Literature Review of Explainable AI Methods
CIS 6250: Theory of Machine Learning

Ryan French, Conor Gibbons

Abstract

Explainable artificial intelligence is a subfield of artificial intelligence that focuses on designing and building AI systems that can be understood and explained by humans. The goal of Explainable AI is to develop AI systems that are human interpretable and understandable. Interpretability is important for building trust and ensuring accountability in AI systems, many highly effective AI models are complex and difficult to interpret. To address this challenge, researchers have developed a range of techniques for improving the interpretability and understandablility of AI systems, including many different model-agnostic explanation methods. This review explores some of the current techniques and methodologies in the field of Explainable AI.

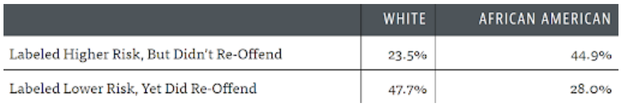
Introduction

As machine learning and artificial intelligence continues to permeate into the daily lives of humans and decision making processes new tooling is needed to reveal insight into the black box nature of modern artificial intelligence (AI). Trust in the system and integrity of the outputs should be a fundamental goal when deploying artificial intelligence systems. Increasing human understanding of the black box decision making process of AI is one way to garner trust. Currently, there exists two definitions of trust in the outputs of a model: trusting a prediction, i.e. providing local understanding, and trusting a model, i.e. providing global understanding [1]. High impact decision making processes that rely on artificial intelligence based models tend to favor simpler, more interpretable models such as linear regressions models. However, advancements in the field of AI continuously generate newer, more accurate, albeit, more complex models.

Given these developments and the increasing application of AI technologies across economic sectors, stakeholders from academia, industry, and civil society have called for the federal government to become more knowledgeable about AI technologies and more proactive in considering public policies around their use. The Congressional Research Service in 2021 created committees to encode four laws in the AI space related to algorithmic bias, Generative Adversarial Networks (GANs), deepfakes, and financial support for cutting-edge AI research. [1] Per the CRS, AI systems may perpetuate or amplify bias, may not yet be fully able to explain their decisionmaking, and often depend on vast datasets that are not widely accessible to facilitate research and development (R&D). The current lack of knowledge into such networks and funding in this space limits society's ability to deal with societal and ethical issues that may arise.

Put broadly, Explainable AI translates the complex machinery of black-box ML/AI models into information digestible by a wide range of stakeholders. As AI becomes more prevalent in our daily lives, a much greater proportion of our non-technical population becomes stakeholders of black-box models. It is therefore essential to spend time, resources, and legislation developing Explainable AI so that frustratingly opaque models can be unraveled for the benefit of the world's stakeholders.

A plethora of alarming events in mainstream media further motivate the rapid development of Explainable AI. For much of the 2010s, Amazon utilized an AI recruiting tool to autonomously sift through thousands of applications [2]. This model went unnoticed for a decade for its opacity and lack of stewardship surrounding knowledge of input features. Ultimately, the model favored male applicants over female applicants due to societal influences in preferring men in tech jobs.



	WHITE	AFRICAN AMERICAN
Labeled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Figure 1. Accuracy metrics of Northpointe's Ft. Lauderdale algorithm

A crime risk AI model deployed in Broward County, Florida, home to Ft. Lauderdale, was reported to label black defendants nearly twice as likely to commit future crimes as white defendants [3]. Figure 1 depicts Type I and II errors with respect to both populations. The algorithm, created by for-profit Northpointe, is one of many deployed across the country assigning risk assessments to convicted citizens for future crimes. Northpointe does not disclose features of its proprietary mode to the media. These risk scores are used in trials to recommend sentences, where defendants argue for its misuse and plaintiffs support its legitimacy.

A research group in 2015 at Mount Sinai Hospital in New York developed a wildly successful ML algorithm called Deep Patient to predict the onset of a variety of indications [2]. The network, trained on records from 700,000 patients, was lauded for its ability to predict schizophrenia, a mental health disorder historically difficult to predict. Deep Patient provided no perspective into how its predictions are made, however, providing medical professionals with no value in diagnosing their patients. Its black box nature severely limited its contribution to society. Feature importance tools, when applied to Deep Patient, would transform the network into one of the most useful AI tools in medtech.

On occasion, it serves to elect to use simpler models for the sole purpose of explainability. An important aspect to introduce into the definition of Explainable AI is to never use black box models in the first place. If coefficients of input features are already easy to interpret, or if simple functions can backpropogate the gradients of explainable weights, models with these attributes inherently fall under the umbrella of Explainable AI. Bhatt et. al in December 2022 circumvented using a traditional CNN when developing retinopathic diagnoses by creating a semi-automated doctor-in-the-loop workflow [4]. Within the workflow, an initial set of features are created, and more features are engineered using medical professionals' initial decisions. These features are then fed into a wide class of models that output easily interpretable feature importance metrics. Had the group used a CNN, their findings would not have been as nearly as easy to communicate to stakeholders, including the patients themselves.

Clearly, modern society has created a dire need for the development of Explainable AI. As the population of AI's stakeholders grows year over year, data scientists must take on the additional responsibility of creating explainable models. Dying are the days of the public blindly accepting black box algorithms; Explainable AI is suddenly not just a desire, but an ethical requirement. This review of the current state of Explainable AI includes global properties of AI models that Explainable AI aims to explain, and feature importance explanation tools such as SHAP and LIME.

