Received XX XX 2020; | Revised XX XX XXXX; | Accepted XX XX XXXX

DOI: xxx/xxxx

RESEARCH ARTICLE

Visual Diagnostics of a Model Explainer – 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 model explainer 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 are often used for the latter purpose. These models have proven to perform well in a wide range of prediction problems, 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 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 [9] 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 variable. In contrast, 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, 11, 18, 19]. Some are specific to one type of model (e.g. [26] and [29]), and others are model-agnostic (e.g. [5] and [31]). In this paper, we focus on one specific model-agnostic method: LIME [23].

LIME (local interpretable model-agnostic explanations) is a method that uses a surrogate model to relate predictor variables to black-box model predictions (i.e. a model explainer) [23]. We distinguish between the terms of model explainer and explainer model: by *model explainer* we denote the method for explaining a complex model using a surrogate model, while the *explainer model*, or simply the *explainer*, is the surrogate model.

While some model explainers 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) [23]. Conceptually, LIME fits a simple (interpretable) model, the explainer model, meant to capture the behavior of the (complex) black-box model in a local region around a prediction of interest. The simple model then provides interpretable estimates for variables that most influenced the prediction made by the complex model.

Figure 1 provides a visualization of this conceptual understanding of LIME. The two plots show the predictions from a hypothetical black-box model plotted against the two hypothetical model predictor variables. The diamond shaped points represent the location of the prediction of interest. A Gower distance metric [10] is used to define locality such that the size of the points represent the inverse of the Gower distance raised to some value and indicate the proximity to the prediction of interest. In the example of Figure 1 , an exponent of 50 is used to emphasize a very local region around the prediction of interest. A ridge regression model weighted by the proximity values is used as the explainer model with the black-box predictions as the response variable and standardized versions of two features in the data as predictor variables. Standardized features allow direct comparisons of the model coefficients. The explainer model is depicted by the black lines in the figure.

The plot on the left of Figure 1 shows the relationship between the black-box predictions and Feature 1. Here, the explainer model is plotted with Feature 2 fixed at the observed value of the prediction of interest. The explainer model captures the relationship in an immediate neighborhood around the prediction of interest with a slope of 0.068. The plot on the right shows no global or local relationships between the black-box predictions and Feature 2. Here, the explainer model is plotted with Feature 1 set as the observed value of the prediction of interest, and it has an appropriately small slope of -0.001. The magnitude of the slope associated with Feature 1 is larger than the slope of Feature 2, which suggests that Feature 1 plays a more important role in the prediction made by the black-box model for the prediction of interest. This explanation agrees with a visual assessment of the relationships between the predictions and predictor variables.

The concept of LIME is relatively simple: use an interpretable model to approximate a complex model in a local region. However, a practical implementation of LIME is not straightforward, and research is being done to improve the procedure [16]. The current implementations of LIME [22] [20] 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. Thus, in order to achieve accurate explanations, the tuning parameter values selected need to be assessed.

We consider an example where the choice of exponent for the weights is crucial to the quality of the explanation. The plots in Figure 2 show the same data as Figure 1 , but the Gower distance metric exponent is decreased to 1 (the default exponent in the *lime* R package). This causes the observations that are further away from the prediction of interest to be given larger weights than before. In Figure 1 , the explainer model captures the relationship between the black-box predictions in an immediate neighborhood of the prediction of interest. This cannot be said of the explainer model in Figure 2 . In addition, the magnitude of the slope associated with Feature 2 (0.011) is larger than that of Feature 1 (0.005). Thus, this explainer model actually provides a misleading explanation that Feature 2 plays a more important role in the prediction of interest.

Several sources in the literature discuss the performance of LIME. One of the biggest difficulties with LIME is determining how to specify a local region [16] [19]. This is due to an unclear definition of a "local region" and how to apply LIME to achieve an appropriate local region as demonstrated by Figure 2 . Alvarez-Melis and Jaakkola [1] raise a concern pertaining to the robustness of explanations from LIME and other model explainers: they find that even small changes in predictor variables can lead to very different LIME explanations. Additionally, Ribeiro et al. [23] 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".

As a result of the various ways LIME can fail, it is important to assess LIME explanations. We suggest the use of visual diagnostics for assessment. In this paper, we lay out the set of claims about LIME made by Ribeiro et al. [23] and propose three visualizations for the assessment of these claims: (1) *explanation scatterplot*, (2) *feature heatmap*, (3) *assessment metric plot*. 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

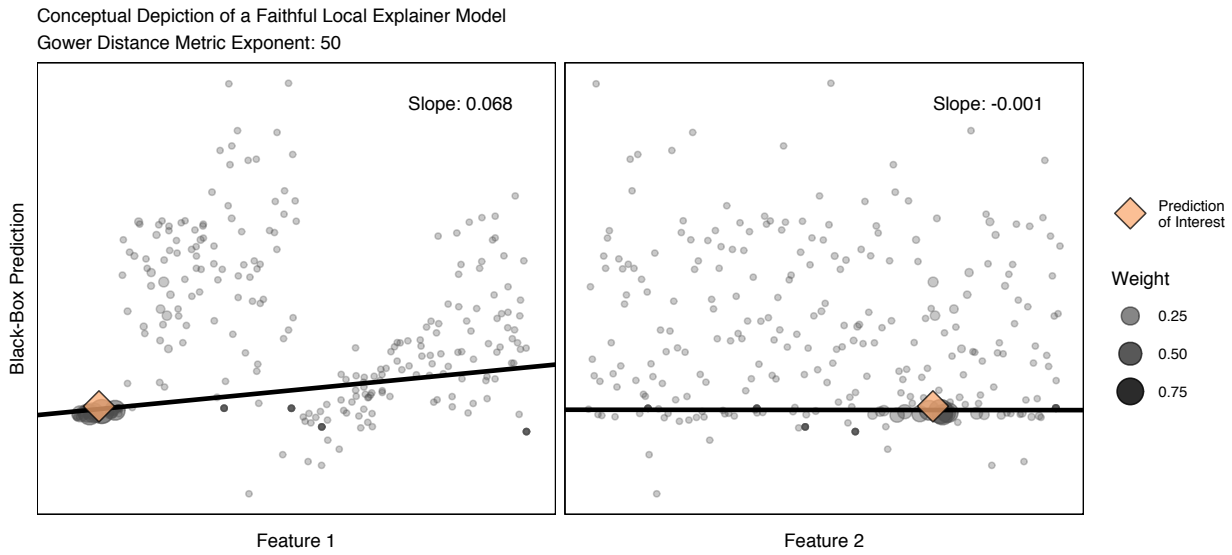

FIGURE 1 A conceptual depiction of a faithful LIME explainer model in the immediate neighborhood of a prediction of interest. The predictions from a hypothetical black-box model are plotted against the standardized values of the two hypothetical predictor variables. The diamond shaped points represent the location of a prediction of interest. The size and opacity of the circular points indicate the weight assigned based on the distance to the prediction of interest computed using the inverse of the Gower distance metric raised to the power of 50. The black lines represent a weighted ridge regression model used as an explainer model that reasonably captures the relationship between the black-box predictions and the features in a local region around the prediction of interest. That is, the explainer is faithful to the complex model and produces a reasonable explanation that Feature 1 plays a more important role in the prediction of interest than Feature 2 since the magnitude of the slope associated with Feature 1 is larger.

