LIME tuning parameter values with the most trustworthy set of explanations, an assessment metric plot is considered (Figure 10). The performance of a set of tuning parameter values often do not agree across metrics. For example, both density methods perform well according to average fidelity but poorly according to MSEE. All quantile bin methods perform well according to average R^2 but poorly based on average fidelity and MSEE. However, there is consistency in results across different Gower powers. The applications using a power of 0.5 perform the best or as well as the other powers across all simulation methods suggesting that the power that leads to a more local explanation is preferred. It is not obvious which set of tuning parameters is the best, but

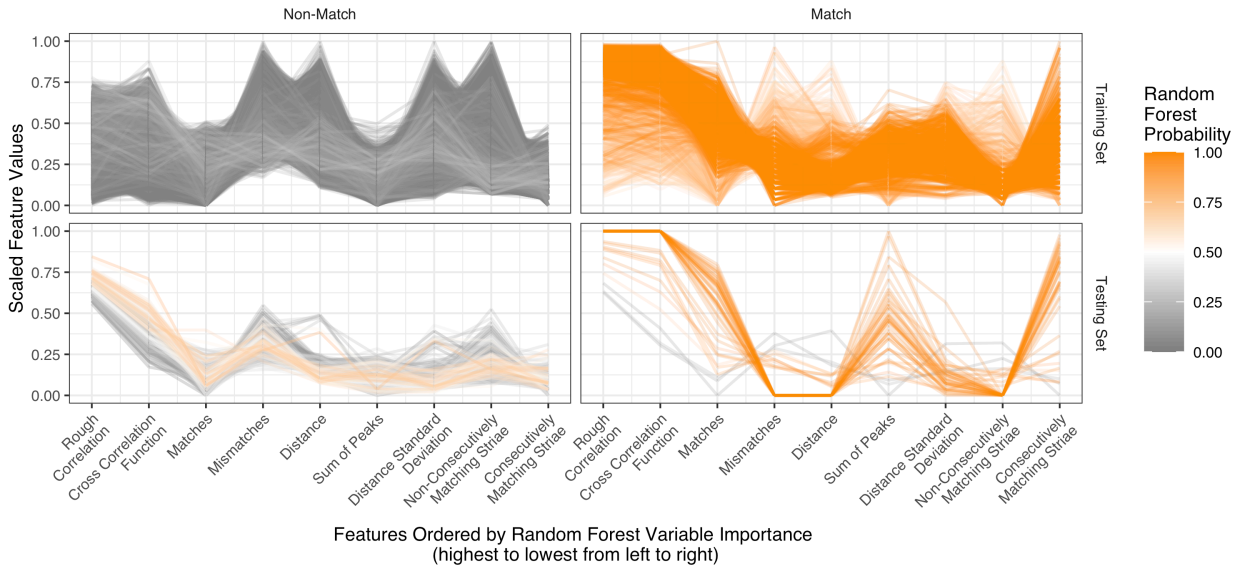

FIGURE 8 Parallel coordinate plots of the bullet matching random forest predictions. Each line corresponds to an observation, and the line color represents the associated random forest probability. There are clear relationships between the feature values and the random forest probabilities.

considering all three metrics, we might conclude that the 4-equal-bins with a Gower power of 0.5 performs better than the other tuning parameter values considered.

For a more detailed view of explanations based on different tuning parameter values, we consider explanation scatterplots for a prediction of interest, a known non-match (NM), in Figure 11 along with the explanation plots from the *lime* R package. Plots for 4-equal-bins and 4-quantile-bins are included to depict tuning parameter values with good (4-equal-bins) and mediocre (4-quantile-bins) performance based on the assessment metric plot. Additional examples of explanation scatterplots including explanations for a known match and explanations produced using the kernel density simulation method are included in the supporting information.

Figure 11 (top left) shows that when 4-quantile-bins are applied to case NM, LIME finds the values of mismatches, rough correlation, and distance are important and support a prediction of the class true. However, this explanation contradicts the random forest prediction which correctly assigns the observation a non-match. The explanation scatterplot (Figure 11 bottom left) clarifies the reason. Case NM falls on the boundary of regions for each of these three features where the random forest predictions transition from below 0.5 to above 0.5. However, the regions used when applying LIME from the 4-quantile-bins where case NM falls contain mostly observations with random forest predictions above 0.5, leading to the unfaithful explanation.

