Received 26 April 2016; | Revised 6 June 2016; | Accepted 6 June 2016

DOI: xxx/xxxx

ARTICLE TYPE

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
*Corresponding author. Email: kgoode@iastate.edu

Present Address
This is sample for present address text this is sample for present address text

Summary

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) [18]. LIME uses a surrogate model to explain the relationship between predictor variables from a black-box model and the black-box model predictions 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 scatterplot, (2) assessment metric plot, and (3) feature heatmap. We apply the visual diagnostics to a forensics bullet matching dataset to show examples where LIME explanations depend on the tuning parameters and the explainer model oversimplifies the black-box model. Our examples raise concerns about claims made about LIME [18] that are similar to other criticism found in the literature (Alvarez-Melis and Jaakkola [1], Laugel et al. [12], and Molnar [15]).

KEYWORDS:
LIME, black-box models, interpretability, diagnostics

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 assessment and understanding of how the model produces predictions. The lack of the ability to assess and understand a model makes it difficult to trust the model, especially in areas areas with high stakes decisions such as health care and forensics science. The increased use of machine learning models in applications and the introduction of the General Data Protection Regulation (GDPR) in 2018 [5] has resulted in a dramatic increase in research in an area known as explainable machine learning, which focuses on developing ways to explain output from machine learning algorithms.

Throughout this paper, we make a distinction 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, *explanability* is defined 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 [4].

Numerous methods have been proposed to provide explanations for black-box model predictions [8, 14, 15]. Some are specific to one type of model (e.g. [2] and [10]), and others are

model-agnostic (e.g. [7] and [21]). In this paper, we focus on the model-agnostic method of LIME [18].

1.1 | Conceptual Description of LIME

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) [18]. 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 are focused 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) [18]. 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 predictor variables used to fit the hypothetical model. The diamond shaped points represent a prediction of interest. A Gower distance metric [6] is used to define locality: the size of the points represent the inverse of the Gower distance as a reflection of the proximity to the prediction of interest. Here, an exponent of 50 is used to emphasize interest in a very local region around the point 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 Features 1 and 2 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 Feature 2 for the prediction of interest. The explainer model captures the relationship in the immediate neighborhood around the prediction of interest with a slope of 0.068. The plot on the right shows that there is no global or local relationship between the black-box predictions and Feature 2. Here, the explainer model is plotted with Feature 1 set to be the observed value of Feature 1 for 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 point of interest. This explanation agrees with a visual assessment of the relationships between the predictions and predictor variables.

1.2 | Motivation for Diagnosing LIME

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 there is research is being done to improve the procedure [12]. The current implementations of LIME[1] [17] offer various tuning parameters the user specifies when applying LIME (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 parameters selected need to be assessed.

We consider an example where the choice of exponent used to specify 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 was 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 the 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 (0.011) associated with Feature 2 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 predicting the observation 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 [12] [15]. 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. [18] acknowledge that if a linear models is used as the explainer, LIME relies on a linear approximation of the explainer model to the complex model

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

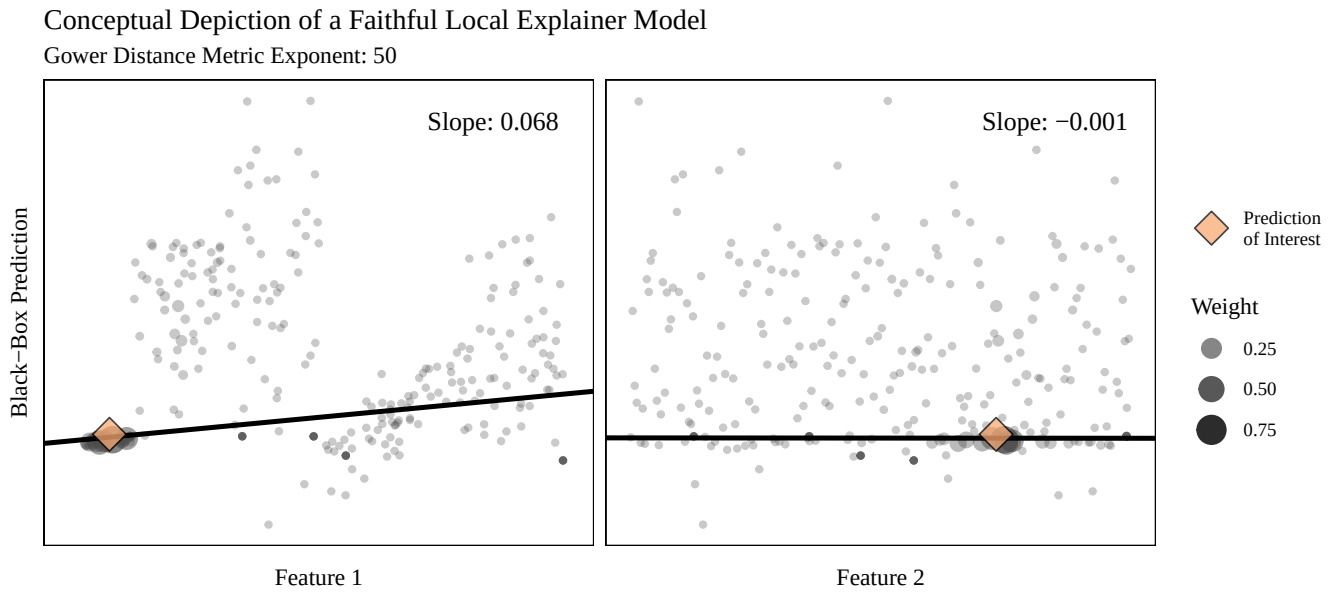

FIGURE 1 A conceptual depiction of a faithful LIME explainer model in the immediate neighborhood of a prediction of interest. The predictions from a hypothetical black-box model are plotted against the standardized values of the two predictor variables used to fit the hypothetical model. The diamond 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 point of interest computed using the inverse of the Gower distance metric raised to the power of 50. The black lines are 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 point of interest. That is, it 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.

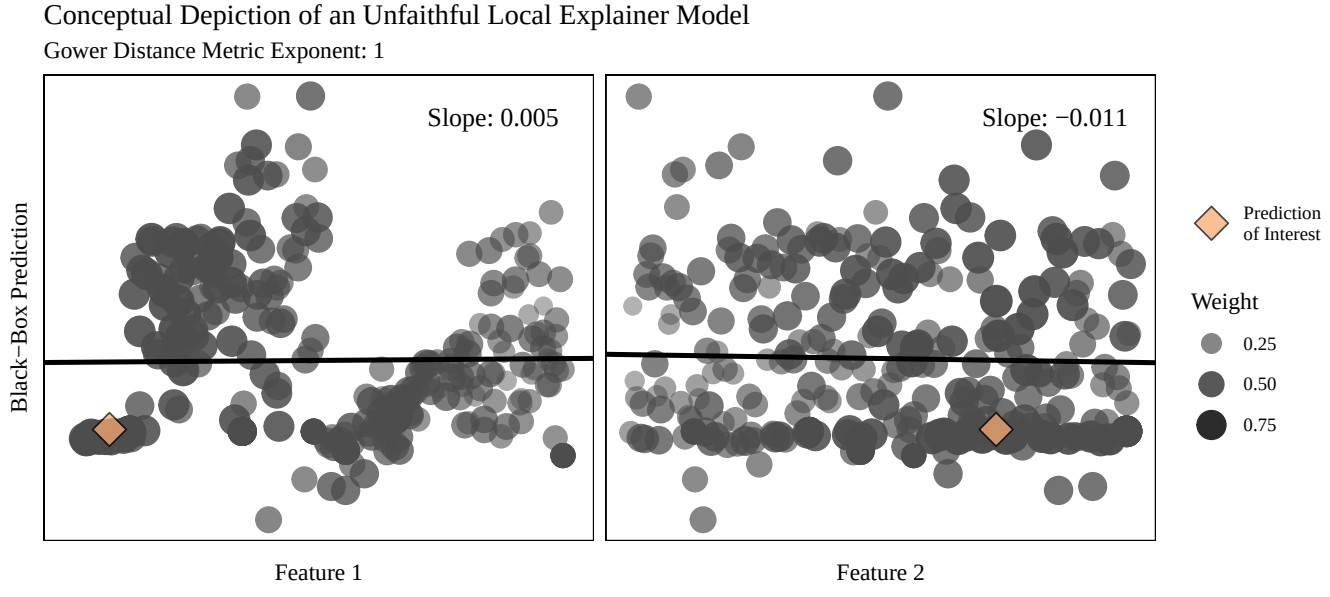

FIGURE 2 A conceptual depiction of an unfaithful local explainer model in the immediate neighborhood of a prediction of interest. A ridge regression model is fit to the same black-box model predictions and predictor variables as in Figure 1 , but the weights are computed using a Gower exponent of 1.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

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 explanations 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. [18] and propose three visualizations for the assessment of these claims: (1) *explanation scatterplot*, (2) *feature heatmap*, (3) *assessment metric plot*. While LIME has been implemented for image, tabular, and text data, we will only focus on tabular data. For additional simplicity, we will only discuss classification prediction models with a dichotomous response variable and continuous predictor variables. However, the proposed diagnostics may be extended to a wider range of situations.