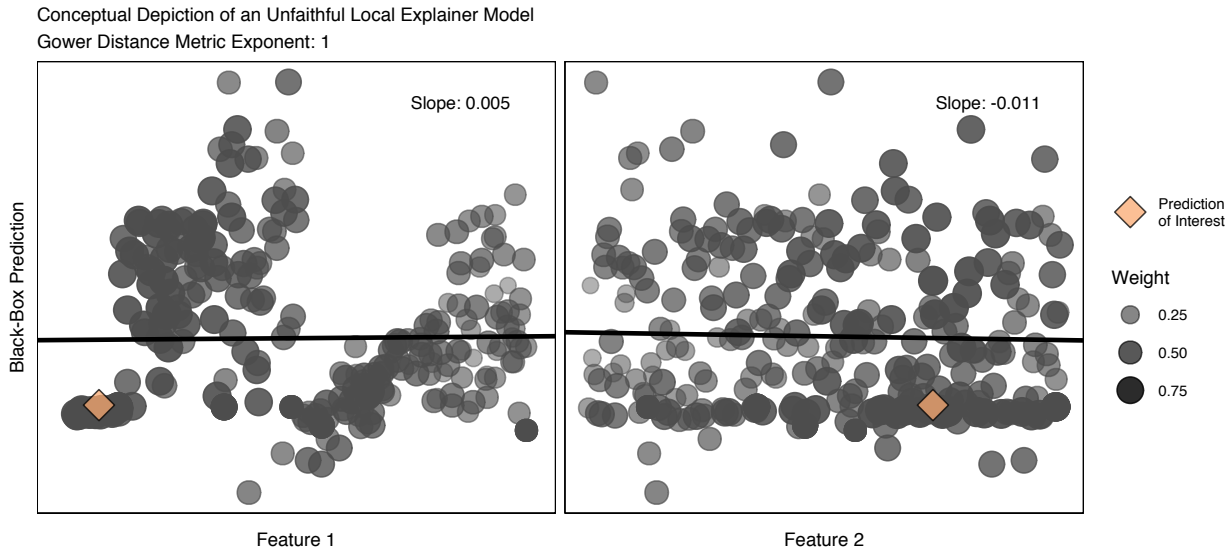

FIGURE 2 A conceptual depiction of an unfaithful local explainer model in the immediate neighborhood of a prediction of interest. A ridge regression model is fit to the same hypothetical black-box model predictions and predictor variables as in Figure 1 , but the weights are computed using a Gower exponent of 1. The explainer model is unfaithful to the complex model in any immediate neighborhood of the prediction of interest.

dichotomous response variable and continuous predictor variables. However, the proposed diagnostics may be extended to a wider range of situations.

The remainder of the paper is structured as follows. Section 2 provides background and claims made by Ribeiro et al. [23] 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. [23] brought about by the visualization examples in this paper that agree with Alvarez-Melis and Jaakkola [1], Laugel et al. [16], and Molnar [19].

All runs of LIME in this paper are executed in a forked version[1] of the R package *lime* (version 0.5.1) by Pedersen and Benesty [20]. 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 diagnostic plots included in this paper are created using the R package *limeaid* [8].

2 | BACKGROUND ON LIME

Ribeiro et al. [23] provide an implementation of LIME in a Python package [22]. An adaption of the Python package in R has been implemented and made available by Thomas Lin-Pedersen [20]. The discussion of parameter choices and implementation details of LIME in this paper are based on the R package.

The general form of the LIME algorithm can be divided into three steps [see also 16]:

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.

During the application of LIME, the user is asked to select 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. An overview of the options available for the tuning parameters is included in Appendix A.

In the original paper, Ribeiro et al. [23] 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.

3 | VISUAL DIAGNOSTICS FOR LIME

In this section, we introduce three visual diagnostic plots for the assessment of LIME. The plots focus on different levels of application of LIME (e.g. on explanation versus a set of explanations) to 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.

[1] https://github.com/goodekat/lime

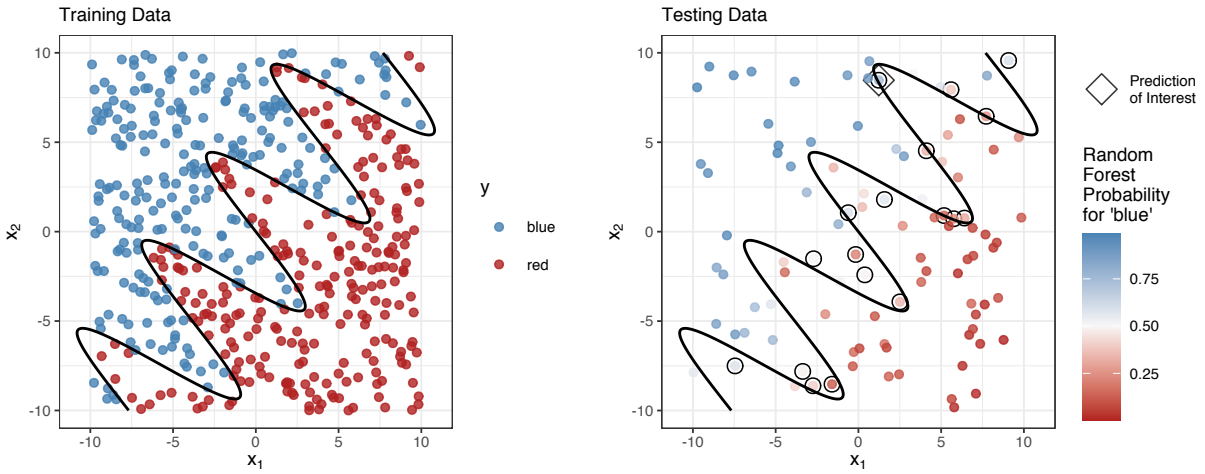

FIGURE 3 Plots of x_2 versus x_1 from the training (left) and testing (right) sets of the sine data introduced in Section 3. The true classification boundary is shown as the solid black line in both plots. The color of the training data represents the value of the observed response variable (y). The color of the testing data represents the random forest probability that an observation belongs to the category of blue. The 18 cases that are misclassified by the random forest are identified by black circles. The prediction of interest to explain is indicated by a diamond.

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 three features and one response variable. The features, x_1, x_2, and x_3, are randomly sampled from Unif(-10, 10), Unif(-10, 10), and N(0, 1) distributions, respectively. A binary response variable y is created using a rotated sine curve. In particular, let $x_1' = x_1\cos(\theta) - x_2\sin(\theta)$ and $x_2' = x_1\sin(\theta) + x_2\cos(\theta)$ where $\theta = -0.9$. Then y is defined as

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

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 divided into training and testing sets of 500 and 100 observations, respectively. A random forest is fit using the R package *randomForest* (version 4.6.14) [17] with the default settings. The model is applied to the test set to obtain predictions.

Figure 3 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 of the rotated sine function plotted 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. These are identified by open circles in Figure 3 .

From these scatterplots, we also 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 waves. However, the sine is a good-natured function that can be approximated well linearly in local regions.