Figure 11 (top right) shows that when 4-equal-bins are used, the features LIME identifies as important are distance and mismatches, which support a match and matches, which supports a non-match. This explanation is more faithful having at least one feature supporting a non-match and has a better linear fit, but the most important feature identified by LIME still supports a match, contradicting the random forest. The explanation scatterplot (Figure 11 bottom right) again shows the features supporting a match arise from NM falling on the boundary of random forest prediction regions that are not well captured by the LIME bins.

Without an assessment of the LIME explanations, the default method would produce poor explanations such as the one shown in Figure 11 (left column). Our assessment here led to better explanations when the four equal bins are used. However, the explanation quality is still lacking from the one prediction considered. Additional investigations using explanations scatterplots would help to determine whether to trust this set of explanations, continue tuning the LIME parameters, or use another approach to provide better insight.

5 | DISCUSSION

This paper highlights that while an explainer model is meant to provide clarity, it actually adds another layer of complexity to predictive models by requiring yet another model that needs to be assessed. Without an assessment of the explainer model, LIME is a black-box procedure requiring blind trust in the explainer model. We suggest the use of visual diagnostics to counteract the black-box nature of LIME. We provide three

FIGURE 9 Feature heatmap of 36 LIME applications to the bullet comparison data test set. The vertical stripes of features selected indicate a dependence between the LIME explanations and tuning parameter values.

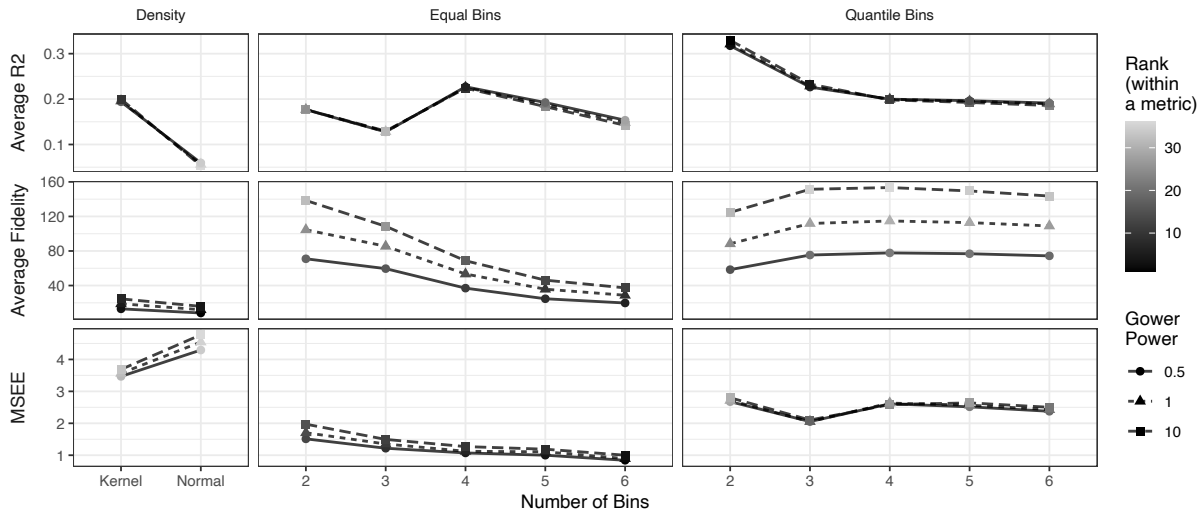

FIGURE 10 Assessment metric plot comparing LIME results from applications to the bullet comparison data test set. There are discrepancies in metric performances, but overall, the 4-equal-bin implementation with a Gower power of 0.5 appears to perform the best.