1.3 | Structure of Paper

The remainder of the paper is structured as follows. Section 2 provides background and claims made by Ribeiro et al. [18] 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 model fit to a forensics bullet matching dataset. Section 5 concludes with a discussion on extension and limitations of the diagnostic plots and concerns about LIME in regards to the claims made by Ribeiro et al. [18] brought about by the visualization examples in this paper that agree with Alvarez-Melis and Jaakkola [1], Laugel et al. [12], and Molnar [15].

2 | BACKGROUND ON LIME

LIME was introduced in 2016 by Ribeiro et al. [18]. The original authors provide an implementation of LIME in a Python package[2]. An adaption of the Python package in R is by Thomas Lin-Pedersen [17]. In this paper, any details provided about the implementation of LIME are based on the R package.

As described by Laugel et al. [12], the general form of the LIME algorithm can be divided into three steps:

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 observations 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 must 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. [18] 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 – 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:

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

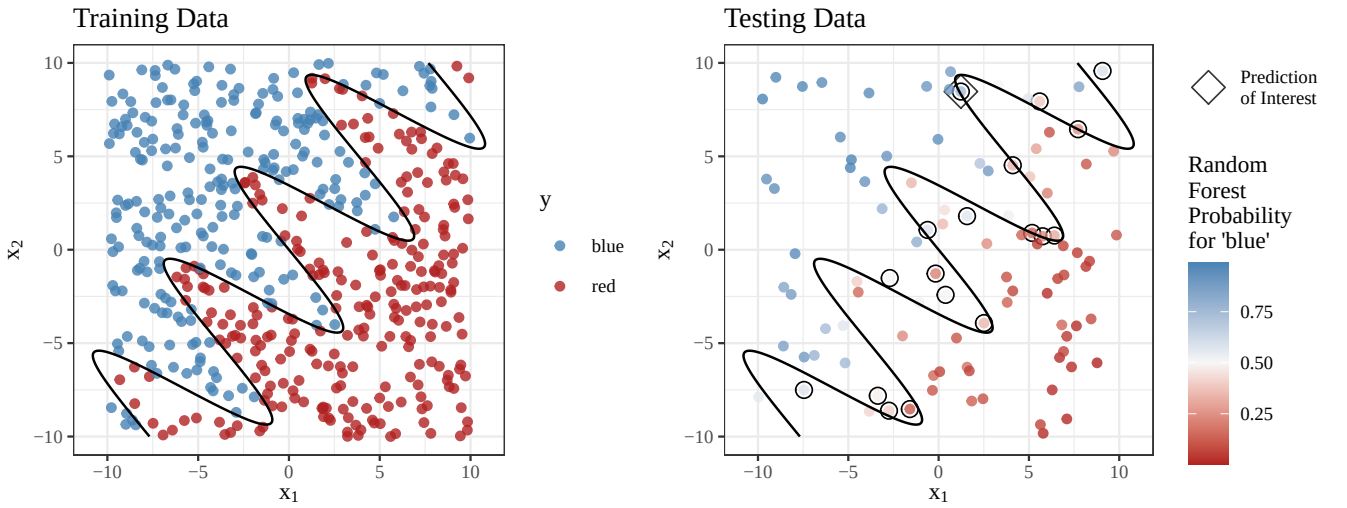

FIGURE 3 Plots of x_2 versus x_1 from the training (left) and testing (right) sets of the sine data introduced in Section 3. The true classification boundary is shown as the solid black line in both plots. The color of the training data points represents the value of the observed response variable (y). The color of the testing data points represents the random forest probability that an observation belongs to the category of blue ($\hat{y}$). 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.

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 parameters.

3. *Assessment Metric Plot* (Section 3.3): Comparison of performance metrics for LIME across applications of LIME with different tuning parameters.

The sine data

To demonstrate the visualizations, 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 a training set of 500 observations and a testing set of 100 observations. A random forest model is fit using the R package *randomForest* (version 4.6.14) [13] 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 the prediction probabilities from the random forest model ($\hat{y}$). The random forest model misclassifies 18 points on the classification boundary. These are identified by circles in Figure 3 .

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 . For the introduction of the other three plots, we use all observations in the sine datatest data as points of interest.

We perform six applications of LIME with different tuning parameters to the random forest predictions from all observations in the sine data test set. To do this, we use the R package *limeaid* [3] (0.0.1), which applies a forked version[4] of the development version of the R package *lime* (0.5.1) [17]. The forked version allows us to export internal values from *lime*. The tuning parameters are as follows. For each application, we specify that LIME returns the top two features in the explanation, because we expect x_1 and x_2 to be the important features. Note that the complexity of an explanation goes hand in hand with

[3] Available on GitHub at https://github.com/goodekat/limeaid
[4] https://github.com/goodekat/lime

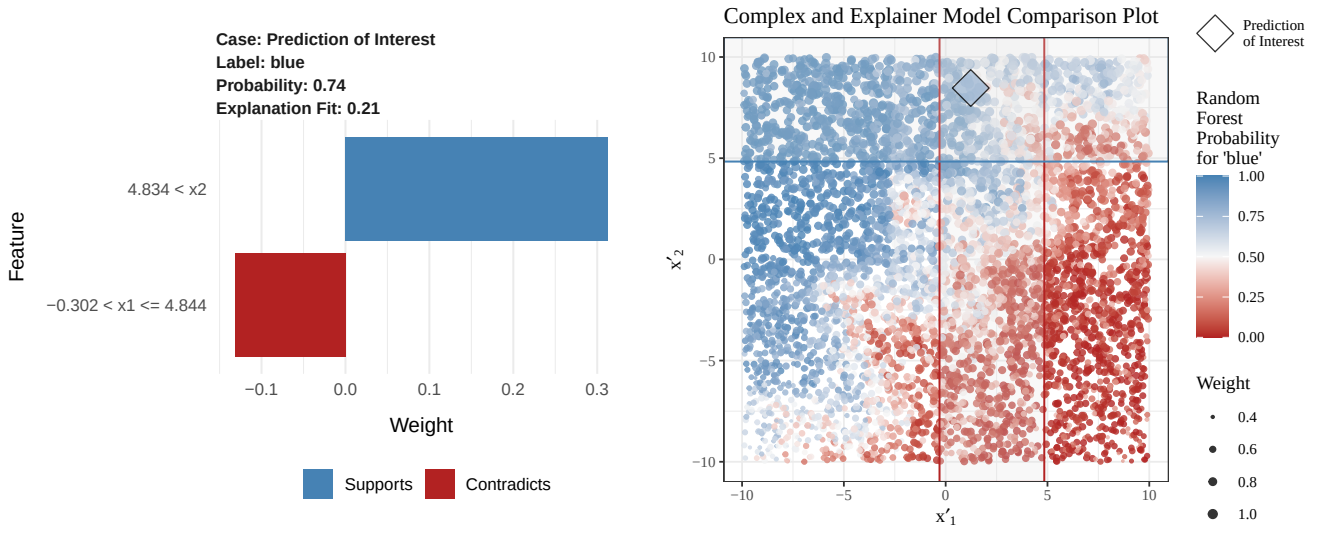

FIGURE 4 (left) Visualization from the *lime* R package of the LIME explanation for the sine dataindividual prediction of interest. The bars represent the explainer model coefficients. (right) *Explanation Scatterplot* The points represent the simulated data colored by the random forest probabilities 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 color of the lines indicates the explainer model explanation for whether the corresponding feature value 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.

the number of features selected to be included in the explanation. Setting the number of features to a small number will make the explanation simpler but not necessarily more correct. A quantile bin based simulation method is used for five of the implementations with the number of bins varying from 2 to 6 by application. The sixth implementation of LIME uses a kernel density simulation method. 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.