Global Model Properties

Bias

When applying a deep, black box algorithm, it is imperative to communicate the model's intentions and global properties to all stakeholders. AI/ML models are susceptible to inherent bias, often introduced by training data rather than the model attributes themselves. One of Explainable AI's main goals is to limit bias and offer windows into black boxes, allowing engineers to determine root causes of potentially harmful scenarios.

Research performed by Mark Yatskar of the Penn CIS department[5] explores the failures of ELMo's contextual word embeddings surrounding ethically constructed training sets. ELMo, Embeddings from Language Models, is an industry-leading bi-directional LSTM model in the Natural Language Processing (NLP) space. ELMo uses a geometric representation of language called word embeddings that numerically plots each word in a training corpus in a fixed dimensional space, often around 100 dimensions. Zhao et. al, the team including Professor Yatskar, performed a series of intrinsic analysis on ELMo's word embeddings and found a series of critically important biases in how language was modeled in 2019. First, the frequency of male entities (5.3 million) far exceeded the frequency of female entities (1.6 million) in the training corpus for ELMo, the One Billion Word Benchmark [6]. Second, the embeddings unfairly associate some occupations with primarily female contexts and pronouns, and other occupations with primarily male contexts and pronouns.

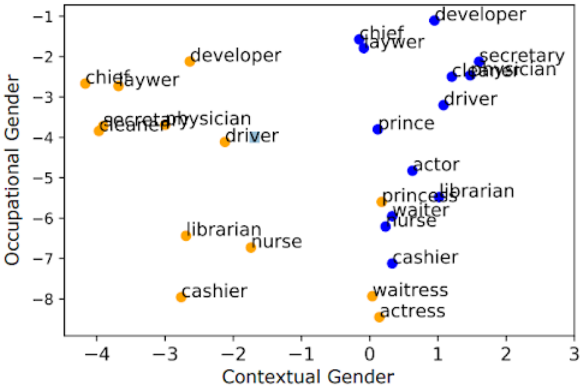


Figure 2. Selected words from Zhao et. al projecting to the first two principal components where the blue dots are the sentences with male context and the orange dots are from the sentences with female context.

Figure 2 shows how the y axis, "Occupational gender", one of the first two principal components of ELMo word embeddings, projects each embedding onto genderdependent space.

Similarly, Bolukbasi et. al wrote a groundbreaking paper in 2016 titled Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings [7]. These researchers dive into the geometric relationship between male-classified and female-classified occupations to the extent that vectors of gender bias can be analyzed. In one particular example, illustrated by Figure 2, the vector drawn from 'man' to 'woman' is approximately equal to the vector drawn from 'computer programmer' to 'homemaker'.

$$\overrightarrow{man} - \overrightarrow{woman} \approx \overrightarrow{computer\ programmer} - \overrightarrow{homemaker}$$

Zhao et. al propose a data augmentation method for counteracting the issues in a training set as biased as the One Billion Word Benchmark: data augmentation. By simply replacing all values of 'he' with 'she', and swapping other pairs with their counterparts of the other gender, they were largely effective at mitigating bias, though concluded fully eliminating bias in any training set is a near impossible task.

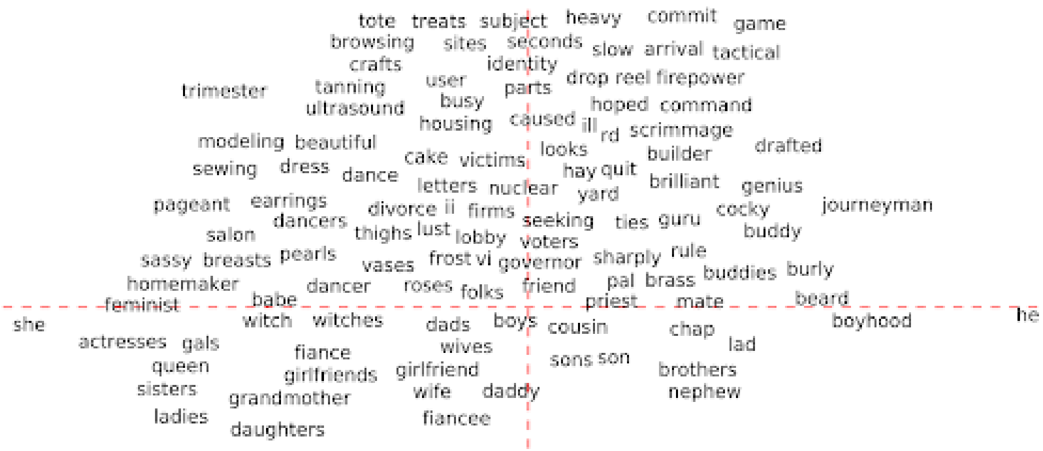


Figure 3. Selected words from Zhao et. al projecting to the first two principal components where the blue dots are the sentences with male context and the orange dots are from the sentences with female context.

Concept Drift

A major challenge with deployed, real-world ML models is the notion of concept drift. Since data change over time, the relationship between a deployed model's inputs and outputs might dramatically change to the extent that predictions are no longer useful after some observed time from initial deployment. ML algorithms work under the assumption that the data distribution used to evaluate the model is similar to the data distribution of the reference data. Explainable AI looks to combat concept drift by altering models as time elapses in a sense that the models' alterations can be easily communicated to stakeholders.