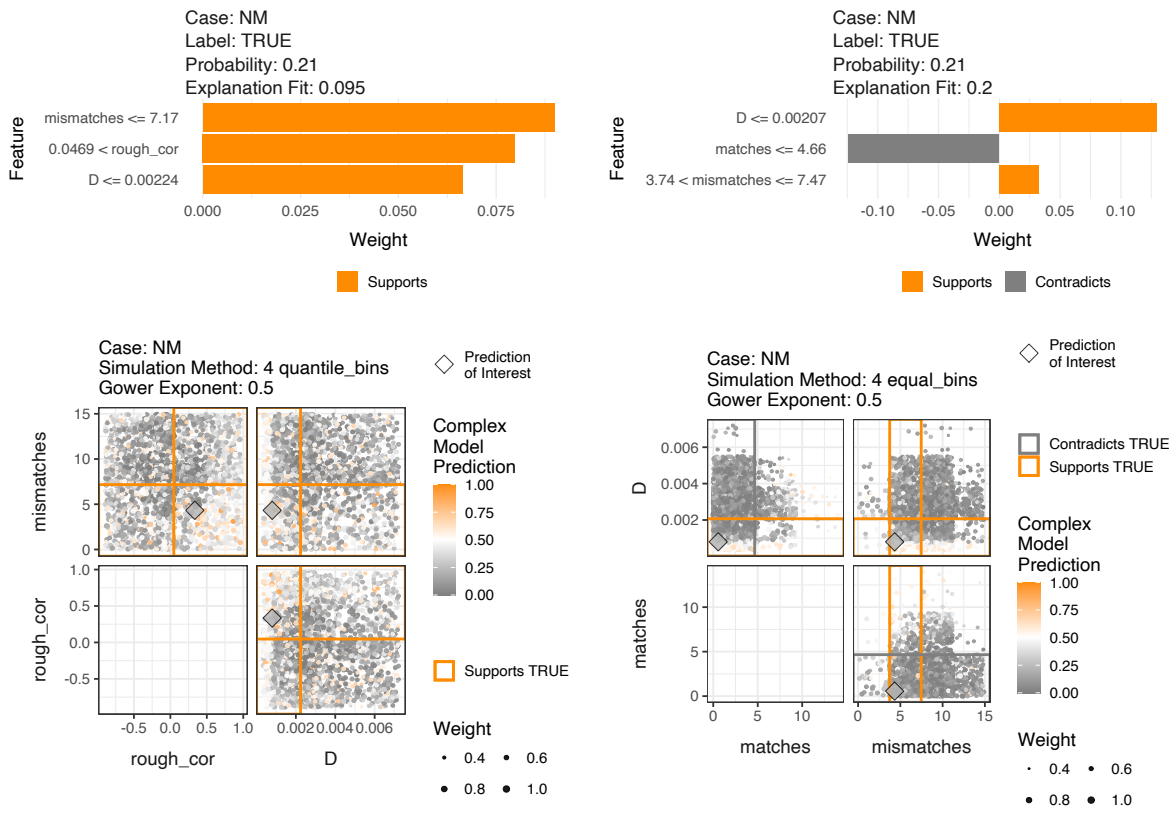

FIGURE 11 Plots of LIME explanations (first row) and explanation scatterplots (second row) for case NM in the bullet test data for two tuning parameter values: 4-quantile-bins (first column) and 4-equal-bins (second column).

diagnostic plots intended to assess whether LIME explanations meet the claims made by Ribeiro et al. [22]. The examples in this paper bring to light many scenarios in which the claims are not met or are difficult to assess. We reconsider each of the claims and how the visual diagnostics identify their failings.

As previously discussed, the **interpretability** of the LIME explanations is controllable by the complexity of the explainer model. For example, the number of bins selected for simulation controls the interpretability of the explanations. If too many bins are selected, the bin range that is reported in the LIME explanation will be too small to be meaningful in the context of the feature. An appropriate choice of the number of bins will keep the bin range meaningful. Thus, the claim of interpretability does not need to be assessed using the visualizations. However, diagnostic visualizations do present a different perspective on the meaning of interpretability.