3.1 | Explanation Scatterplots

The *explanation scatterplot* is a visual diagnostic for assessing the LIME claims 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 default method in the *lime* R package in which the data are simulated uniformly from four quantile bins. 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, and the indicator variables are used as the explainer model features. The explanation scatterplot achieves a juxtaposition of the complex and explainer models by plotting the LIME simulated data colored by the complex predictions overlaid by lines representing boundaries of the indicator variables used to fit the explainer model. The size of the points represents the weight assigned by LIME, and the color of the lines denote whether LIME indicates that a feature supports or contradicts a class prediction. A scatterplot is created for all pairwise combinations of features included in the LIME explanation. Appendix B addresses the explanation scatterplot formats in other LIME simulation scenarios.

To demonstrate an explanation scatterplot, we focus on the misclassified point indicated in Figure 3 by the diamond as the prediction of interest. The left plot in Figure 4 shows the visualization of the explanation produced by *lime* for the prediction of interest. The figure shows that the random forest returned a probability of 0.744 that the observation of interest belongs to category blue (denoted as the "label" in the figure). The colored bars indicate the features chosen by LIME and whether they support or contradict a random forest classification of blue. Note that the features are represented as regions in the figure. These are the indicator variable regions created by LIME.

The explanation for the prediction of interest depicted in Figure 4 can be 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 the x_1 and x_2 locations of the prediction of interest provide more support for a random forest classification of blue over red. This explanation agrees with the random forest probability of 0.744 for blue, even though both classify the observation incorrectly.

Figure 4 also includes an "explanation fit" value. This value corresponds to the deviance ratio from the R package *glmnet* [20] for a ridge regression model, or 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 accessing the quality of fit of a model [19], and it should not be the only metric used for assessment of a model. Based on this metric or this plot, it is not yet possible to make an informed assessment of the trustworthiness of the explanation.

The plot on the right of Figure 4 is the complex and explainer model comparison plot for the prediction of interest from the sine data. This figure shows the relationship between the random forest model and the LIME explainer model. The random forest model is represented by the simulated values of x_2' versus x_1' colored by the random forest predictions. The explainer model is represented by the solid horizontal and vertical line identify the boundaries of the quantiles that contain the prediction of interest. The color of the lines represents whether the coefficient of the corresponding indicator variable in the explainer model supports a random forest prediction of 'blue' (blue lines) or not (red lines). Additionally, the size of the simulated data points represents the proximity weight used to fit the ridge regression explainer model. The prediction of interest is represented by the diamond shaped point.

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 remain relatively high outside of the intersection of the two quantile bins suggesting that the LIME explanation is highly influenced by points outside of the bin containing the prediction of interest. While it is still difficult to say whether or not the claim of localness has been violated, it is possible to say that the weights assigned to the simulated data do not agree with the local region assigned by the intersection of the quantile bins. Next, let us consider the claim of faithfulness.

Within the intersection of the x_1' and x_2' quantile bins, there appears to be more random forest probabilities above 0.5. This indicates why the magnitude of the coefficient associated with x_2 is larger than that of x_1. However, the pattern of the random forest predictions in this plot suggest that a better explanations for this prediction exist. One example of a better explanation would be to say that the prediction of interest received a random forest prediction greater than 0.5 since the observation of interest falls in the region where x_1 is less than 2.5 and x_2 is greater 4.5. Another more localized explanation would be to say that the region around the prediction of interest shows that observations with $-0.3 < x_1 <= 2.5$ and $4.8 < x_2 <= 8.75$ are all assigned probabilities greater than 0.5. The procedure implemented by LIME is not flexible enough to capture either of these explanations.

The explanation scatterplot shows issues with a definition of locality and an oversimplification of the random forest model. LIME does provide an explanation for the prediction of interest, which makes sense based on the quantile bins used. However, the visualization of the complex and explainer model comparison plot solidifies the understanding that better explanations for the prediction of interest exist.

The sine data explanation only includes two features. In situations were more than two features are included in a LIME explanation, the explanation scatterplots can be extended to a generalized scatterplot matrix (reference) that includes all pairwise combinations of features. Generalized scatterplot matrices (and scatterplot matrices in general) fail to provide helpful information when the number of features becomes large (reference). It is common in machine learning prediction problems for there to be a large number of features, and therefore, a generalized scatterplot matrix of explanation scatterplots would be ineffective. However, in the application of LIME, the user selects the number of features to return in the explanation (as previously mentioned). Pedersen and Benesty [17] encourages users to select less than 10 features to include in the explanation (need to check how to cite vignettes: https://cran.r-project.org/web/packages/lime/vignettes/Understanding_lime.html). As a result, specifying a small number of features to include a LIME explanation leads to the feasibility of using generalized scatterplot matrices of explanation scatterplots to assess individual LIME explanations.

3.2 | Feature Heatmap

Explanations produced by LIME are likely to be affected by the choice of tuning parameters. An example of this was 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, there are no recommendations or procedures provided for how to determine which method to use besides for the default settings in *lime* (confirm this). In order to compare the explanations produced by LIME using different tuning parameters, we view an overview of all the explanations in the *feature heatmap* visual diagnostic plot.

The feature heatmap uses colors to identify the features selected by LIME for sets of predictions and tuning parameters faceted by the position of importance assigned by LIME. That is, for LIME applied for t sets of tuning parameters to n cases to select the f top features, create f heatmaps (one for each of the positions of importance) with the cases on the y-axis, the tuning parameters on the x-axis, and the cells colored by the feature chosen for the corresponding case and tuning parameter. This plot is used assess the locality of the explanations and to compare results from different input options.

Two hypothetical examples of feature heatmaps are included in Figure 5 . The plots were created with the assumption that LIME was applied to select the top feature out of $p = 8$ features for $n = 10$ cases with $t = 5$ sets of tuning parameters. If LIME produces local explanations, the top features selected by LIME will vary for predictions located in different regions of the feature space (unless a feature or several features are the deterministic features throughout the entire feature space). Additionally, if the tuning parameters did not affect the feature selected by LIME, the selected feature would be the same across all tuning parameters for a prediction.

The plots in Figure 5 depict two different situations that may arise when applying LIME. Situation 1 shows an example where the features selected across implementation methods within a case are consistent, and the features selected by vary across cases. This is the ideal situation, because regardless which implementation method is used, the LIME explanations do not vary, and the explanations appear to be local since the features vary between cases. Situation 2 shows an example where the features selected across implementation methods within a case vary, but the features selected within an implementation method across cases are the same. This situation ~~is not ideal, because it~~ indicates that the features selected by LIME are dependent on the implementation method used, and the explanations are not 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.

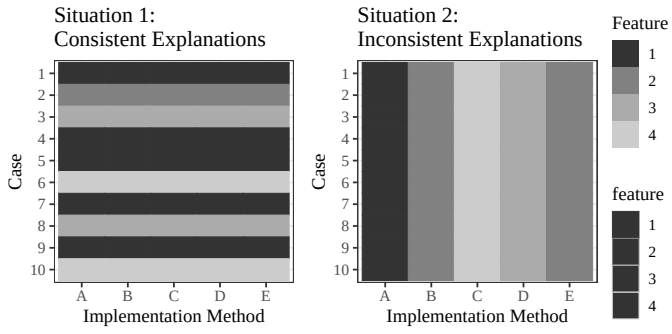

FIGURE 5 Two hypothetical examples of feature heatmaps showing the top feature chosen for 10 cases across 5 different implementations of LIME (sets of tuning parameters). The color of the cell indicates the feature chosen by LIME. The plots represent two possible situations that may arise when applying LIME to a set of predictions.