One popular method for handling concept drift is reweighting [8], which leverages a loss function weighted heavily on recent samples. Jain and Shenoy use the following linear combination of exponential time decay factors, decaying each feature's recency weight by an appropriate scale.

$$w_i = \sum_k z_k e^{-a_k t_i} = \mathbf{z}^T e^{-\mathbf{a} t_i}$$

The backpropogation of this now time-weighted loss function allows models to re-weight as often as every time they see a new sample.

Naively trained models are particularly susceptible to concept drift. If a model learns features that are not unique to the training data, then as these data change, the model may instead infer consistently trivial features as opposed to changing informative features. RNNs, for example, can often converge to modeling sequence length as an important feature [9]. Baillargeon et. al explain that the difference of lengths of different classes of text inputs can very easily create concept drift; they were able to mitigate the trivial nature of this feature by leveraging regularization terms in their loss functions.

Strategies such as "Forgetful Forests" [10] throw away training data considered old and only use more recent, relevant samples. Yuan et. al construct a Forgetful Decision Tree and Forgetful Forest that sequentially and probabilistically forget old data and combine the retained old data with new data. Every time they train their trees/forests, they compare the new model's accuracy to its previous accuracy. If accuracy increases, they increase the window of training data they use in hopes the model continues to generalize well. If accuracy instead decreases, they decrease the size of the training data window significantly to only look at more recent data. They then empirically construct a rate of change of this window size parameter that depends on the rate of change of accuracy measurements and the current window size to enable quicker learning.

Concept drift detectors are becoming increasingly necessary with the maturation of ML algorithms in every-day decision making. Figure 4 illustrates a workflow of when concept drift should be addressed in deployed ML models [11].

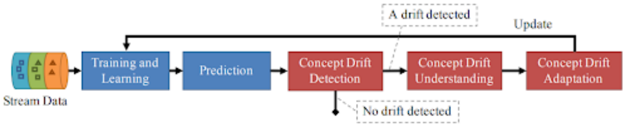


Figure 4. Framework for handling concept drift in machine learning

Lu et. al explore several classes of concept drift detectors. Error rate-based drift detection tracks changes in accuracy of online models. If error rate changes show to be statistically significant, a new learner is trained. Error rates can be defined as the distance between accuracy metrics or computing the distance between two correct classifications in a discrete setting, among others. Probabilistic time windows can be introduced in attempt to eliminate arbitrary hyperparameters in retrained models.

Data distribution-based drift detectors compare the distributions of existing training data and new training data. If these two distributions are statistically significantly different, a retraining flag will be triggered. A major hyperparameter of these detectors is window size and location, which build density functions f1 and f2 of the model's inputs. A distance function is then defined as

$$dist_{L^1} = \int |f_1(x) - f_2(x)|\, dx$$

Kefir et. al theoretically prove the bounds on the probabilities of missed detections and false alarms using Chernoff bounds and VC-dimension [12]. KL-divergence is another effective metric at determining suitable distances between f_1 and f_2 to detect drift.

Multiple statistical tests can also be run on the same two density functions in series or in parallel to raise a concept drift flat. As shown in Figure 5, the results of many hypotheses tests H_i $i = 1, ..., z$ can be aggregated to produce detection results between distributions W_{hist} and W_{new}.

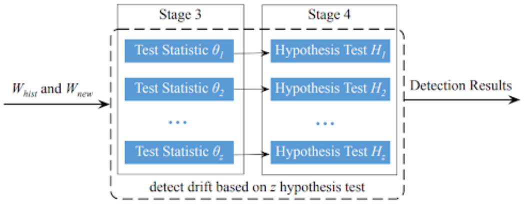


Figure 5. Parallel hypotheses test for drift detection

The results of these drift detection algorithms should be used in online ML models to limit class imbalance and identify hidden biases in streamed data. A glaring issue with drift detection is, while it has the ability to answer when drift occurs, it struggles to answer how or why. Once drift occurs, we turn to feature importance explanation methods covered below to provide additional, necessary context.

Additional critics of detection drift include Poenaru-Olaru et. al, who show empirical evidence that error-based drift detectors signal an unsatisfactory rate of false alarms for wide windows [13]. This questions the detectors' ability to detect drifts in environments where the features are slowly changing over time. An example is inflation, which does not have a clear immediate impact on financial features, but affects them over a longer period of time.

Clearly, neither drift detection nor bias resolution are simple issues to overcome when training black-box ML models. Diagnostic tests such as class balance metrics and drift detection algorithms should be run on not only the first instance of deployed models, but every time models are trained, in order to preserve their ethical and accurate properties. The broad intentions of models must always be explained to stakeholders, and the necessary checks must be run regularly and often to maintain the integrity of these checks. The implications of explainable AI on model-level properties are not limited to bias and concept drift, though they both must be assessed at each stage of a model's lifecycle to reasonably ensure reliability.