We apply LIME using six sets of tuning parameter values to all observations in the sine data test set to observe variability across tuning parameter values. A quantile bin based simulation method (samples are simulated uniformly from a specified number of quantile bins) is used for five of the LIME applications 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 are the default method in *lime*. 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 the weights (Gower distance raised to an exponent of 1) are used for all applications. (See Appendix A for descriptions of the tuning parameters.)

For the presentation of the explanation scatterplot, we focus on the misclassified point with (x_1, x_2, x_3) coordinates of (1.23, 8.47, -0.99) indicated by a diamond in Figure 3 . 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 data.

A Visual Representation of a LIME Explanation

Before introducing the visual diagnostics, let us consider a commonly used visualization of a LIME explanation that is shown in Figure 4 (left). The plot is created using the *lime* R package and provides a visual representation of a LIME explanation for the prediction of interest indicated in Figure 3 . In particular, the explanation is obtained using 4-quantile-bins to simulate the data. In this scenario, 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 explainer model features.

The plot reports 0.74 as the random forest probability that the observation of interest belongs to category blue (denoted as the “label" in the plot). The lengths of the bars represent the coefficients from a ridge regression model (the explainer model) fit using the R package *glmnet* [25] associated with the indicator variables chosen via feature selection. The color of a bar denotes the sign of the coefficient and whether the feature "supports" or "contradicts" a random forest classification of ‘blue’. The "explanation fit" is the deviance ratio from *glmnet*. In other words, this is the R^2 value associated with the explainer model. In this case, it is 0.21 suggesting that the explainer model is not a good linear fit. However, it is commonly accepted that R^2 has limitations for assessing the quality of fit of a model [24] and should not be used as the only metric in a model assessment.

The explanation for the prediction of interest depicted in Figure 4 is interpreted as follows. A random forest classification of blue is supported by the prediction of interest having a value of x_2 that is greater than 4.834, but the prediction of interest having a value of x_1 that is greater than -0.302 and less than or equal to 4.844 provides support against a classification of blue. Since the weight associated with x_2 has a larger magnitude than the weight associated with x_1, LIME explains that x_2 plays a more important role in the random forest prediction. Additionally, consider 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, which agrees with the random forest probability of 0.744 for blue. Note that both models classify the observation incorrectly.

3.1 | Explanation Scatterplots

Based on the plot on the left of Figure 4 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.

The explanation scatterplot 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. These lines represent 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. Appendix B addresses the explanation scatterplot formats in other LIME simulation scenarios.

An explanation scatterplot corresponding to the LIME explanation depicted on the left in Figure 4 is shown on the right hand side of Figure 4 . 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 2-quantil-bin 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. Nevertheless, it is possible to say that the weights assigned to the simulated data do not capture the local region identified by the intersection of the quantile bins.

Now, consider the claim of faithfulness: It can be said that the majority of the points in the x_2 quantile 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. 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

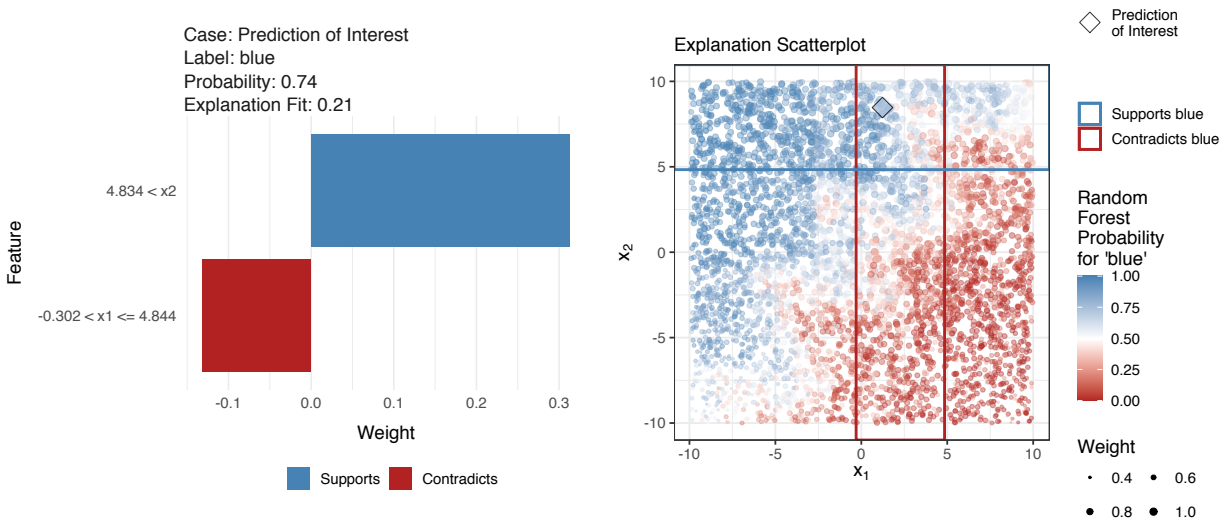

FIGURE 4 (left) Visualization from the *lime* R package of a LIME explanation for the sine data prediction of interest identified in Figure 3 . (right) *Explanation Scatterplot*. The points represent the simulated data colored by the random forest probabilities and sized by the assigned LIME weights. The prediction of interest is shown as the diamond shaped point. The explainer model indicator variables associated with x_1 and x_2 (quantile bins containing the prediction of interest) are depicted by the solid lines. The line colors represent the explainer model coefficient signs and indicate whether the feature supports a prediction of 'blue' (blue lines) or supports a prediction of 'red' (red lines). This figure shows that the quantile bins are not flexible enough to capture the relationship between the random forest predictions and the x_1 and x_2 values.

to check the linearity claim. See Appendix C for the residual plot associated with the explanation considered in Figure 4 , which shows a violation of the linearity claim.

This example explanation 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 [14] [27]. 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 [20] 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 Figures 1 and 2 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 *lime*. 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.

Two hypothetical examples of feature heatmaps are included in Figure 5 . The plots are created with the assumption that LIME is applied to select the top feature out of $p = 4$ features for $n = 10$ cases with $t = 5$ sets of tuning parameter values. Situation 1 is an example where the features selected are consistent across tuning parameter values within a case but vary across cases within a tuning parameter value. This is the ideal situation, because the LIME explanations do not depend on the

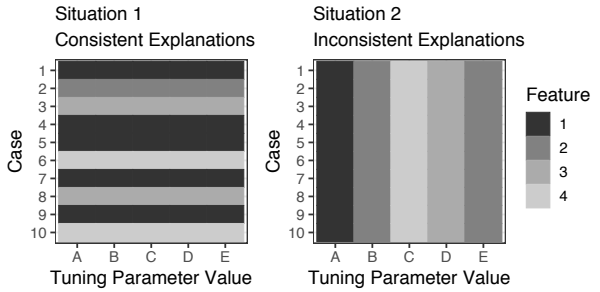

FIGURE 5 Hypothetical examples of feature heatmaps in two possible situations. The heatmaps show the top feature chosen for 10 cases across 5 different sets of tuning parameter values. The color of the cell indicates the feature chosen by LIME. Situation 1 is the ideal, because the explanations vary across cases but do not dependent on specific tuning parameter values.