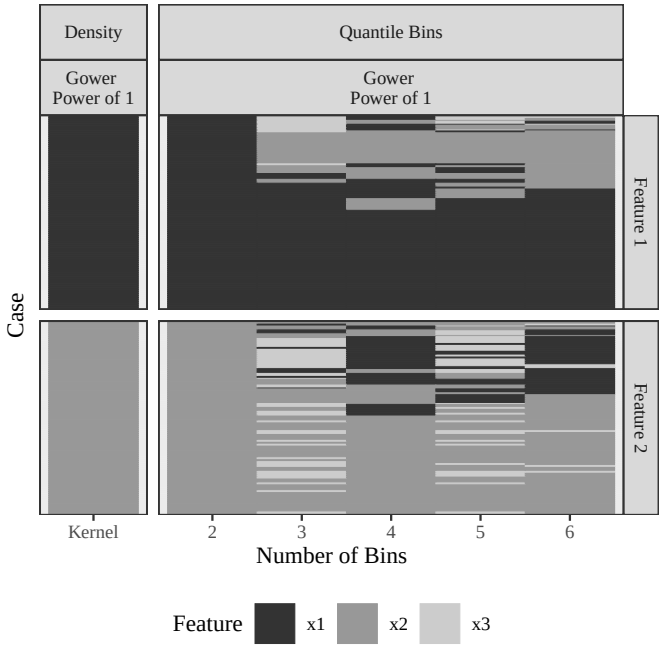

FIGURE 6 Feature heatmap of the various LIME implementations applied to the sine data. The cases from the test set are plotted on the y-axis, and the implementation methods are included on the x-axis. The colors of the tiles indicate the feature selected by LIME for the corresponding case and implementation method. The top faceted row includes the first features selected by LIME, and the bottom faceted row includes the second features selected by LIME. The faceted columns separate the kernel density implementation method from the quantile bin implementations. The vertical striping seen in the plot indicates that LIME is not consistent in selecting features across implementation methods.

Figure 6 shows the feature heatmap for the LIME implementations applied to the sine data. The top row of the plot contains the features selected as the most important by LIME, and the bottom row contains the second most important feature selected by LIME. For the quantile bin implementations, the original features prior to the interpretability transformation are plotted since it is obvious that different features would be selected when the sizes of the bins change from one implementation to the next. This figure shows a clear difference in the explanations produced by LIME for the kernel density and two quantile bin methods and the other quantile bin methods. For both the kernel density and two quantile bin implementations, LIME selects x_1 as the most important feature and x_2 as the

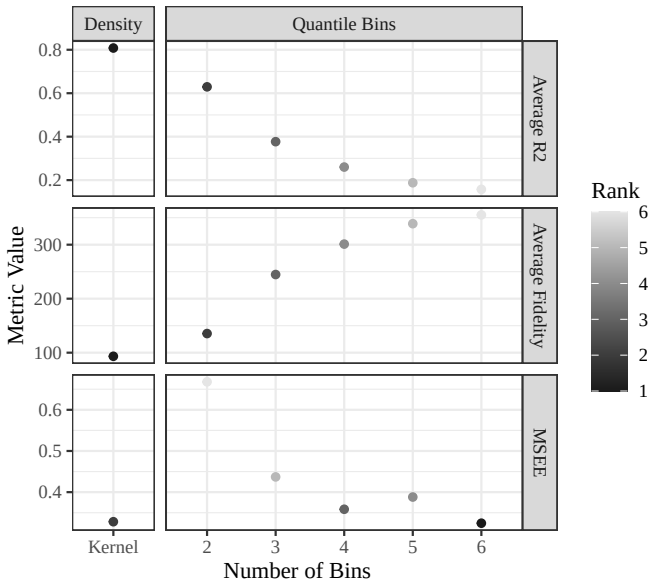

FIGURE 7 Example of a metric comparison plot using the sine data. Each row of the plot corresponds to one of three metrics: average fidelity, average R^2, or MSEE. The tuning parameters from the different LIME implementations are plotted on the x-axis, and the metric values are shown on the y-axis. The points are colored by rank within a metric indicating best to worst (dark to light). The kernel density simulation methods performs well according to all three metrics.

second most important feature across all cases in the test set. There is more variability in the features selected by LIME for the three to six quantile bin implementations. For these implementations, there is a mix of horizontal and vertical stripes, which suggests that LIME is not consistently providing the sample explanation across implementation methods. It is also interesting, that the implementations of 3 and 5 bins leads to the selection of the random noise variable of x_3 as an important variable in many predictions, which should not be the case.

3.3 | Assessment Metric Plot

The feature heatmap for the sine data in the previous section showed an example inconsistent LIME explanations across tuning parameter applications. As a result, it is necessary to choose tuning parameters that produce reliable explanations. One way to do this is to apply LIME using different implementations, and compute assessment metrics to determine the optimal tuning parameters. We will discuss three metrics that could be used for this purpose and present a comparison of the metrics in a plot.

Each metric presented below is computed on a set of LIME explanations associted with a set predictions of interest obtained using the the same tuning parameters.

- *Average* R^2: A metric for the assessment of the model fit and linearity assumption computed as the average of ridge regression explainer models R^2 values (deviance ratios from the R package *glmnet*).
- *Average Fidelity*: Provides comparison of the complex and simple model predictions for each of the values in the simulated dataset accounting for the assigned weight computed as the average of the fidelity metric presented in Ribeiro et al. [18] (weighted distance between explainer and complex model predictions for simulated data).
- *Mean Squared Explanation Error (MSEE)*: Measures the faithfulness of the explainer model to the complex model by comparing their predictions similar to the average fidelity, but only the observation of interest is used to compute the average squared deviation over the observations in the test set.

Notation and formulas for these metrics are included in Appendix **??**.

This is where I left off...

***Right now, I have not included any variability in these plots. I could add standard deviation bars or change these to box plot.

In Figure 7 , the average fidelity is lowest for the kernel density simulation method increases as the number of quantile bins increase. This would suggest that the LIME implementation using the kernel density estimate provides the best local approximation to the random forest model on average. Note that the points in the plot are colored such that darker indicates a better metric value and lighter indicates a worse metric value.

The 6 quantile bin simulation method has the lowest MSEE followed closely by the kernel density method, which suggests that these methods have explainer models that best approximate the complex model for the prediction of interest on average. The MSEE increases as the number of quantile bins increases.

The average R^2 value is highest for the kernel density simulation method and decreases as the number of quantile bins increase. This metric agrees with the average fidelity, and it suggests that the the kernel density method leads to the best fitting explainer models on average.

For the sine data, the metrics of average fidelity and average R^2 are in perfect agreement what do you mean by perfect agreement? and how do you use the fidelity measure? with which simulation methods lead to the recommended LIME

methods. The MSEE is in agreement with ? in terms of the kernel density method performing well in terms of faithfulness, but it orders the quantile bin methods in the opposite order from the average fidelity and R^2 values. This may be due to the MSEE only taking into account the prediction of interest and not the full simulated dataset.

The kernel density method performed well in regards to all three metrics. Recall that **??** indicated that the LIME kernel density method selected the same feature across all cases in the test dataset 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

The sine data is a relatively simple problem where the true decision boundary can be easily visualized and diagnostics for the performance of LIME can directly be compared to ground truth. In practice, we usually encounter more complicated situations. The remainder of this section provides a discussion of the visual diagnostics at the example of 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 come from the same source (are 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). This process is based on a visual and therefore subjective assessment of the evidence under a comparison microscope. 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 [16].

In response, Hare et al. [11] 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. [11] obtain signatures of striations from two bullet lands (Figure 8). 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 model to determine the probability of a comparison resulting from the same source (matching signatures) or from different sources (non-matching signatures).

FIGURE 8 (Top left) Traditionally rifled gun barrel. Grooves and lands alternate to give bullets a spin during the firing process. Barrel 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) Zoomed in image of a land engraved area showing striations (vertical lines).

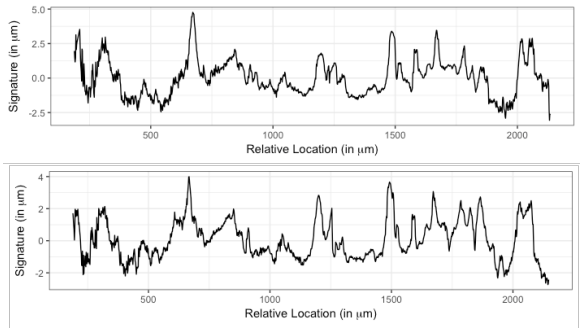

FIGURE 9 Bullet signatures extracted from two bullets fired through the same barrel. The two signatures come from the same barrel-land and therefore have very similar patterns. The features used in the random forest from Hare et al. [11] are different metrics that measure the similarity between two such signatures.

The random forest model trained in Hare et al. [11] is available as object rtrees from the *bulletxtrctr* R package. rtrees is

trained on a set of scans obtained from one set of bullets of the James Hamby Consecutively Rifled Ruger Barrel Study [9] and based on 83028 land-to-land comparisons.

rtrees was tested on another set of bullets from the Hamby study with 432 rows of land comparisons. Figure 10 shows parallel coordinate plots of the rtrees training (top row) and testing (bottom row) datasets. The plots on the left contain comparisons know to be from from different sources, and the plots on the right contain comparisons know to be from the same source. Each line represents one observation in the data, and the color of the line represents the by the corresponding random forest probability. The standardized feature values are plotted on the y-axis, and the features are plotted on the x-axis, which are ordered from left to right from the highest to lowest random forest feature importance.