Perturbation Based Explainable AI Methods

Local interpretable model-agnostic explanations (LIME)

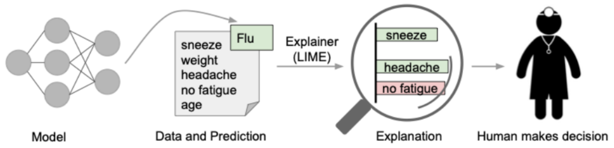


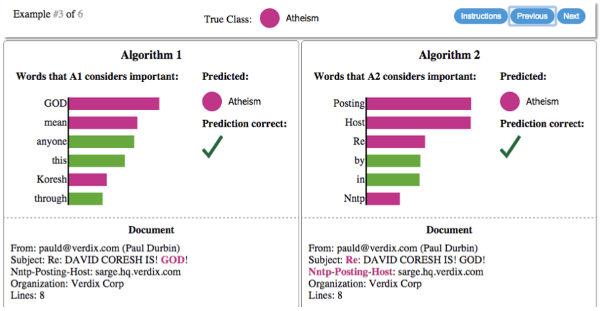
Figure 6. Explaining individual predictions. A model predicts that a patient has the flu, and LIME highlights the symptoms in the patient's history that led to the prediction. Sneeze and headache are portrayed as contributing to the "flu" prediction, while "no fatigue" is evidence against it. With these, a doctor can make an informed decision about whether to trust the model's prediction.

LIME is a localized explanation technique developed by Ribeiro Et. al that can explain the output of any classifier. LIME proposes to build a global understanding of any classifier in an interpretable way by providing explanations for a sufficient amount of individual predictions.[14] LIME allows explanations for each prediction of any classifier to be made by fitting a localized model to neighborhoods of datapoints. The localized explanation model is fit on the classifier's outputs of in sample data points, as well as the outputs of out of sample data points. These out of sample data points are generated via small perturbations of features of the original data points such that they remain local. The goal of fitting a localized model to these points is to produce a new model with reduced complexity and higher interpretability (e.g. linear regression) that maintains similar outputs to the true model.

The authors of LIME suggest that many machine learning practitioners overestimate the accuracy of their models during training and validation. They argue that the models are prone to performance drop off "in the wild" that may not be caught prior to deployment. LIME can provide a global understanding of a model's predictions by gathering a collection of individual predictions such that the space of possible inputs is sufficiently covered. Explaining predictions of complex models in an interpretable and human understandable way is an important aspect in gaining trust and acceptance of models being used in practice. In addition to providing insight into predictions LIME provides another layer of model feedback that can be used in conjunction with traditional machine learning metrics such as accuracy.

Advantages of LIME

LIME is a powerful tool that scaffolds localized explainable models on top of complex models like neural nets and ensembles. The localized models can be probed to identify feature importance in predictions while also maintaining a high level of local fidelity, i.e. the explainable model behaves how the complex model behaves in the vicinity of the instance being classified or predicted.



From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Figure 7. Explaining individual predictions of competing classifiers trying to determine if a document is about "Christianity" or "Atheism". The bar chart represents the importance given to the most relevant words, also highlighted in the text. Color indicates which word contributes to (green for "Christianity", magenta for "Atheism").

Ribeiro Et. al conducted an experiment in which non-experts in the field of machine learning were tasked with selecting a "best" model, i.e. one that generalized best to the real world on the 20 newsgroups dataset. This experiment provided two useful outcomes. The first outcome was that non-experts were able to use the outputs of LIME to identify the more generalizable model (Figure 2). [14] The LIME outputs in this experiment were the words, i.e. features, used to categorize a piece of text as religious or not religious. Without prior machine learning expertise the participants were able to choose the better model based on these LIME outputs alone. The second outcome of the experiment is that LIME outputs can be used for feature engineering. Experiment participants were able to iteratively improve model accuracy by removing features deemed to be insignificant by LIME.

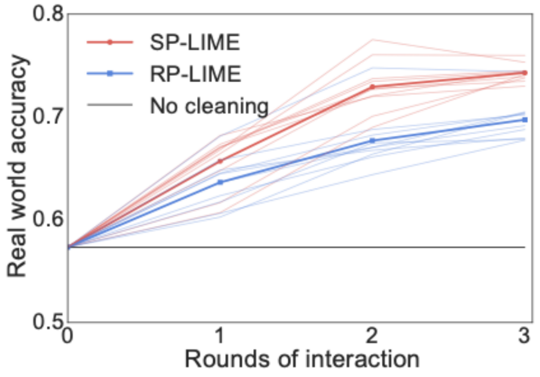
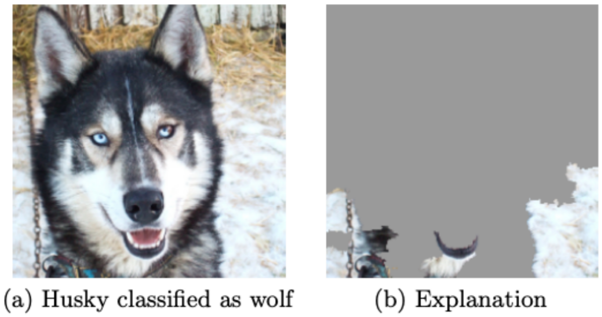


Figure 8. Feature engineering experiment. Each shaded line represents the average accuracy of subjects in a path starting from one of the initial 10 subjects. Each solid line represents the average across all paths per round of interaction.

In another experiment Ribeiro Et. al trained an classifier to be intentionally biased while classifying images as either a husky or a wolf.[14] The classifier predicted an image to be a wolf if the image had snow or any light color in the background and husky otherwise. Participants in the experiment were then shown the predictions of a dataset which included an image of a husky with a snowy background and a wolf with a non-snowy background. The predictions were both wrong due to the intentionally biased training on image background instead of animal presence; however, 10 out of 27 subjects still maintained trust in the model after seeing the result. Upon revealing the LIME outputs of the model on the data set only 3 out of 27 subjects maintained their trust in the validity of the model.