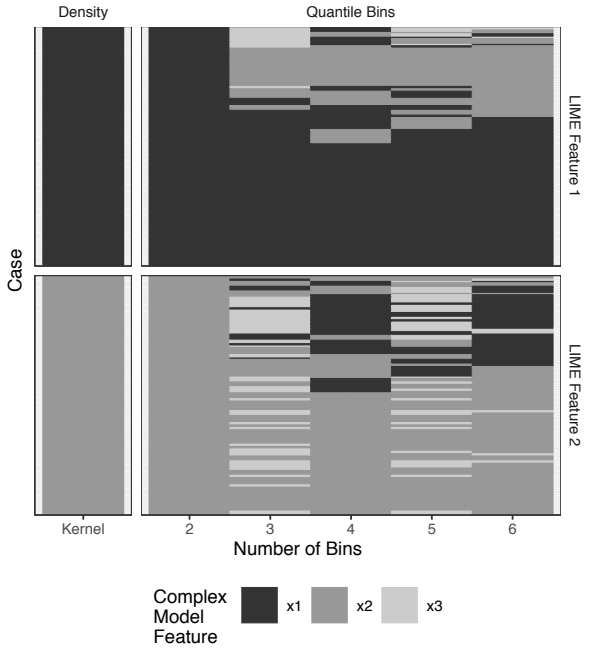

FIGURE 6 *Feature Heatmap*. An example feature heatmap of the explanations from the applications of LIME with different tuning parameter values to the sine data test set. The cases from the test set are plotted on the y-axis, and the tuning parameter values (simulation methods here) are included on the x-axis. The colors of the tiles indicate the feature selected by LIME for the corresponding case and tuning parameter value. The plot is faceted by the first and second most important features selected by LIME (top and bottom facets, respectively). The vertical facets separate the kernel density method from the quantile bin methods. The vertical striping indicates that the LIME explanations are not consistent across tuning parameter values.

tuning parameters but do depend on the location of the observation in the feature space. Situation 2 is an example where the selected features vary across tuning parameter values within a case but are consistent across cases within a tuning parameter value. This situation indicates that the features selected by LIME are dependent on the tuning parameters, and the explanations may not be local, because the same feature is chosen regardless of the case. In practice, it is expected that the plot will exhibit a combination of these two situations.

Figure 6 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 the indicator variable transformation are included since it is obvious that different features would be selected when the sizes of the bins change. This figure shows that for kernel density and 2-quantile-bins, LIME selects x_1 as the most important feature and x_2 as the second most important feature across all cases in the test set. These explanations are not local. 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 should not be the case. The pattern seen in the explanations for 3- to 6-quantile-bins may suggest local explanations, but the dependence on tuning parameters makes it unclear which set of explanations to use.

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 D.

- *Average* R^2: Assess the model fit and linearity claim by computing the average of the explainer model R^2 values (deviance ratios from the R package *glmnet*).

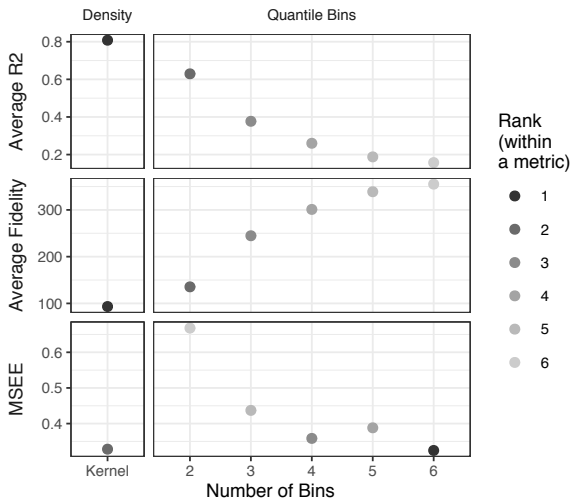

FIGURE 7 *Assessment metric plot*. The assessment metrics computed on the applications of LIME to the sine data test set are included in this example assessment metric plot. Each horizontal facet corresponds to one of three metrics: average R^2, average fidelity, or MSEE. The LIME tuning parameter values (simulation methods in this example) are plotted on the x-axis, and the metric values are shown on the y-axis. The points are colored by rank (within a metric) indicating best to worst (dark to light). The kernel density simulation methods performs well across all three metrics.

- *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. [23] (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 7 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 colors of the points represent the rank of 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 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, so it is not surprising that they would agree in this example. Another factor might be that MSEE only takes the prediction of interest into account and not the full simulated dataset. The contradiction between metrics makes it difficult to identify which simulation method to trust.

Recall that Figure 6 indicated that the kernel density method selected the same feature across all cases in the test set for both the first and second features. It appears that a global trend may be the best explanation for this example, which may be reasonable considering that we know that both x_1 and x_2 are the two features that should be the features used by the random forest to distinguish between response categories.

4 | APPLICATION TO BULLET MATCHING DATA

In this section, we provide a discussion of the application of 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 8). 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 [21].

In response, Hare et al. [13] 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. [13] obtain signatures of striations from two bullet lands (Figure 9). 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. [13] was trained on a set of scans of bullets from the James Hamby Consecutively Rifled Ruger Barrel Study [12], which included 10,384 land-to-land comparisons Hare et al. [13]. 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.

XXX minor change to emphasize that this is not a completly new model, just an expanded training set - otherwise you open yourself up to criticism of not using a validated model. The bulletxtrctr model was used in an external validation in Vanderplas et al. [30]. Here, we train a random forest model using the same nine comparison features as Hare et al. [13] to predict whether two lands come from the same source (i.e. are a match) in an expanded set of 83,028 land-to-land comparisons from two sets of bullets in the James Hamby Consecutively Rifled Ruger Barrel Study [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.

XXX mention overall goodness of fit or oob error rates? We first 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 facet) and testing data (bottom facets) (Figure 10). 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

FIGURE 8 (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).

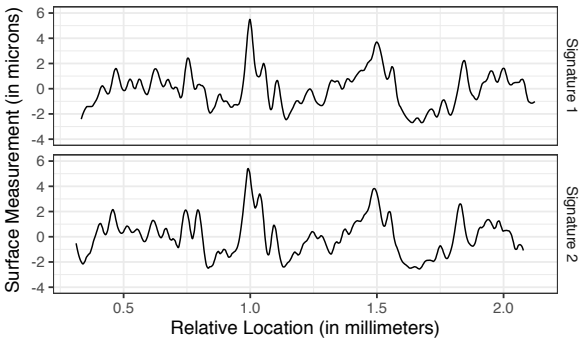

FIGURE 9 Example bullet signatures. The bullet signatures correspond to the same land and therefore have very similar patterns. The Hare et al. [13] random forest is fit using various features that measure the similarity between two such signatures.

similar random forest scores. For example, the observations in the test set known to be a match but have a random forest score below 0.5 have relatively low values of rough correlation, cross correlation function, and matches and relatively large values of mismatches, distance, and non-consecutively matching striae compared to observations with a random forest probability close to 1.