Even though an explanation will be interpretable if the complexity of the explainer model is appropriately chosen, a lack of understanding of the explainer model could lead to an incorrect interpretation of the explanation. For example, the visualization of a LIME explanation from the *lime* R package [19] (Figure 3 left; Figure 11 top) provides an (over-)simplification of the explainer model that could lead to misinterpreted LIME explanations. Supplementing Pedersen and Benesty [19]'s compact visualization of the explanation with an explanation scatterplot that shows a more detailed visualization of the explainer model promotes a more complete understanding of the explanation (Figure 3 right; Figure 11 bottom).

Even with an explainer model that is interpreted correctly, the interpretation is worthless if the explainer model is not **faithful** to the complex model. Explanation scatterplots allow for a comparison of the explainer model to the complex model. The examples in this paper show cases where the explainer model oversimplifies the model and bins do not accurately capture the regions with similar random forest probabilities that contain the prediction of interest (Figure 3 right; Figure 11 bottom). Using fewer bins would clearly not help improve the faithfulness of the explainer model in these examples, and while an increase in the number of bins would lead to a finer resolution of the random forest classification boundaries, interpretability of the explainer model would quickly be lost. Perhaps this could be improved by allowing the bin creation to account for the relationships between the features and response variable or a different number of bins for each feature.

We also propose an assessment metric plot for the visual comparison of two faithfulness metrics: MSEE and average fidelity. Both metrics measure faithfulness, but MSEE only accounts for the prediction of interest while average fidelity accounts for all observations in the simulated data. As a result, the measures may disagree (Figures 5 and 10), and the user is left to determine which is preferable.

The metric comparison plot also includes a comparison of average R^2 values, which is a metric that can be used to assess the claim of **linearity**. Most of the average R^2 values in the examples from this paper are below 0.5 suggesting a poor linear fit of the explainer models. A poor linear fit of the explainer model is also seen with the residual plot shown in Figure B1 .

The final claim, **localness**, is addressed by feature heatmaps and explanation scatterplots. The vertical stripes seen in the feature heatmaps (Figures 4 and 9) suggest global explanations produced by LIME since the same explanation is returned for almost all observations regardless of feature values. The explanation scatterplots visualize the locality of the explanation using point size to indicate weight. However, the lack of definition of a local region makes it difficult to assess whether the weights capture an appropriately local region.

These diagnostic visualizations are a starting point that already highlight seemingly common issues with the LIME claims. Additions, such as visualizations of the step-by-step process of the creation of LIME explanations and interactivity would enhance the assessment process. For example, a diagnostic plot that provides a summary of multiple LIME explanations, such as the feature heatmap, could be clicked on to reveal more detailed figures, such as an explanation scatterplot, for a prediction of interest.

The largest limitation to the diagnostic visualizations is the dimensionality of the data shown, both in the number of dimensions or features as well as the number of observations. Fortunately, in the situation of LIME, both of these aspects are rather well controlled: LIME relies heavily on simulations to generate data scenarios that are close to the data observed but exhibit variability. Effects from overplotting should be relatively mild, because output from simulations is shown, which is expected to be (relatively) continuous such that overplotting only occurs for points with (relatively) similar values. In that respect, the diagnostics shown for the sine data and the bullet example are representative of what is expected. But in cases where overplotting does become problematic, the user could either simply reduce the size of the simulations or use some well-studied binning techniques in the visualizations, as discussed for example in Carr et al. [2] or Unwin et al. [27].

While it would be ideal if LIME could be used as a method to provide easily understandable explanations for black-box models as Ribeiro et al. [22] claim, there is still work to be done to achieve that reality. The examples using diagnostic plots to assess LIME in this paper show frequent instances when the claims about LIME are not met. We hope that our plots provide motivation to assess LIME explanations, to not blindly use the default settings (even if how to tune the parameters is not clear), and to encourage work improving LIME, so that it can be a lime and not a lemon.

ACKNOWLEDGMENTS

HH was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

Author contributions

Katherine Goode, Heike Hofmann: Conceptualization, Methodology, Investigation, Visualizations, Writing - Reviewing and Editing. **Katherine Goode**: Writing - Initial Draft, Software.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

SUPPORTING INFORMATION

Additional examples of the visual diagnostics and more info on the bullet matching data may be found in the Supporting Information section of the online version of the manuscript. The code and data associated with the manuscript are also included in the online Supporting Information section and in the following GitHub Repository: https://github.com/goodekat/LIME-diagnostics-paper.