(a) Husky classified as wolf

(b) Explanation

Figure 9. Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

Figure 10. "Husky vs Wolf" experiment results.

The subjects in the experiment were graduate students who had taken at least one graduate level machine learning course. When asked to describe how the model was making distinctions between wolves and huskies there was a 2 times increase in the mention of snow as a potential feature after seeing the LIME outputs of the model. Although the size of this experiment was not large (n = 27) the results show the promise of LIME to enable machine learning practitioners to exercise additional caution when validating models.

In addition to explanations for model predictions on individual instances Ribeiro Et. al propose a Submodular pick (SP) algorithm to produce a global understanding of a model.[14] Their algorithm proposes to produce a sufficient number of explanations on instances that cover the feature space. The goal is to provide non-redundant explanations that give a global understanding of model behavior. Importantly, their algorithm places a budget B on human patience to ensure it can run in reasonable time.

```
Algorithm 2 Submodular pick (SP) algorithm
Require: Instances $X$, Budget $B$
  for all $x_i \in X$ do
      $\mathcal{W}_i \leftarrow$ explain$(x_i, x'_i)$                ▷ Using Algorithm 1
  end for
  for $j \in \{1 \ldots d'\}$ do
      $I_j \leftarrow \sqrt{\sum_{i=1}^{n} |\mathcal{W}_{ij}|}$      ▷ Compute feature importances
  end for
  $V \leftarrow \{\}$
  while $|V| < B$ do            ▷ Greedy optimization of Eq (4)
      $V \leftarrow V \cup \mathrm{argmax}_i\, c(V \cup \{i\}, \mathcal{W}, I)$
  end while
  return $V$
```

Figure 11. SP Algorithm

Disadvantages of LIME

While LIME is a potential solution to the black-box explainability problem it still comes with a handful of disadvantages that prevent it from becoming a universally accepted method of model interpretability.

First, the aforementioned SP algorithm requires human intervention. Thus, LIME is not capable of reliably and repeatedly producing a global understanding of any arbitrary model by itself. Another notable disadvantage is that LIME places no guarantees on global fidelity, i.e. features that

LIME explanations deem important for one instance are not necessarily important for the entire model and vice versa. While the proposed SP algorithm attempts to circumvent this fault it is still not a perfect solution. Secondly, because LIME fits an explanation model of lower complexity to the neighborhood of a data point it lacks robustness.

Alvarez-Melis Et. al define robustness of an explanation method such that similar inputs reliably reproduce similar explanations and that the local explanation model can be used in lieu of the complex model in a small epsilon neighborhood[15]. Alvarez-Melis Et. al argue that robustness is a highly desirable feature of AI explanation methods. Additionally, they believe that a single point-wise explanation method of complex models is naive. Their work shows that LIME performs well on classifiers such as linear SVMS; however, as model complexity increases, e.g. neural nets, in some cases the robustness assumption can break down with neighboring points becoming inconsistent with each other.

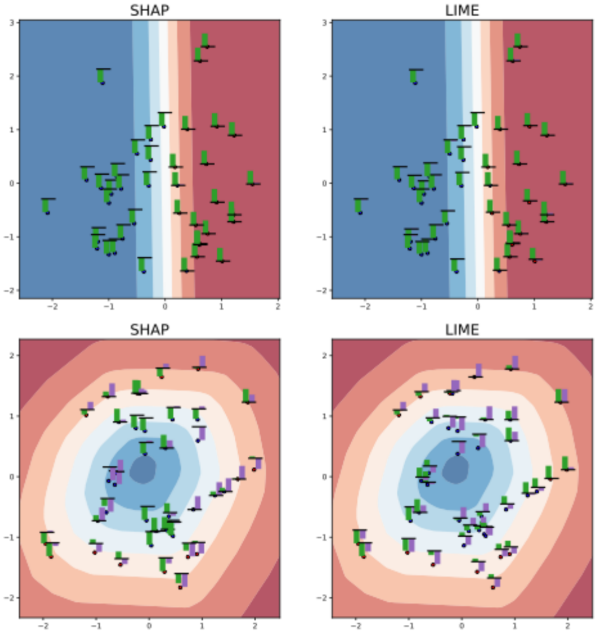


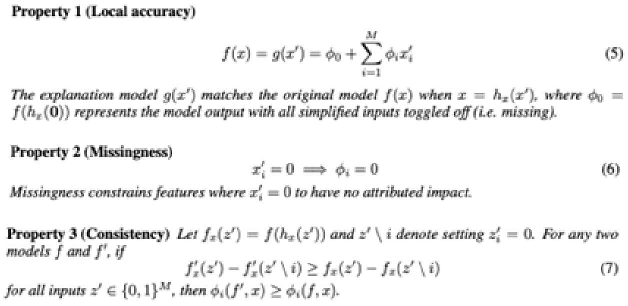
Figure 12. LIME and SHAP explanations for two simple binary classifiers: a linear SVM (top row) and a two-layer neural network (bottom). The heatmaps depict the models' positive-class probability level sets, and the barchart inserts show the interpreters' explanations (attribution values for x in green and y in purple) for test point predictions. While both LIME and SHAP's explanations for the linear model are stable, for the non-linear model (bottom) they vary significantly within small neighborhoods.