LIME is applied to all test set observations using different tuning parameter values: 12 sampling methods (2 to 6 equally

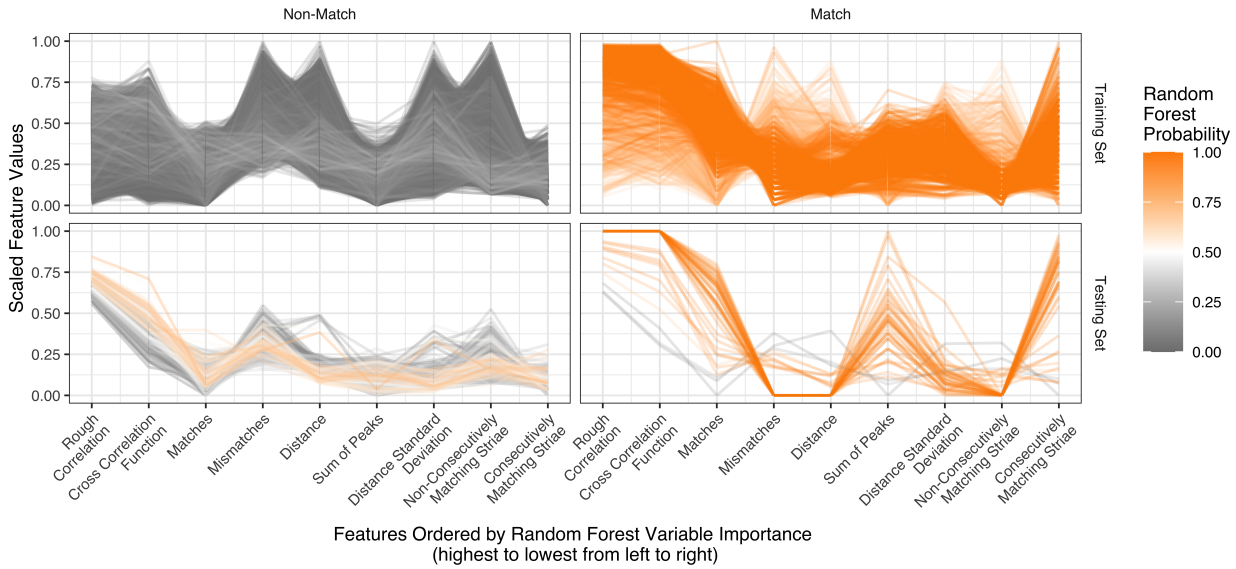

FIGURE 10 Parallel coordinate plots of the Hare et al. [13] random forest predictions from the training (top facet) and testing (bottom facet) data. The observations are separated by known matching (right column) and non-matching (left column) signatures. The y-axis shows the standardized feature values, and the x-axis shows the features used to fit the random forest (ordered by random forest impurity based feature importance). 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.

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 by selecting the features with the highest weights in a ridge regression model (default option).

4.3 | LIME Assessment Visualizations

To get an overview of the LIME explanations from the 36 applications, we consider a feature heatmap (Figure 11). 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-bin simulations produce the same explanations for almost all cases and LIME tuning parameter values. This suggests that the LIME 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 explanation for an observation often varies 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 12). 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 global explanation is preferred by LIME. It is not apparent which set of tuning parameters is the best. 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.

To provide a more detailed view of explanations obtained using different tuning parameter values, we take a closer look at explanations for two observations of interest, which will be referred to as case M (a known match) and case NM (a known non-match), using explanation scatterplots. Figure 13 includes the visual representation plots of explanations (from

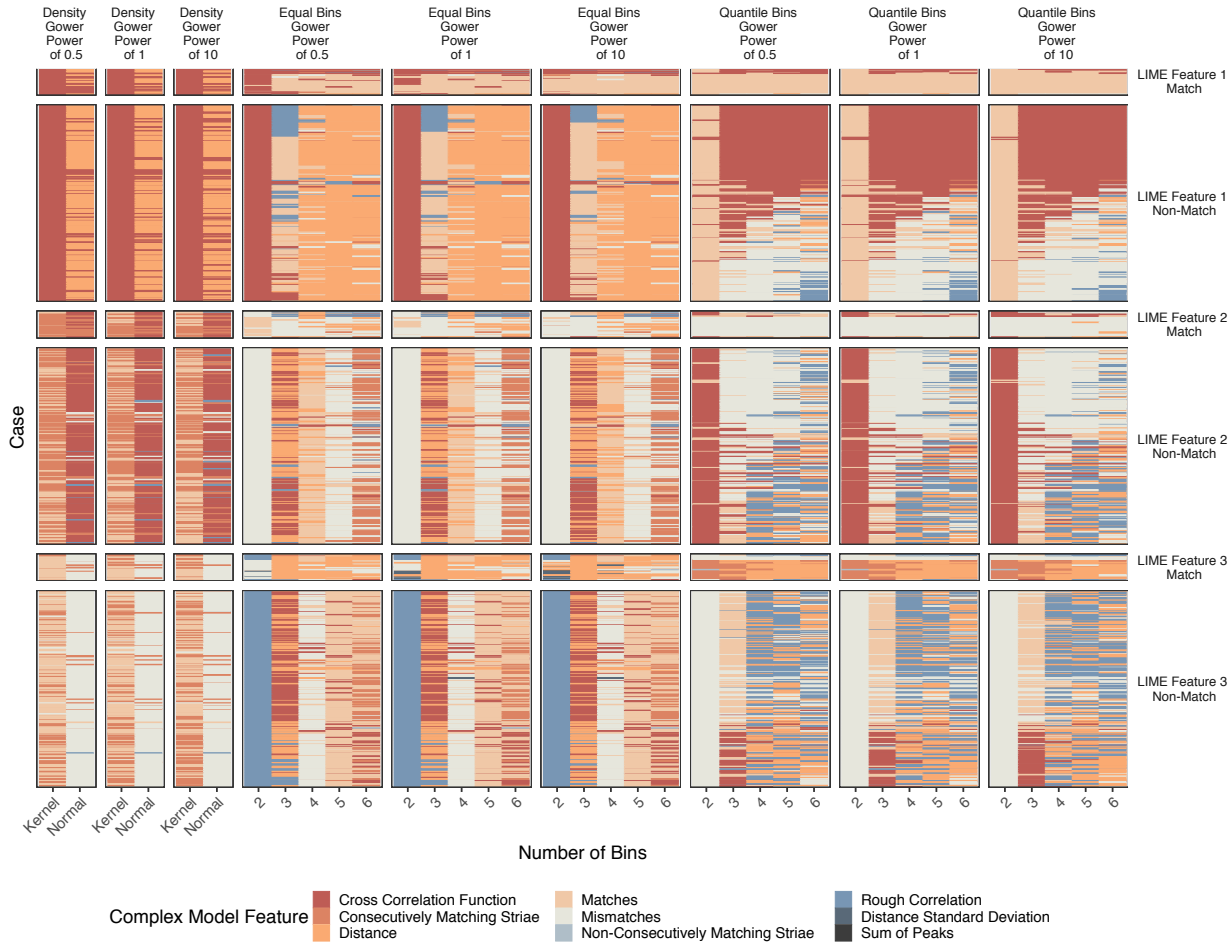

FIGURE 11 Feature heatmap of 36 LIME applications to the bullet comparison data test set. In addition to faceting the results by simulation method and LIME feature selection order, facets for the Gower power and whether the observation is a match or non-match are included. The vertical stripes of features selected indicate a dependence between the LIME explanations and tuning parameter values.