The majority of observations with random forest probabilities close to 1 have a clear pattern of corresponding feature values as can be seen in the plots where same source is known to be true. The observations where the random forest is wrong in returning high probabilities when same source is known to be true (i.e. same source is true but the random forest probabilities are between 0 and 0.5) have feature values that reflect those of observations where same source is known to be false as seen in the plots on the left. These plots provide insight into why the random forest is producing poor random forest probabilities for these observations.

Since firearm identification is commonly used as evidence for convictions in court cases, it is important to be able to understand and assess the model that is used to quantify the probability that a bullet was fired from a gun. An application of LIME to rtrees could provide an understanding of the key variables used by the random forest model to make a prediction. However, just as it is important to assess the random forest model for this high-stakes application, it is also important to assess the LIME explanations to make sure they are providing a trustworthiness understanding of the random forest model. We will apply LIME to rtrees and use our visual diagnostics to assess different implementations of LIME.

4.2 | Application of LIME to Bullet Matching Data

We apply LIME to each prediction from the bullet test data obtained from the 'rtrees' random forest model for each of the four sampling methods (equally spaced bins, quantile bins, kernel density estimation, and normal approximation). For each of the bin based sampling methods, LIME was applied for 2 to 6 bins. It was decided to use 6 bins as the maximum, since the larger the number of bins, the more complex the explanation becomes. Each simulation method was implemented three times with Gower exponents of 0.5, 1, and 10. Thus, a total of $12 \times 3 = 36$ different implementations of LIME were performed. These implementations of LIME were performed using our R package *limeaid*. Each implementation was set to return 3 features by LIME and feature selection was performed using the highest weights method.

4.3 | LIME Assessment Visualizations

To get an overview of the LIME explanations from the 36 implementations, we consider our heatmap diagnostic plot for comparing the LIME implementations first shown in Figure 11 . The plot facets the features selected in the LIME explanations by the simulation method, the Gower power, the order the feature was chosen, and whether the observation is from a known match or non-match of bullet signatures. This plot highlights several key features of the LIME explanations from the bullet matching dataset.

First of all, the density simulation methods produced the same explanations for almost all cases and LIME implementation methods. As a result, the LIME explanations for the density implementation methods are global and not local. Second, within a bin based simulation method, the features selected by LIME for an observation often vary based on the number of bins used but do not appear to vary based on the Gower power used. Especially with the equal bin explanations, there are vertical stripes that suggest a dependence of the LIME explanations on the number of bins used. The vertical stripes are not as apparent with the quantile bins. Lastly, there are clear differences between the LIME explanations for the cases that are matches and those that are not matches which are produced by the bin based simulation methods. This provides evidence that suggests that the features which the random forests uses to classify a match are different from the features used by the random forest to classify a non-match. The most important insight Figure 11 highlights is the dependence of the explanations on the LIME implementation method for many of the observations in the test data.

To try to identify the LIME implementation method with the most trustworthy set of explanations, we created an assessment metric plot shown in Figure 12 . The three metrics of average fidelity, average R^2, and MSEE were computed for each of the simulation methods and Gower power used to implement LIME. The plot is faceted by the simulation method, and the shape of a point represents the Gower power. The color of a point represents the rank within a metric. That is, the darker a point, the better performance of that implementation method based on the metric.

The density simulation methods performed well across all implementation methods. This is an interesting result, since Figure 11 shows that the density methods results in global

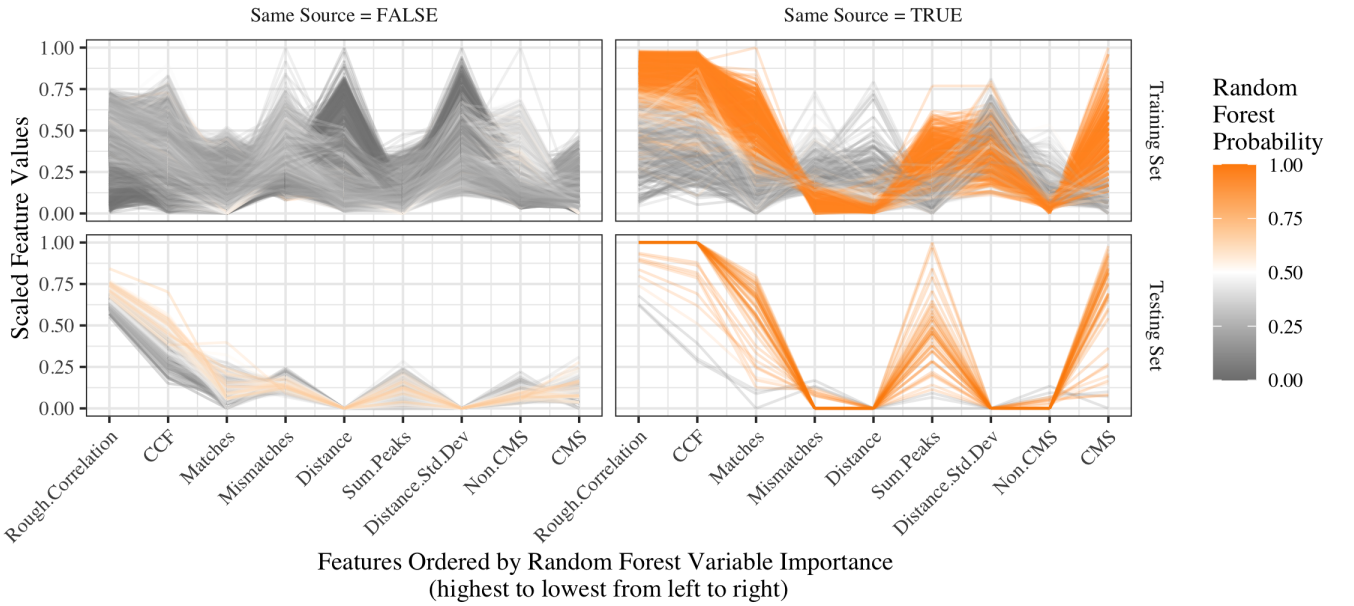

FIGURE 10 Parallel coordinate plots of rtrees training and testing data. The training data are included in the plots on the top row, and the testing data are included in the plots on the bottom row. The left column contains the observations which are know to be comparisons of lands that are not from the same gun, and the known same source comparisons are on the right. The y-axis shows the standardized feature values, and the x-axis shows the features used to fit the random forest (ordered by feature importance). Each line corresponds to a comparison, and the color of the line represents the associated random forest probability. These figures show a clear relationship between the feature values and the probability returned by the random forest.

explanations. The metric values for the bin based implementation methods do not agree across metrics. For example, 2 and 3 quantile bins performed well according to the average R^2 but poorly according to MSEE. As a result, with the bin based methods, it is difficult to know which method to recommend. When considering the metrics across the Gower power used, the implementations using a power of 0.5 perform best or as well as the other powers in across all simulation methods. This suggests that the power that leads to a more global explanation is preferred by LIME in this example.

Unfortunately, Figure 11 leaves us without a clear indication of which set of explanations to use to explain the random forest. Perhaps we can say that the kernel density method, 3 equal bins, and 3 quantile bins are some of the best performing methods. To better understand the explanations provided by these three methods, we take a closer look at explanations from these methods for two observations of interest where one is a known non-match (case 20) and the other is a known match (case 325).

Figures 13 and 14 contain plots that provide this closer look for the known non-match and match, respectively. The top row of plots in each figure are the explanation plots from *lime*. The bottom row of plots show scatterplots of the features selected by LIME colored by the random forest model prediction. The location of the prediction of interest is indicated on the scatterplots by a diamond, and black solid lines are included on the plots representing the bin divisions for implementations using bins. The bottom row plots are created using *limeaid*. The plots within a column are associated with one of the three implementation methods (kernel density, 3 equal bins, or 3 quantile bins).