The inherent trade-off between high fidelity and high interpretability demonstrated above is perhaps one of the biggest faults of LIME. It is not obvious and Ribeiro Et. al do not provide solutions regarding the problem of producing highly accurate explanation models that are still human interpretable. Finally, a fundamental feature of LIME as an AI explanation method is that it is model agnostic. Model agnosticism requires the model explanation optimization function parameters to be chosen heuristically. As Lundberg Et. al demonstrate in their work with SHAP, moving towards a model-driven optimization framework provides computational speedups versus vanilla LIME[16].

LIME is a step in the right direction towards explainable AI, albeit an imperfect one. It demonstrates potential to be a powerful tool in model validation, but it by no means can produce 100 percent reliable explanations for any model.

Shapley Additive Explanation Values (SHAP)

SHAP proposes to be a unified framework for interpreting predictions. Lundberg Et. al propose that SHAP unifies the theory of six existing additive measures of feature importance with LIME being one[16]. Furthermore, they argue that there exists a unique solution to the additive measure of feature importance that adheres to a set of desirable properties. The set of properties includes Local accuracy, Missingness, and Consistency. Local accuracy requires the output of the explanation model with a simplified input to match the output of the same input to the original model. Missingness states that features missing from the original input should have no impact on model explanations. Finally, the property of Consistency, which is how Lundberg Et. al refer to monotonicity, requires that if some feature input increases or stays the same regardless of other inputs that features impact does not decrease. [16]



Property 1 (Local accuracy)

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{5}$$

The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$, where $\phi_0 = f(h_x(\mathbf{0}))$ represents the model output with all simplified inputs toggled off (i.e. missing).

Property 2 (Missingness)

$$x'_i = 0 \implies \phi_i = 0 \tag{6}$$

Missingness constrains features where $x'_i = 0$ to have no attributed impact.

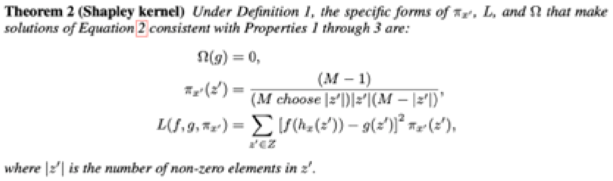
Property 3 (Consistency) *Let $f_x(z') = f(h_x(z'))$ and $z' \setminus i$ denote setting $z'_i = 0$. For any two models f and f', if*

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \tag{7}$$

for all inputs $z' \in \{0,1\}^M$, then $\phi_i(f', x) \geq \phi_i(f, x)$.

Figure 13. Essential properties of features needed to create SHAP values

These properties have been associated with classical game theory based estimations of Shapley values. Lundberg Et. al were the first to apply the game theoretic solution concept of distributing gains and costs of players to feature attribution. This newly developed feature attribution method claims to build upon previously implemented methods to improve computational efficiency and to be better aligned with human intuition.[16]



Theorem 2 (Shapley kernel) *Under Definition 1, the specific forms of $\pi_{x'}$, L, and Ω that make solutions of Equation 2 consistent with Properties 1 through 3 are:*

$$\Omega(g) = 0,$$

$$\pi_{x'}(z') = \frac{(M-1)}{(M \text{ } choose \text{ } |z'|)|z'|(M-|z'|)},$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} \left[f(h_x(z')) - g(z')\right]^2 \pi_{x'}(z'),$$

where $|z'|$ is the number of non-zero elements in z'.

Figure 14. Shapley Kernel

Advantages of SHAP

SHAP values are hard to compute and must be approximated. Lundberg Et. al used insights from preexisting feature attribution methods to develop both model agnostic and model aware novel approximation methods. Kernel SHAP is their novel approach to model agnostic Shapley value approximation. By connecting Shapley values from game theory to weighted linear regression via theorem 2 the values can be approximated with greater accuracy in fewer function evaluations of the original model than LIME.

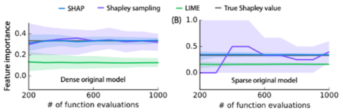


Figure 15. Empirical efficacy of SHAP vs LIME

The values produced by the Shapley kernel follow from the previously mentioned properties. Moreover, the feature attribution values produced with LIME can differ significantly. SHAP ensures the properties of local accuracy and consistency are satisfied while approximating Shapley values. This updated framework provides greater rigidity on feature attribution that LIME cannot guarantee.

Unlike LIME, SHAP can improve the computation efficiency of model-agnostic approximations by restricting the approximation method to specific model classes. For low ordered linear models it follows from Theorem 2 that the Shapley values can be calculated directly from the weighted coefficients. Computational efficiency of Shapley values can also be improved for deep neural networks by leveraging connections between SHAP and DeepLIFT, a feature importance methodology for deep neural networks.