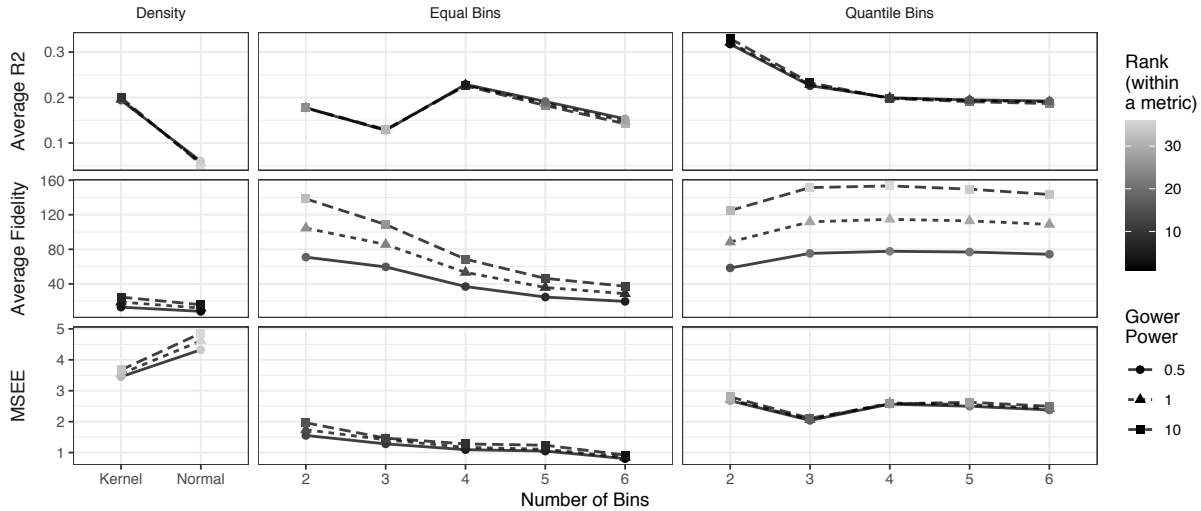

FIGURE 12 Assessment metric plot for the applications of LIME on the bullet comparison data test set. There are discrepancies in performance across metrics for a set of tuning parameter values, but based on the overall performance of all three metrics, 4-equal-bins with a Gower power of 0.5 appears to perform the best.

lime) and explanation scatterplots for cases M and NM. 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. Plots for the kernel density simulation method are included in Appendix B (Figure B2).

First, focus on the explanation for case NM with 4-quantile-bins. The explanation plot from *lime* shows that observed values with rough correlation (rough_cor) greater than 0.05 and mismatches value less than or equal to 7.17 support a random forest prediction in favor of a match, and the scatter plots show that many of the simulated values with rough correlation greater than 0.05 and mismatches less than 7.17 have random forest predictions greater than 0.5. Additionally, the LIME explanation indicates that a value of cross correlation function (ccf) greater than 0.28 and less than or equal to 0.35 supports a non-match, and the scatter plots shows that the simulated values within this region have mostly random forest predictions less than 0.5. However, the random forest assigns a probability of 0.19 to case NM, which is not explained by the LIME explanation. Instead, consider that a better explanation based on the scatterplots would be that the random forest assigned a prediction probably close to 0 since the observation has a rough correlation less than 0.5, mismatches greater than 4, and cross correlation function 0.6.

The other explanation scatterplots can be interpreted in a similar manner. Overall, the plots identify that the 4-quantile-bin explanations are not faithful to the random forest model predictions. The 4-equal-bin explanations are more faithful to the random forest, but the regions captured by the bins do not align well with the regions containing similar probabilities produced by the random forest.

Without applying LIME with multiple tuning parameter values to the bullet test data or viewing diagnostic plots of the LIME explanations, it may be very possible to formulate reasons why the LIME explanations make sense. However, the sequence of plots in this section (Figures 11 , 12 , and 13) suggest that we should be cautious to trust any of these LIME explanations. It appears that either LIME needs to be further tuned to provide trustworthy and good explanations, or a different approach may 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 of its own requiring blind trust in the explainer model. We suggest the use of visual diagnostics to counteract the black-box nature of LIME and provide three diagnostic plots.

The visualizations are intended to provide insight on how LIME works, assess the ability of the explainer model to capture the complex predictive model, and compare LIME explanations produced by different tuning parameter values. While the visualizations accomplish these tasks, they also expose examples of the failings of LIME. To address the discovered failings of LIME, we reconsider each of the claims about the performance of LIME made by Ribeiro et al. [23] in light of the insights gained from the diagnostic visualizations.

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 can control 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 as long as the complexity of the explainer model is appropriately chosen, a lack of understanding of the process used to create the explanation could lead to an incorrect interpretation of the explanation. For example, the visualization of a LIME explanation available from the *lime* R package [20] (shown in Figures 4 and 13) is a major simplification of the explainer model, which could lead to under-interpreted or misinterpreted LIME explanation. Supplementing Pedersen and Benesty [20]'s compact visualization of the explanation with an explanation scatterplot that shows a more detailed visualization of the explainer model (such as Figures 4 and 13) promotes a more complete understanding of the explanation. Even with an explainer model that is interpreted correctly, the interpretation is worthless if the explainer model is not **faithful** to the complex model. This claim can be assessed using the diagnostic plots suggested in this paper. Many of the visualizations in this paper highlight problems with the faithfulness of the explainer models. The 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 bins do not accurately capture the regions with similar random forest probabilities that contain the prediction of interest and oversimplify the model (Figures 4 and 13). 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