References

[1] Alvarez-Melis, D. and T. S. Jaakkola, 2018: On the robustness of interpretability methods. [14].
URL https://arxiv.org/abs/1806.08049

[2] Carr, D. B., R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield, 1987: Scatterplot Matrix Techniques for Large N. *Journal of the American Statistical Association*, **82**, no. 398, 424–436, doi:10.1080/01621459.1987.10478445.

[3] Committee on Identifying the Needs of the Forensic Sciences, National Research Council, 2009: *Strengthening Forensic Science in the United States: A Path Forward*. https://www.ncjrs.gov/pdffiles1/nij/grants/228091.pdf.

[4] Emerson, J. W., W. A. Green, B. Schloerke, J. Crowley, D. Cook, H. Hofmann, and H. Wickham, 2013: The Generalized Pairs Plot. *Journal of Computational and Graphical Statistics*, **22**, no. 1, 79–91, doi:10.1080/10618600.2012.694762.

[5] Fisher, A., C. Rudin, and F. Dominici, 2019: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, **20**, no. 177, 1–81.
URL http://jmlr.org/papers/v20/18-760.html

[6] Friedman, J. H., 2001: Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, **29**, no. 5, 1189–1232, doi:10.1214/aos/1013203451.

[7] Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, 2018: Explaining Explanations: An Overview of Interpretability of Machine Learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, F. Bonchi and F. Provost, Eds., IEEE, 80–89.

[8] Goodman, B. and S. Flaxman, 2017: European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, **38**, no. 3, 50–57, doi:10.1609/aimag.v38i3.2741.
URL https://www.aaai.org/ojs/index.php/aimagazine/article/view/2741

[9] Gower, J. C., 1971: A general coefficient of similarity and some of its properties. *Biometrics*, **27**, 857–871, doi:10.2307/2528823.
URL https://www.jstor.org/stable/2528823

[10] Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, 2018: A Survey Of Methods For Explaining Black Box Models. *ACM Computing Surveys*, **51**, no. 5, doi:10.1145/3236009.

[11] Hamby, J. E., D. J. Brundage, and J. W. Thorpe, 2009: The Identification of Bullets Fired from 10 Consecutively Rifled 9mm Ruger Pistol Barrels: A Research Project Involving 507 Participants from 20 Countries. *AFTE Journal*, **41**, no. 2, 99–110.

[12] Hare, E., H. Hofmann, and A. Carriquiry, 2017: Automatic matching of bullet land impressions. *Annals of Applied Statistics*, **11**, no. 4, 2332–2356, doi:10.1214/17-AOAS1080.

[13] Jensen, M. S., R. Yao, W. N. Street, and D. J. Simons, 2011: Change blindness and inattentional blindness. *Wiley interdisciplinary reviews. Cognitive science*, **2**, no. 5, 529–46, doi:10.1002/wcs.130.

[14] Kim, B., K. R. Varshney, and A. Weller, Eds., 2018: *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, ICML.

[15] Laugel, T., X. Renard, M. Lesot, C. Marsala, and M. Detyniecki, 2018: Defining locality for surrogates in post-hoc interpretablity. [14].
URL http://arxiv.org/abs/1806.07498

[16] Liaw, A. and M. Wiener, 2002: Classification and Regression by randomForest. *R News*, **2**, no. 3, 18–22.
URL https://CRAN.R-project.org/doc/Rnews/

[17] Ming, Y., 2017: *A Survey on Visualization for Explainable Classifiers*. Ph.D. thesis, The Hong Kong University of Science and Technology.

[18] Molnar, C., 2019: *Interpretable Machine Learning*. lulu.com.
URL https://christophm.github.io/interpretable-ml-book/

[19] Pedersen, T. L. and M. Benesty, 2020: *lime: Local Interpretable Model-Agnostic Explanations*. R package, see also https://lime.data-imaginist.com.
URL https://github.com/thomasp85/lime

[20] President's Council of Advisors on Science and Technology, 2016: *Report on forensic science in criminal courts: Ensuring scientific validity of feature-comparison methods*. https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/PCAST/pcast_forensic_science_report_final.pdf.