In addition to the benefits of computational speed ups feature attribution from SHAP can also be better aligned with human intuition than other preexisting methodologies. Lundberg Et. al found that user explanations of simple models were more consistent with SHAP than either LIME or DeepLIFT.[16]

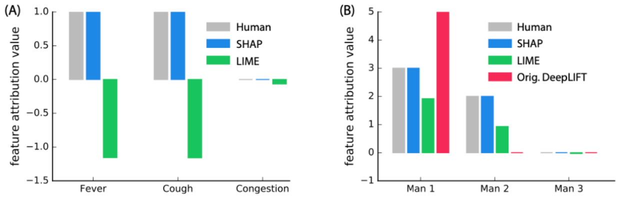


Figure 16. Human feature impact estimates are shown as the most common explanation given among 30 (A) and 52 (B) random individuals, respectively. (A) Feature attributions for a model output value (sickness score) of 2 . The model output is 2 when fever and cough are both present, 5 when only one of fever or cough is present, and 0 otherwise. (B) Attributions of profit among three men, given according to the maximum number of questions any man got right. The first man got 5 questions right, the second 4 questions, and the third got none right, so the profit is $5.

In the first experiment participants were shown the output of a sickness score model that was higher when only one of two symptoms was present. Figure 4 shows one possible output of this model in which SHAP and human feature attribution values are perfectly aligned. While this is not a rigorous proof of the superiority of SHAP over LIME it does reveal the possibility that feature attribution methods can differ significantly from each other and from human intuition. The second experiment required participants to decide how to distribute profits amongst three fictional people based on the amount of questions they answered correctly. Again the SHAP feature attribution values were consistent with human intuition. Furthermore, in this experiment, SHAP feature attribution for max functions improves upon DeepLIFT's inability to perfect due to its reliance on max pooling.

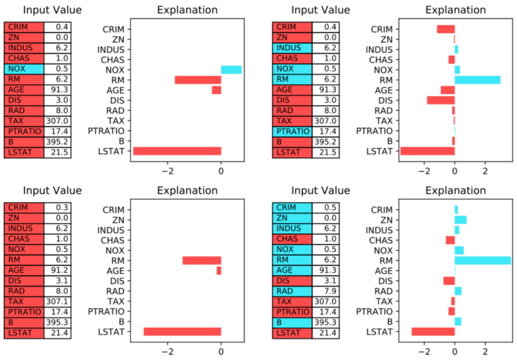
Lundberg Et. al recognition and unification of a class of additive feature attribution methods is a promising development in the field of explainable AI. The feature attribution values produced by this unifying theory seem to be more consistent with human intuition than those produced by previous methods. Additionally, the computation speed ups enabled by model-specific approximation methods may enhance the explainability of deeply complex models in the future. SHAP has provided a sound framework that enables future improvements in the explainable AI field.

Problems with Perturbation Based Explainable AI Methods

Perturbation based explainable AI methods such as SHAP and LIME rely on generating data points via small feature perturbations such that the newly generated data point remains in an epsilon neighborhood of the original. Alvarez-Melis Et. al and Slack Et. al have identified two major flaws that compromise the integrity of perturbation based explanation methods. [17]

Robustness

The first flaw is the lack of robustness. Some goals of explainable AI methods are to increase human understanding and trust in machine learning models. Explainable AI methods that fail to place any guarantees on the level of robustness of its outputs are fundamentally failing at achieving those goals. Lundberg Et. al propose that they have included a locality parameter to mitigate the unintended consequence of sample noise entirely changing the outputs of LIME's explanation models. [16] However, as dimensionality of the input space and decision space of the model increases the locality of a single data point can contain points with differing labels.



(a) SHAP ($L = 6.98$) (b) LIME ($L = 18.17$)

Figure 17. Top: example x_i from the Boston dataset and its explanations (attributions). Bottom: explanations for the maximizer of the Lipschitz estimate $L(x_i)$ as per (1).

Figure 17 shows the SHAP and LIME feature explanations for two neighboring examples from the BOSTON dataset. [15] Feature explanations such as the ones present in figure 3 would not appear to be trustworthy to an independent human auditor. The current perturbation based methodology of explainable AI that is reliant on mapping complex decision surfaces down onto lower dimensional surfaces inherently produces outputs that are highly susceptible to a lack of robustness. Future work in the field of explainable AI must pivot to new frameworks that can bypass this issue.

Perturbation Based Adversarial Attacks

Another inherent flaw of perturbation based explanation methods is that they require generating new data points that are possibly even out of distribution from the underlying distribution of the original sample. Slack Et. al show that an adversary can construct classifiers that exploit the post hoc nature of explainable AI methods such as LIME and SHAP.[17] The cornerstone of their method relies on the construction of an accurate intermediary classifier that can identify out of distribution data points generated by the perturbations.

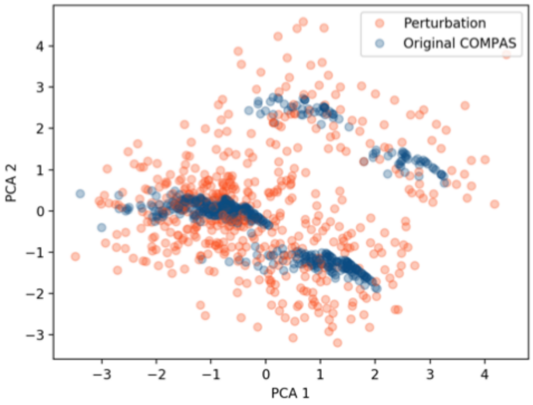


Figure 18. PCA applied to the COMPAS dataset (blue) as well as its LIME style perturbations (red). Even in this lowdimensional space, we can see that data points generated via perturbations are distributed very differently from instances in the COMPAS data. In this paper, we exploit this difference to craft adversarial classifiers.