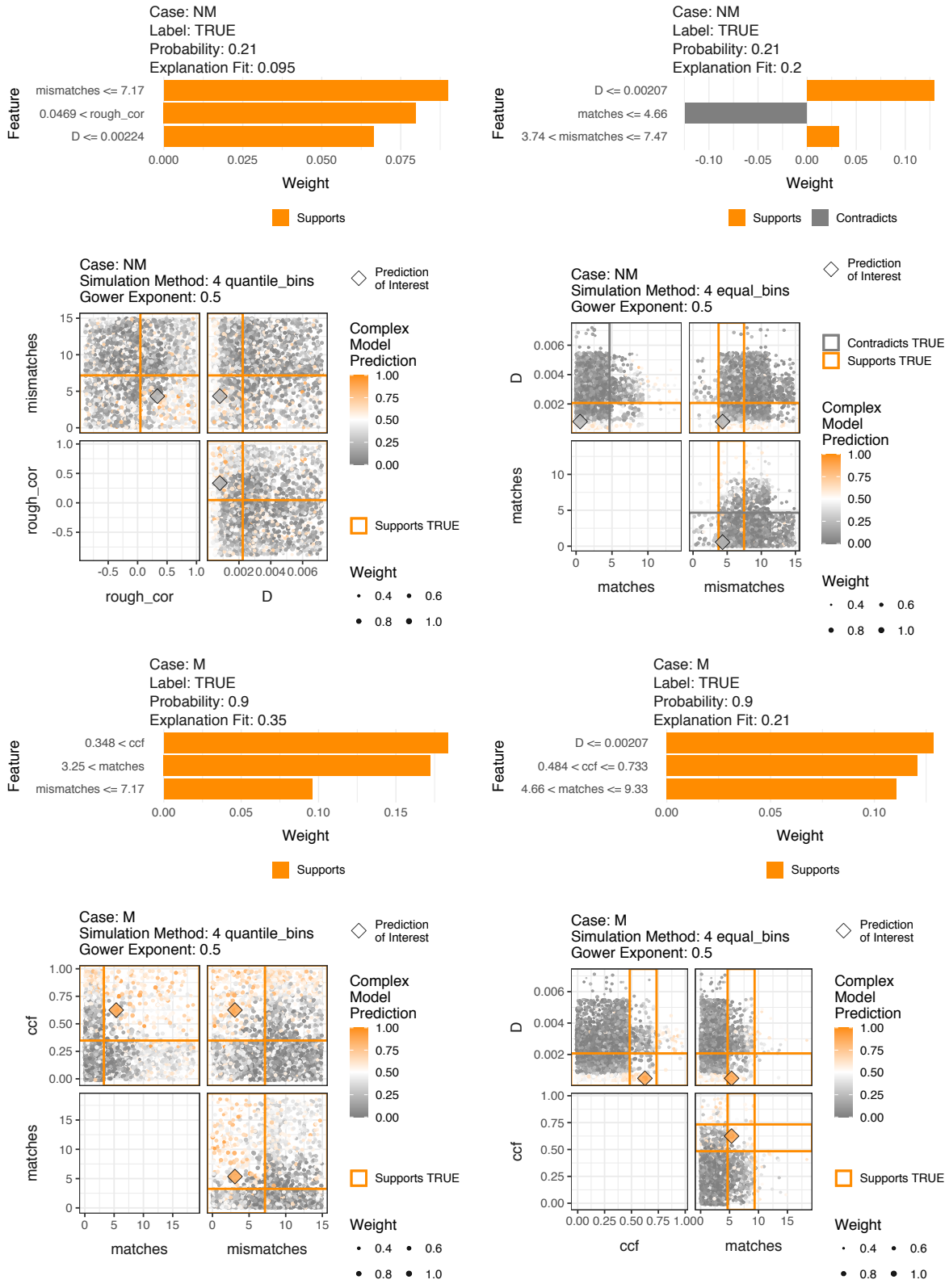

FIGURE 13 Plots of LIME explanations (first and third rows) and explanation scatterplots (second and fourth rows) for cases M and NM in the bullet test data for two tuning parameter values: 4-quantile-bins (first column) and 4-equal-bins (second column). The plots provide insights into the LIME explanations and allow for the assessment of the explanation quality.

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.

In addition to assessing faithfulness by visually comparing the complex and explainer model, we propose a visual comparison of two faithfulness metrics (MSEE and average fidelity). The examples of faithfulness metric comparisons in this paper (Figures 7 and 12) both produced conflicting results. This makes it difficult to decide on a recommendation of a set of tuning parameter values that produce the explanations with the most faithful explainer model.

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. The poor linear fit of the explainer model is also seen with the residual plot (Figure C3).

The final claim, **localness**, is addressed by the feature heatmap and metric comparison plots. As stated, the feature heatmap reveals that the density simulation methods in the sine data example results in global explanations where the same features are repeatedly chosen across all (or almost all) observations in the set of explanations. This finding agrees with that of Laugel et al. [16] who found LIME produced global explanations with the normal approximation simulation method. For the bin based simulation methods in the bullet data example, the feature heatmap showed that the features chosen for the explanations varied between the two classification categories (match versus non-match). This is an interesting finding that suggests that different features can play a role in the predictions of observations in different response categories. Furthermore, while the metric comparison plot in the bullet example does not provide agreement between metrics on a best bin based method, all metrics agree that a Gower power of 0.5 for computing the model weights associated with distance of a simulated data point from the prediction of interest is the best. This suggests that a less local explanation provides a better explanation of the performance of the random forest.

Some of the visualizations in the paper generalize easily to any application of LIME such as the feature heatmap and metric plot. Other plots such as the visualizations of the LIME procedure would require extensions such as the use of scatterplot matrices to compare explanations with more than two features. The addition of interactivity to the diagnostic plots would provide additional enhancement of the assessment process. For example, a diagnostic plot that provides a summary of multiple LIME explanations, such as the feature heatmap, could be displayed and clicked on to reveal more detailed figures associated with individual predictions of interest, such as an explanation scatterplot.

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 exhibits 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. [28].

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. [23] claim, that dream is not yet a reality. The examples using diagnostic plots to assess LIME in this paper show frequent issues with LIME. We hope that our plots provide motivation to assess LIME explanations, to not blindly use the default settings (even if it is not clear how to tune the parameters), and to encourage work on 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

The code used to produce the manuscript and the data from examples in Section 4 are available in the following GitHub Repository: https://github.com/goodekat/LIME-diagnostics-paper. ~~Additional information about the bullet matching data are available at XXX.~~

References

[1] Alvarez-Melis, D. and T. S. Jaakkola, 2018: On the robustness of interpretability methods. [15].
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] Goode, K., 2020: *limeaid: Diagnose LIME Explanations*. R package version 0.0.1.

[9] 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

[10] 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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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

[17] 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/

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

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

[20] 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

[21] 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.

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

[23] 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.

[24] Sapra, R., 2014: Using R^2 with caution. *Current Medicine Research and Practice*, **4**, no. 3, 130–134, doi:10.1016/j.cmrp.2014.06.002.

[25] 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/

[26] 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

[27] 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

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

[29] 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

[30] 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

[31] Štrumbelj, 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.

How to cite this article: Goode K., H. Hofmann, 2020, Visual Diagnostics of a Model Explainer – 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 [20].

- 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

B EXPLANATION SCATTERPLOTS UNDER OTHER SIMULATION SCENARIOS

Section 3.1 introduces explanation scatterplots under the default simulation method in the *lime* R package: 4-quantile-bins. The structure of an explanation scatterplot remains the same if any bin based simulation method is used, i.e., any number of quantile or equally spaced bins. However, if the kernel density or normal approximation simulation methods are used, the format of the explanation scatterplot changes. In the density based simulation method scenarios, LIME uses the standardized versions of the predictor variables to fit the explainer model. Thus, the explainer model needs to be represented differently in the explanation scatterplot.

When the kernel density or normal approximation simulation methods are applied, the explanation scatterplot depicts the complex model by plotting the complex model predictions versus a feature selected in LIME the explanation from the simulated data. The explainer model is included as a line on the figure where all features excluding the one plotted on the x-axis are set to the observed values of the prediction of interest. An explanation scatterplot is created for each feature included in the LIME explanation. As with the bin based simulation method, the size of the points represent the weight assigned by LIME.

Figure B1 provides example explanation scatterplots for each feature in the default LIME explanation obtained using the kernel density simulation method for the sine data prediction of interest. Figure B2 includes explanation scatterplots for the explanations generated using kernel density simulation for the bullet example cases M and NM discussed in Section 4.3.

C 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 C3 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 in the visualization. There is a clear increasing trend in the residuals as the explainer model predictions increase. This is a clear violation of the linearity assumption with the ridge regression model.