[21] Ribeiro, M. T. and contributors, 2020: *lime*. https://github.com/marcotcr/lime, Python package.

[22] Ribeiro, M. T., S. Singh, and C. Guestrin, 2016: "why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 1135–1144.

[23] Simon, N., J. Friedman, T. Hastie, and R. Tibshirani, 2011: Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software*, **39**, no. 5, 1–13.
URL http://www.jstatsoft.org/v39/i05/

[24] Simonyan, K., A. Vedaldi, and A. Zisserman, 2014: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *Workshop Proceedings of the International Conference on Learning Representations (ICLR)*.
URL https://arxiv.org/abs/1312.6034

[25] Strumbelj, E. and I. Kononenko, 2014: Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, **41**, no. 3, 647–665, doi:10.1007/s10115-013-0679-x.

[26] Sweller, J., 2011: Chapter two - cognitive load theory. *Psychology of Learning and Motivation*, J. P. Mestre and B. H. Ross, Eds., Academic Press, volume 55, 37–76.
URL http://www.sciencedirect.com/science/article/pii/B9780123876911000028

[27] Unwin, A. R., M. Theus, and H. Hofmann, 2006: *Graphics of Large Datasets: Visualizing a Million*. Springer, New York.

[28] Urbanek, S., 2008: Visualizing Trees and Forests. *Handbook of Data Visualization*, C.-h. Chen, W. Härdle, and A. Unwin, Eds., Springer-Verlag, Berlin, Germany, volume 3, 243–266.
URL https://haralick.org/DV/Handbook_of_Data_Visualization.pdf

[29] Vanderplas, S., M. Nally, T. Klep, C. Cadevall, and H. Hofmann, 2020: Comparison of three similarity scores for bullet lea matching. *Forensic Science International*, **308**, 110167, doi:https://doi.org/10.1016/j.forsciint.2020.110167.
URL http://www.sciencedirect.com/science/article/pii/S0379073820300293

[30] Wickham, H., D. Cook, and H. Hofmann, 2015: Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **8**, no. 4, 203–225, doi:10.1002/sam.11271.

How to cite this article: Goode K., H. Hofmann, 2020, Visual Diagnostics of an Explainer Model – Tools for the Assessment of LIME Explanations, *Stat Anal Data Min: The ASA Data Sci Journal*, *volume, number and page*.

APPENDIX

A LIME TUNING PARAMETER OPTIONS

The following tuning parameters for the LIME algorithm are available in the *lime* R package [19].

- Data simulation methods:
  - Equally spaced bins: observations are uniformly sampled from equally spaced bins (number of bins may be specified)
  - Quantile bins: observations are uniformly sampled from quantile bins (number of bins may be specified)
  - Normal density approximation: observations are sampled from a normal distribution with mean and standard deviation computed from the corresponding feature
  - Kernel density approximation: observations are sampled from an kernel density approximation of the corresponding feature
- Number of observations to simulate
- Distance metric for determining proximity to the prediction of interest: Gower distance (where the power may be specified) or exponential kernel (where the kernel width may be specified)
- Number of features to return in an explanation
- Feature selection method for determining the features to return in an explanation: forward selection applied to a ridge regression, features with the largest magnitude coefficients in ridge regression, LASSO, classification/regression tree splits

The *lime* R package uses a ridge regression fit using the *glmnet* R package [23] as the explainer model. Note that the ridge regression penalty parameter (referred to as λ in *glmnet*) is usually treated as a tuning parameter, but the *lime* R package always sets λ to 2 divided by the number of observations in the simulated dataset.