The scatterplots provide a visualization that helps to explain the LIME explanation. For example, consider the explanation for case 20 when 3 quantile bins were used. The explanation plot from *lime* shows that case 20 having an observed CCF value greater than 0.320 supports a random forest prediction in favor of a match. The scatter plots show that many of the simulated values with CCF greater then 0.320 have random forest predictions greater than 0.5. Additionally, the LIME explanation indicates that a value of matches less than or equal to 1.66 contradicts a prediction in favor of a match. The scatter plots shows that almost all simulated values with a value of matches less than or equal to 1.66 have random forest predictions close to 0. Similar statements can be made about the 3 equal bin explanation for this case and the bin based implementations for the known match observation explanations.

While the scatterplots provide visual explanations for the LIME explanations, some of them indicate that LIME falls

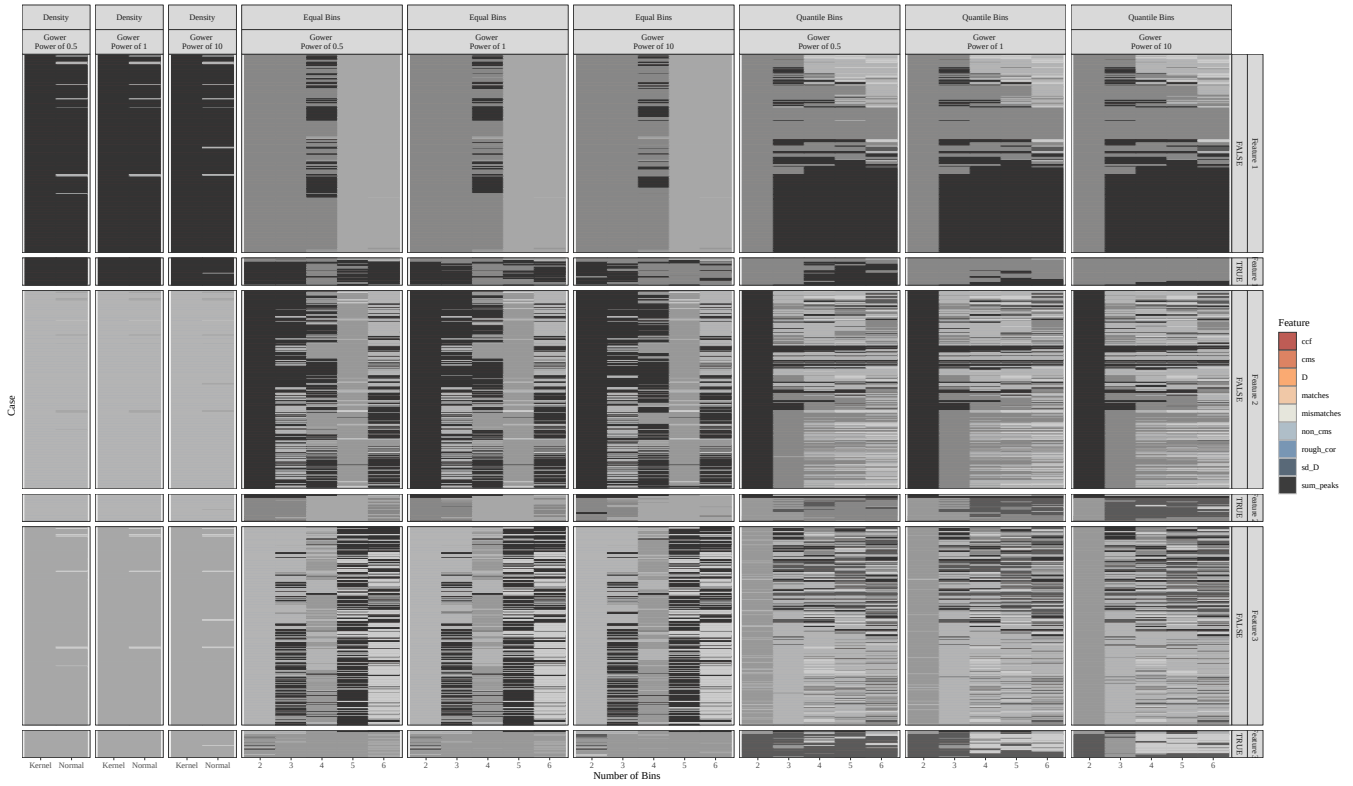

FIGURE 11 Feature heatmap of the 36 LIME implementations applied to the bullet comparison data test set. In addition to faceting the results by simulation method and order a feature was selected by LIME as done in **??**, this plot has been faceted by the Gower power and whether the the observation is from a match (TRUE) or non-match (FALSE). This plot shows several key insights about the LIME explanations. First, the features selected by LIME for the density simulation methods are mostly the same for all implementations and all cases. Second, the features selected by LIME for the bin based methods are affected by the number of bins but not the Gower power. Third, the explanations produced by LIME using bin based methods result in different important features for the matches and non-matches. ***Need to rotate the facet labels, so I can make the text bigger.

short of a good explanation of the random forest predictions. Again, consider the explanation for case 325 when 3 quantile bins were used. From looking at the scatter plot of CCF versus matches, a better explanation for the random forest prediction would be to say that because the prediction of interest has a value of CCF less than 0.8 (approximately) and a value of matches less than 7 (approximately), the random forest is providing a prediction that supports a non-match. The relationship between ccf and rough correlation does not provide much evidence to support the random forest prediction one way or the other due to the mixture of random forest predictions ranging from 0 to 1 in most regressions, but there is a pentagon shaped region at the bottom of the scatter plot of matches versus rough correlation that the prediction of interest falls in that supports the random forest prediction of a non-match.

The scatter plots of the kernel density implementations for both the match and non-match case show that almost all of the simulated values have random forest predictions less than 0.5, which provide evidence in favor of a non-match. It appears that simulation method is not providing enough values to cover the feature space that hit regions where the random forest provides predictions above 0.5. The training data has a much smaller amount of matches (1221) than non-matches (81807), and this simulation method is clearly affected by this imbalance in the two classification categories. Furthermore, this may be the cause of the minimal amount of variability in the LIME explanations across all of the cases in the test data.

Without applying multiple implementations LIME 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 , 14 , and 13) suggest that we should be cautious to trust any of these LIME explanations. While some of them provide an explanation that may be reasonable, there could be better explanations (as seen in the 3 quantile bin case of Figure 14), and some of them provide explanations that make no sense (as seen in **??**). It appears that either LIME needs to be further tuned to provide

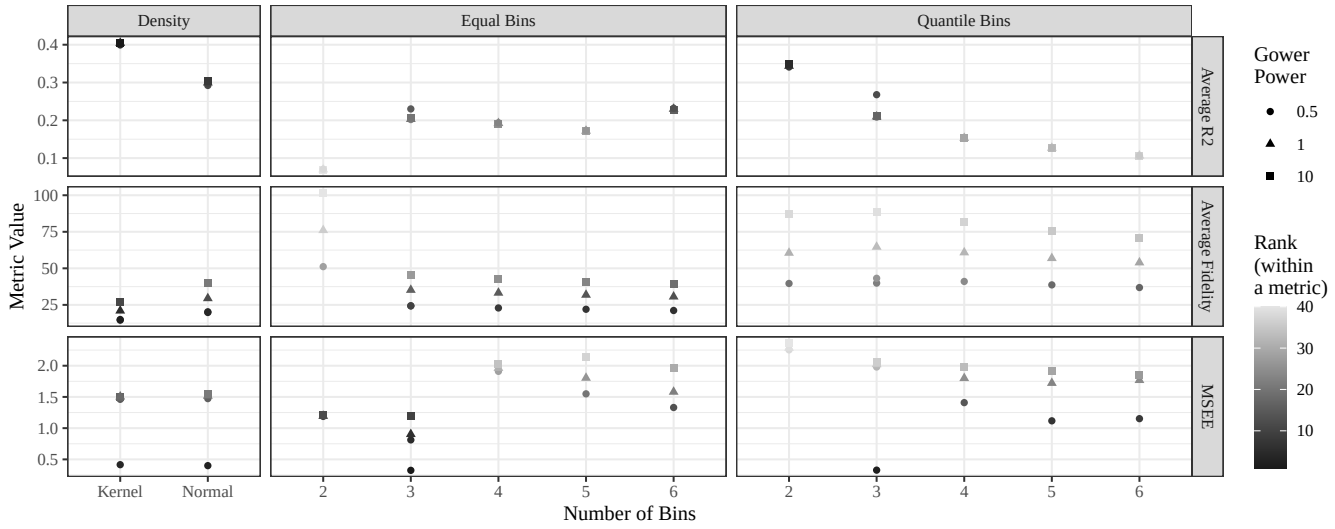

FIGURE 12 Plot of LIME assessment metrics for the 36 different implementations of LIME applied to the bullet comparison data test set. The shape of the points represents the power used for the Gower distance calculation, and the color of the points represents the rank associated with that observation within a metric. The density simulation methods perform well for all metrics, but the bin based metrics often do not agree in terms of performance across metrics.

trustworthy and good explanations, or a different approach may provide better insight. As glimpsed with the scatter plots in Figures 14 and 13 , the more traditional approach to explanation random forest predictions by plotting the random forest predictions over the feature space (partial dependence plots), could provide better insight into the complex model with no need to assess the explanation method if the sampling method is well chosen.