D 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

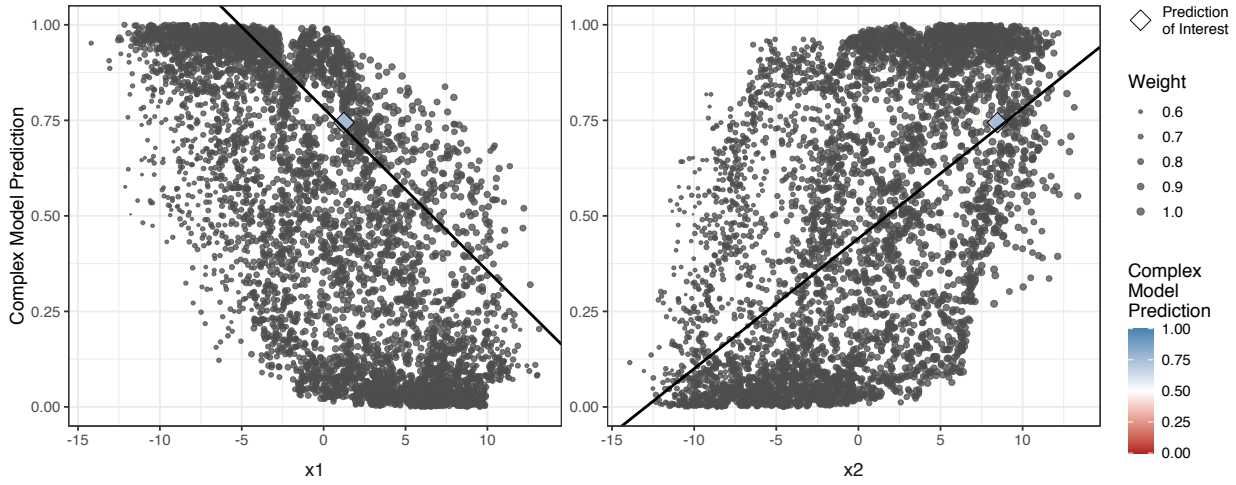

FIGURE B1 Explanation scatterplots for the sine data prediction of interest with the kernel density simulation method.

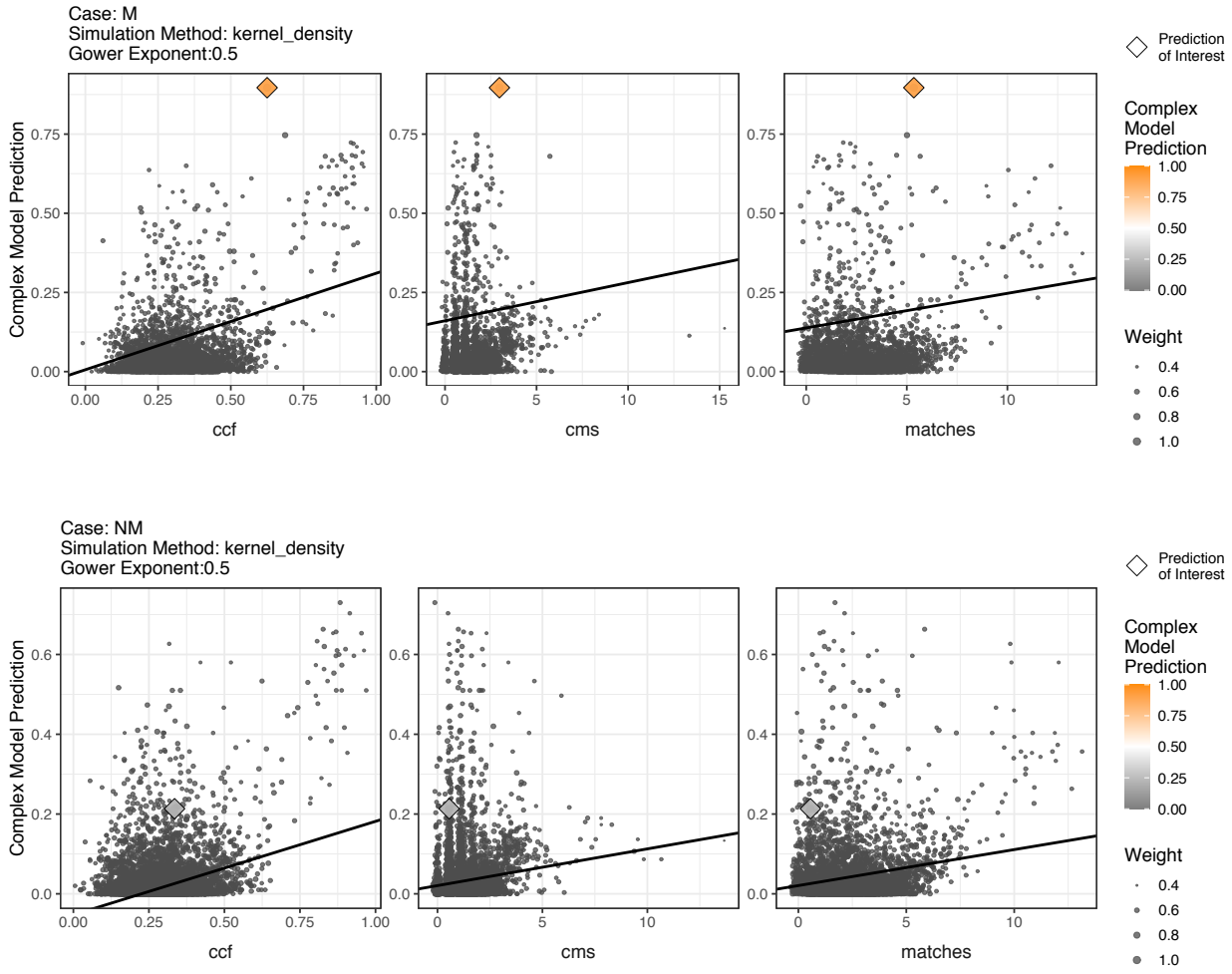

FIGURE B2 Explanation scatterplots for LIME explanations using kernel density simulation for the cases M and NM of the bullet comparison.

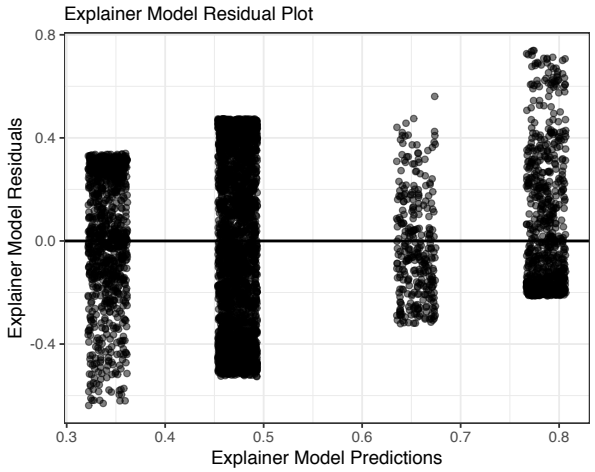

FIGURE C3 Residual plot of the explainer model associated with sine data prediction of interest from Section 3.1. The residuals are plotted against the predicted values. The points are jittered in the x-direction to alleviate the overplotting of points. There is an upward trend in the residuals as the explainer model predictions increase, which suggests a violation of the linearity assumption.

$x'_{e,t,s}$ 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. [23].

Mean squared explanation error is denoted by MSEE and computed as

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