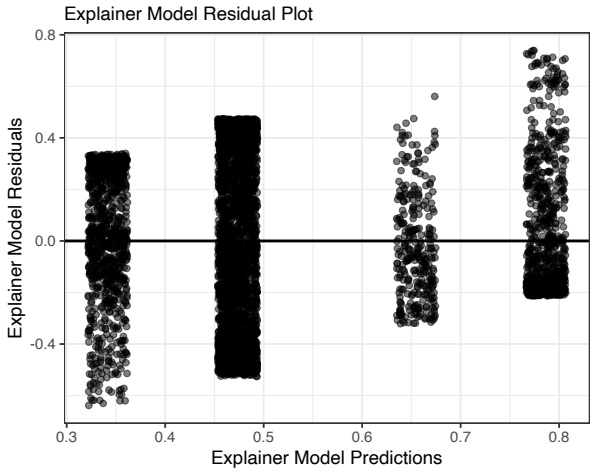

FIGURE B1 Residual plot of the explainer model associated with sine data prediction of interest from Section 3.1, which suggests a violation of the linearity assumption.

B EXPLAINER MODEL RESIDUAL PLOT

In order to assess the claim of linearity for the sine data prediction of interest discussed in Section 3.1, we use one of the most basic diagnostics in a statistician's tool box and draw a residual plot for the explainer model. This is shown in Figure B1 with the explainer model residuals on the y-axis and explainer model predictions on the x-axis. The points along the x-axis have been jittered to ease the effect of the over-plotted points. There is a clear increasing trend in the residuals as the explainer model predictions increase, which indicates a violation of the ridge regression linearity assumption.

C DETAILS ON ASSESSMENT METRICS

Suppose f is a complex model, and let $\mathbf{X}$ be a matrix of observed data with K features and E observations where x_e is an observed feature vector for observation e. Let $f(x_e)$ be the complex model prediction for observation e. It is of interest to explain the predictions made by f applied to X using LIME.

For x_e and a set of tuning parameter values t, let $\mathbf{X}'_{e,t}$ be the LIME simulated dataset with K features and S rows such that $x'_{e,t,s}$ is the feature vector for simulated data point s corresponding to explanation e and tuning parameter values t. Let $\mathbf{Z}'_{e,t}$ be the matrix of simulated data transformed to bin indicator variables (for bin based simulation methods) or standardization (for density based simulation methods) with K features and S observations such that $z'_{e,t,s}$ is the interpretability transformed feature vector for explanation e, tuning parameter values t, and simulated data point s. Note that $z_{e,t}$ will represent the transformed version of x_e.

Next, let ω_t represent a proximity distance metric corresponding to tuning parameter values t. Then $\omega_t\left(x_e, x'_{e,t,s}\right)$ is the weight assigned to $x'_{e,t,s}$, which is the proximity between x_e and $x'_{e,t,s}$. Allow $g_{e,t}$ to be the explainer model for an explanation e and tuning parameter values t. Thus, $g_{e,t}\left(z'_{e,s,t}\right)$ is the explainer model prediction for the interpretability transformed simulated data point s.

For a set of E explanations and a set of tuning parameter values t, we define the assessment metrics as follows:

Average R^2 is denoted as R^2_{ave} and computed as

$$R^2_{\text{ave}} = \frac{1}{E}\sum_{e=1}^{E} R^2_{e,t}$$

where $R^2_{e,t}$ is the R^2 value for $g_{e,t}$.

Average fidelity is denoted by $\mathcal{L}_{\text{ave}}$ and computed as

$$\begin{aligned}\mathcal{L}_{\text{ave}} &= \frac{1}{E}\sum_{e=1}^{E}\mathcal{L}(f,\, g_{e,t},\, \pi_t)\\ &= \frac{1}{E}\sum_{e=1}^{E}\sum_{s=1}^{S}\omega_t\left(x_e, x'_{e,t,s}\right)\left(f\left(x'_{e,t,s}\right) - g_{e,t}\left(z'_{e,t,s}\right)\right)^2.\end{aligned}$$

where $\mathcal{L}$ is the fidelity metric originally defined in Ribeiro et al. [22].

Mean squared explanation error is denoted by MSEE and computed as

$$MSEE = \frac{1}{E}\sum_{e=1}^{E}\left(f\left(x_e\right) - g_{e,t}\left(z_{e,t}\right)\right)^2.$$