Figure 18 shows that PCA conducted on a dataset and its corresponding perturbed points reveals clear source demarcations. Upon identification of the origination of the data point it is either sent to an unbiased classifier if it was generated OOD or to an intentionally biased classifier if it is an in-sample data point. Using these tactics the adversary is able to generate post-hoc feature attribution values from two different classifiers. A well crafted adversarial attack is able to appear both understandable and trustworthy to an auditor of the model even though the behavior of the underlying model is egregiously harmful. Slack Et. al were able to create an adversarial attack against both LIME and SHAP with a model trained on a German credit dataset.[17] The intentionally biased model was deciding behind the scenes solely on the gender of the credit applicant while another model was deciding on other arbitrary features whether or not the application was a good borrower.

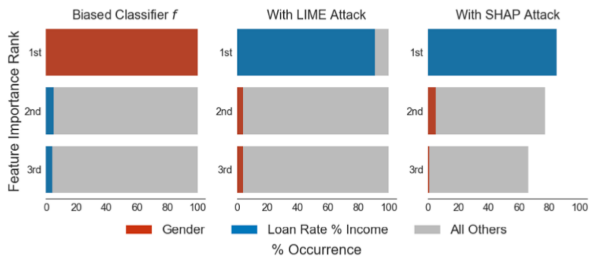


Figure 19. An unbiased classifier ψ uses an existing feature (Loan Rate % Income) to make predictions, and Gender is the sensitive feature. Feature importances for the biased classifier f shown in the figure (left) are generated using LIME; SHAP also produces similar feature importance values.

Figure 19 shows that Slack Et. al were able to devise classifiers such that the unbiased classifiers using OOD samples did not reveal the explicit bias trained into the model. It is important to note that Slack Et. al were not scoring these models on any sort of metric other than deception of the reviewers of the explanations. [17]

Perturbation based feature attribution explanation methods, especially those conducted post-hoc can place no guarantee on the validity or accuracy of their produced explanations. When such methods break down due to lack of robustness or can be gamed to be arbitrarily biased, questions about human trustworthiness, interpretability and true understanding need to be raised. Extensive documentation and open sourcing of machine learning processes, especially decision making processes, are some ways to combat these flaws. Currently, perturbation based explainable AI methods cannot be expected to produce reliable explanations of proprietary black box models without a thorough audit process.

Conclusion

As the popularity of black box machine learning models soars, the explainability of such models must keep pace. Each year, additional communities and stakeholders are affected implicitly and explicitly by black box models and must be able to understand the relationship between models' inputs and outputs. Human trust in machine learning is of paramount importance when considering real-world deployments of models. Models can be biased based on their training data or tendencies to pick features not representative of nature's distributions. Methods such as class balancing and feature importance plots help combat these glaring, news-sparking issues. Newer methodologies such as LIME and SHAP can also be utilized during model validation as a sanity checks to surface the inclusion of obvious problematic features. Models can also drift from their initial purposes over time as data change. Concept drift, if not monitored, can lead to vastly inaccurate if not harmful predictions on recent data. Drift detection techniques such as applying distance functions between distributions of consecutive windows can save models from becoming antiquated. Additive feature attribution methods may also be used to identify feature drift. As features change over time, explanation models in theory are able to capture the change in impact on the model output. For these reasons machine learning practitioners should consider adding additive feature attribution methods to their model development process.

On the other hand, explainable AI feature attribution methods are still not perfect. They provide new ways to gain insight into the behavior of complex models, but these insights are not without their flaws. Adding a layer of interpretability by simplifying complex models into explanation models is not a trivial task. Future work in the field may include better methodologies of aligning local and global understanding of predictions, better integration of feature attribution methods with machine learning models, and more improved robustness capabilities of feature attribution methods.

References

[1] Laurie A. Harris. *Artificial Intelligence: Background, Selected Issues, and Policy Considerations*. Congressional Research Service, 2021.

[2] ODSC Open Data Science. Ai black box horror stories — when transparency was needed more than ever, Oct 2019.

[3] Julia Angwin, Jeff Larson, Lauren Kirchner, and Surya Mattu. Machine bias, May 2016.

[4] Ayushi Raj Bhatt, Rajkumar Vaghashiya, Meghna Kulkarni, and Dr Prakash Kamaraj. Explainable artificial intelligence in retinal imaging for the detection of systemic diseases, 2022.

[5] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. Gender bias in contextualized word embeddings, 2019.

[6] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2013.

[7] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings, 2016.

[8] Nishant Jain and Pradeep Shenoy. Learning on nonstationary data with re-weighting, 2022.

[9] Jean-Thomas Baillargeon, Hélène Cossette, and Luc Lamontagne. Preventing rnn from using sequence length as a feature, 2022.

[10] Zhehu Yuan, Yinqi Sun, and Dennis Shasha. Forgetful forests: high performance learning data structures for streaming data under concept drift, 2022.

[11] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2018.

[12] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. *Proceedings 2004 VLDB Conference*, page 180–191, 2004.

[13] Lorena Poenaru-Olaru, Luis Cruz, Arie van Deursen, and Jan S. Rellermeyer. Are concept drift detectors reliable alarming systems? – a comparative study, 2022.

[14] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier, 2016.

[15] David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods, 2018.

[16] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017.

[17] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. How can we fool LIME and shap? adversarial attacks on post hoc explanation methods. *CoRR*, abs/1911.02508, 2019.