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. We suggest the use of visual diagnostics to counteract the black-box nature of LIME and provide three diagnostic plots. Here we discuss the limitations and possible extensions of the diagnostic plots and concerns about LIME raise by applications of the diagnostics plots in this paper.

5.1 | Diagnostic Plot Limitations and Possible Extensions

5.2 | Concerns about LIME

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 parameters. While the visualizations accomplish these tasks, they also expose examples of the failings of LIME. To address the discovered failings of LIME, we will reconsider each of the claims about the performance of LIME made by [18] (interpretability, faithfulness, linearity, and localness) in light of the insights gained from the diagnostic visualizations.

As previously discussed, the **interpretability** of the LIME explanations can be controlled 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, the presentation of a LIME explanation in the key visual used by [18] (shown in **??**) could lead to an over simplified interpretation of the explainer model. Without a solid understanding of the details behind the fitting of the explainer model, the deeper meaning of the bars in the figure is lost.

It is customary to interpret this plot by stating that the direction of the bar supports or contradicts the classification in a

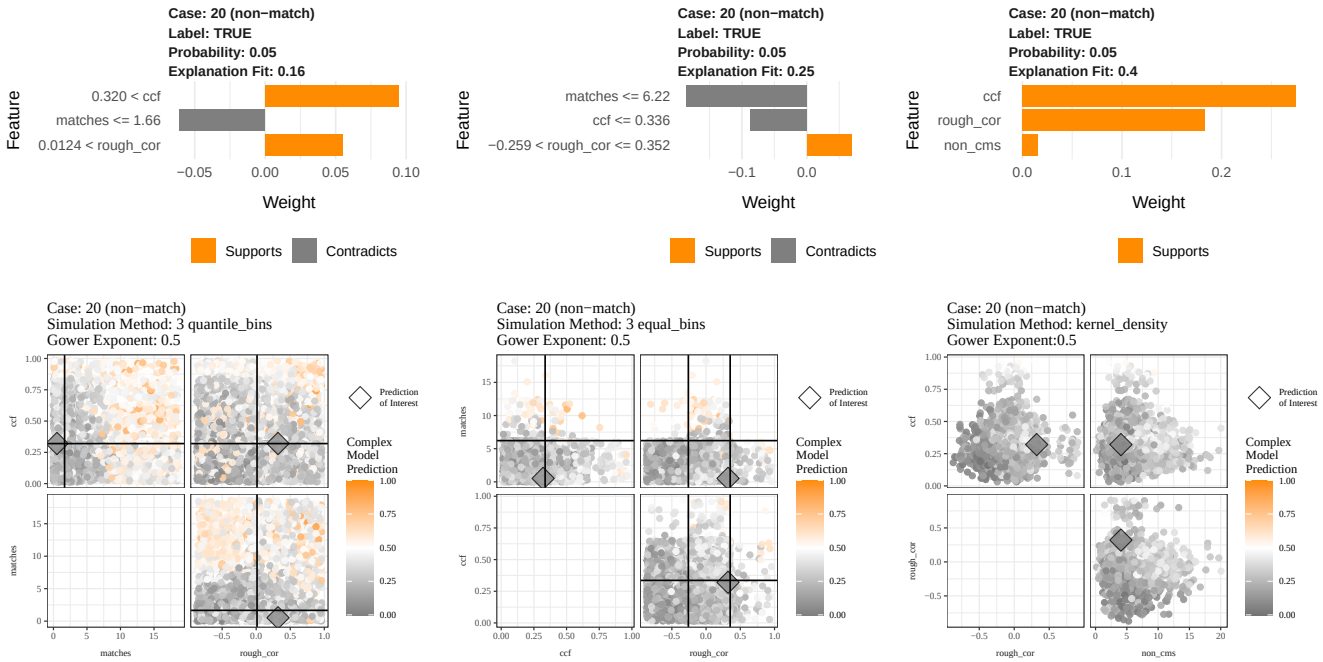

FIGURE 13 Plots of LIME explanations for one case in the test data that is a known non-match. Each column contains plots associated with a different implementation (3 quantile bins, 3 equal bins, or kernel density). The top row of plots are the explanation plots from *lime*. The bottom row shows scatter plots of the simulated data associated with the prediction of interest. The features plotted are those chosen by LIME in the explanations, and the points are colored by the the random forest predictions. Lines showing the divisions created in the feature space by the bin based LIME methods are included as solid black lines. The scatter plots can be used to better understand the LIME explanation and assess if it is a good explanation.

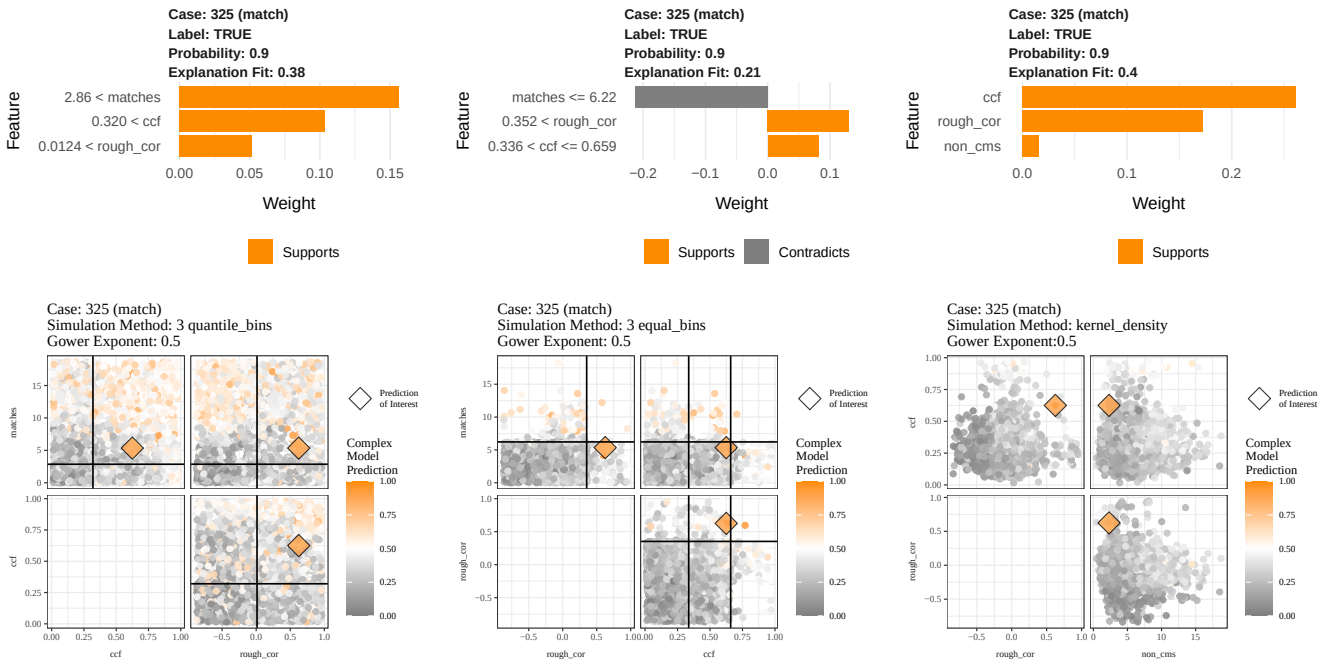

FIGURE 14 Plots with the same structure as 14 but for an observation from the test data that is a known match.

certain category, and the length of the bar signifies the importance of the involvement of a feature in the prediction. While these interpretations are correct, the lengths of the bars represent the coefficients of the ridge regression model used as the explainer model, which contain more meaning than just the direction of support and importance of a feature. The bars indicate how much the average complex model prediction estimated by the linear explainer model changes based on a change in the feature. Furthermore, this interpretation changes depending on whether a bin based or density simulation method was used (i.e the change in the average complex model prediction between an observation in the same bin as the case of interest and an observation not in the same bin versus the change in the mean complex model prediction for a change in the feature by one standardized unit).

Thus, even though an explainer model may be interpretable, the explanation it produces may be under-interpreted or misinterpreted without an understanding of the process that produced the explanation. Supplementing [18]'s compact visualization of the explanation with visualizations of the explanation that depict the simulated data and the explainer model (such as Figures **??** and 14) promotes a full interpretation of the explanation. This is done by providing reminders of the meaning of the lengths of the bars in addition to a visual connection between complex model predictions and the estimated coefficients of the explainer model.

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 easily 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 visualizations that incorporate the bins along with the complex model predictions (Figures **??**, **??**, **??**, **??**, 14 , and 13) highlight issues that accompany the choice to use bin based simulation methods. These visualizations allow for a comparison of the size of the bins to the complex model's decision boundaries. All of the examples in this paper show cases where the bins do not accurately capture the classification boundaries near the predictions of interest of the random forest models by oversimplifying the model. Using less bins would clearly not help improve the faithfulness of the explainer model in these examples, and while an increase in bins would lead to a finer resolution of the random forest classification boundaries, interpretability of the explainer model would quickly be lost. The bins used by LIME are not flexible enough to capture the different sized division boundaries of the random forests shown in this paper. 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.

The figures referenced in the previous paragraph from the sine data example are all based on the implementation of LIME using a set of tuning parameters. From the use of the feature heatmap, we found that different sets of tuning parameters can produce different explanations. It makes sense that one set of tuning parameters could lead to a more faithful explainer model than another. As a result, we proposed 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 confusing results. In both examples, the density based simulation methods resulted in the best or close to the best performance even though the feature heatmaps (Figures **??** and 11) showed that the density simulation methods produced global explanations with no variation in features selected by LIME. For the bin based simulation methods, the two metrics often did not agree or contradicted one another, which makes it difficult to decide on a recommendation of a set of tuning parameters that produces 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 was also seen with the residual plot (**??**).

The final claim, **localness**, is also addressed by the feature heatmap and metric comparison plot. As stated, the feature heatmap revealed that the density simulation methods in the examples of this paper resulted in global explanations where the same features were repeatedly chosen across all (or almost all) observations in the set of explanations. This finding agrees with that of [**?**] who also 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, but the patterns of features selected by LIME within a classification category do not vary. Again, this is a suggestion of global explanations. Furthermore, while the metric comparison plot in the bullet example did 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 was best. This suggests that a less local explanation provided 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 plots of the simulated data and explainer model.

While it would be ideal if LIME could be used as a method to provide easily understandable explanations for black-box models as [18] claim, that dream is not yet a reality. The examples using diagnostic plots to assess LIME in this paper show frequent issues with LIME. The practice of assessing the performance of methods has once again shown its importance. We hope that our plots provide motivation to assess LIME explanations, to not blindly use the default settings (even if it is not clear which tuning parameters to use), and perhaps to encourage work on improving LIME, so that it can be a lime and not a lemon.

ACKNOWLEDGMENTS

This is acknowledgment text. Provide text here.

Author contributions

This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text. This is an author contribution text.

Financial disclosure

None reported.

Conflict of interest

The authors declare no potential conflict of interests.

SUPPORTING INFORMATION

The following supporting information is available as part of the online article:

How to cite this article: Goode K., H. Hofmann, 2019, 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

B EXPLANATION SCATTERPLOTS UNDER OTHER SIMULATION SCENARIOS

Section 3.1 introduces explanation scatterplots under the default simulation method in the *lime* R package: four quantile bins. The structure of an explanation scatterplot remains the same for if any bin based simulation method is used to create the simulated data (any number of quantile or equally spaced bins). However, if the kernel density or normal approximation simulation methods are used as the simulation method, 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 ... provides example explanation scatterplots for each feature in the default LIME explanation for the sine dataindividual prediction of interest indicated in Figure 3 .

Need to updated limeaid to create this plot and then add figure here.

C DETAILS ON ASSESSMENT METRICS

??

Placing the notation definitions here for now - need to make adjustments as necessary: We first introduce notation for the scenario of tabular data with a binary response variable and continuous features. Let $\mathbf{X}$ be an n by p data matrix with p features and n observations, and let $x_i \in \mathbf{X}$ be a real-valued vector

such that $x_i \in \mathbb{R}^p$ for $i = 1, 2, ..., n$. Furthermore, let y be a response variable of length n with elements $y_i \in \{0, 1\}$ for $i = 1, 2, ..., n$. Suppose that f is a classification model where $f : \mathbb{R}^p \to [0, 1]$ that is applied to $\mathbf{X}$ and y. Let the vector of predictions made by f applied to $\mathbf{X}$ be denoted as $\hat{y}$. Note that $\hat{y}_i = f(x_i) \in \hat{y}$ for $i = 1, .., n$. For both bin based methods, the interpretability transformation converts the generated observations to indicator variables, where a 1 indicates that the feature value for an observation is in the same bin as the feature value for the case of interest. That is

$$z'_{i,j} = T\left(x'_{i,j}\right) = I\left[x'_{i,j} \text{ and } x^*_{i,j} \in b_{j,k}\right]$$

for $i = 1, .., m$ and $j = 1, .., p$ where $b_{j,k}$ is bin k for feature j. Let observation z'_i have weight $\omega_{x^*}(\mathbf{x}'_i) = Gower(x^*, x'_i)^c$, where $Gower$ represents the computation of the Gower similarity and c is some constant. This denotes a proximity measure between x^* and x'_i for each $i \in 1, ..., m$. If the exponential kernel is used, the weights are computed after the interpretability transformation such that z'_i has weight $\omega_{z^*}(\mathbf{z}'_i)$.

Ribeiro et al. [18] present the following metric as a measure of fidelity:

$$\mathcal{L}(f, g, \omega_{x^*}) = \sum_{i=1}^{m} \omega_{x^*}(x_i)\left(f(x_i) - g\left(z'_i\right)\right)^2,$$

where ...

This metric is computed as

$$\mathcal{L}(f, g, \omega_{x^*}) = \sum_{i=1}^{m} \omega_{x^*}(x_i)\left(f(x_i) - g\left(z'_i\right)\right)^2.$$

Smaller values of $\mathcal{L}$ indicate a better local approximation of the explainer model to the complex model.

That is, let

$$MSEE = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(x^*\right) - g\left(z^*\right)\right)^2.$$

Again, smaller values of MSEE would suggest a better approximation of the explainer model.

References

[1] Alvarez-Melis, D. and T. S. Jaakkola, 2018: On the robustness of interpretability methods.

[2] Bau, D., B. Zhou, A. Khosla, A. Oliva, and A. Torralba, 2017: Network dissection: Quantifying interpretability of deep visual representations.

[3] Community, C. o. I. t. N. o. t. F. S. and N. R. Council, 2009: Strengthening forensic science in the united states: A path forward. *National Academies Press*.

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

[5] Goodman, B. and S. Flaxman, 2016: European union regulations on algorithmic decision-making and a "right to explanation". doi:10.1609/aimag.v38i3.2741.

[6] 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

[7] Greenwell, B. M., B. C. Boehmke, and A. J. McCarthy, 2018: A simple and effective model-based variable importance measure.

[8] Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, 2018: A survey of methods for explaining black box models.

[9] 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.

[10] Hara, S. and K. Hayashi, 2016: Making tree ensembles interpretable.

[11] Hare, E., H. Hofmann, and A. Carriquiry, 2016: Automatic matching of bullet lands. *Annals of Applied Statistics*, doi:http://adsabs.harvard.edu/abs/2016arXiv160105788H.

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

[13] 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/

[14] Mohseni, S., N. Zarei, and E. D. Ragan, 2018: A survey of evaluation methods and measures for interpretable machine learning.

[15] Molnar, C., 2019: *Interpretable Machine Learning*.

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

[17] Pedersen, T. L. and M. Benesty, 2020: *lime: Local Interpretable Model-Agnostic Explanations*. Https://lime.data-imaginist.com, https://github.com/thomasp85/lime.

[18] 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.

[19] Sapra, R., 2014: Using r2 with caution. *Current Medicine Research and Practice*, **4**, no. 3, 130–134, doi:10.1016/j.cmrp.2014.06.002.

[20] 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/

[21] 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.