Unified Explanations in Machine Learning Models: A Perturbation Approach

Abstract

A high-velocity paradigm shift towards Explainable Artificial Intelligence (XAI) has emerged in recent years. Highly complex Machine Learning (ML) models have flourished in many tasks of intelligence, and the questions have started to shift away from traditional metrics of validity towards something deeper: What is this model telling me about my data, and how is it arriving at these conclusions?

Previous work has uncovered predictive models generating explanations contrasting domain experts, or excessively exploiting bias in data that renders a model useless in highly-regulated settings. These inconsistencies between XAI and modeling techniques can have the undesirable effect of casting doubt upon the efficacy of these explainability approaches. To address these problems, we propose a systematic, perturbation-based analysis against a popular, model-agnostic method in XAI, SHapley Additive exPlanations (Shap). We devise algorithms to generate relative feature importance in settings of dynamic inference amongst a suite of popular machine learning and deep learning methods, and metrics that allow us to quantify how well explanations generated under the static case hold. We propose a taxonomy for feature importance methodology, measure alignment, and observe quantifiable similarity amongst explanation models across several datasets.

1. Introduction

A long-standing problem in predictive analytics has been the disconnect between modelers (statisticians, mathematicians, and data scientists) at the model development stage and end-users at the organizational and decision-maker levels. The latter group is whom the models are ostensibly built for in the first place, but typically does not have the analytical background necessary for a full comprehension of the resulting model artifacts whereas the former community may lack the requisite knowledge of the problem domain driving the requirements of decision-makers. This "cultural divide" too frequently has the undesirable consequence of diminishing the utility of modeling to its intended audience.

This communication problem has migrated into the arena of machine learning (ML) and artificial intelligence (AI) in recent times, giving rise to the need for and subsequent emergence of Explainable AI (XAI). XAI has arisen from growing discontent with "black box" models, often in the form of neural networks and other emergent, dynamic models (e.g., agent-based simulation, genetic algorithms) that generate outcomes lacking in transparency. This has also been studied through the lens of general machine learning, where classic methods also face an interpretability crisis for high dimensional inputs [1]. Applications such as facial recognition have been met with stern resistance as, too often, mistaken identifications have led to unnecessary and serious disruption in individuals' lives [2, 3, 4]. As another example, the pre-existing bias in historic data-sets used for building ML algorithms (MLAs) has resulted in some people or marginalized communities having firsthand experience with algorithmic unfairness [5, 6]. This is a very sensitive issue for companies for whom the public perception of policy fairness and impartiality is critical to their business and well-being, and has paved the way for increased research interest in algorithmic fairness and equality.

A recent well-received book by John Kay and Mervyn King, "Radical Uncertainty: Decision-Making Beyond the Numbers" [7] highlight similar problems for economic models. The authors suggest a need for reference narratives, which are stories that can be marshaled to address the overriding objective of unraveling "what's going on here?" We adopt this perspective as our long-term strategy for model explainability.

Model transparency is and will continue to be a growing area of interest as ML/AI models

continue to develop, and organizations and users will continue to demand improved accountability. Translating mathematical and data science expertise into decision-making expertise remains a significant obstacle in gaining organizational acceptance of model artifacts. We believe that advances in model explainability and interpretation are essential to bridge this gap.

We make the following contributions here:

1. **Algorithm for Dynamic Feature Perturbation:** We introduce algorithm 1 for dynamic perturbation of a testing set akin to evasion attacks in adversarial ML literature. The algorithm performs, iteratively, continuous and categorical feature perturbation and measures sensitivity on the model's output.

2. **Metric Derivation:** We formulate and adapt two distance-based metrics to systematically quantify relative feature importance under our proposed algorithms.

3. **Evaluation and Comparison:** We compare the similarities between a well-known XAI technique in Shap [8] to our proposed method and analyze harmony and/or disharmony between the static and dynamic case.

2. Background

2.1. Preliminaries

While some work in XAI focuses on predictive model explainability on the static part of the process, we focus on contributions to both the static and dynamic scenarios. We define these terms under the following taxonomy:

- *Static Scenarios:* Given static, partitioned training and testing sets $\in \quad X, Y$, identify feature importances (FIs) using Shap, LIME, relative entropy, model weights, log odds, etc. Under the static case, we generate FIs that allow us to understand the decision boundaries being drawn under the model fitting process.

- *Dynamic Scenarios:* Under prediction scenarios, the effect, or sensitivity, of the model's generalizability when instances of the testing set are artificially perturbed $X^{test'}$ such that they are likely to be out-of-sample against the data the model was fitted against, e.g., $f(\theta, X^{test'})$.

There is a likeness to alternative nomenclature used in the field of Adversarial Machine Learning (AML), where a similar taxonomy is proposed by [9]. Our framework could be seen as being loosely analogous with evasion attacks under the AML setting, under the guise of sensitivity analysis. *Evasion attacks* are adversarial attacks where the underlying attack occurs after the specified learning algorithm is fully trained, e.g., the architecture and the learnable parameters are fixed and immutable.

2.2. Related Work

Previous work [10] explored the taxonomy noted above to analyze how well measures of feature importance hold up under 'what-if' perturbations. We extend this work, not in breadth, but depth, offering a framework to systematically quantify the similarity between the two.

Within the literature on XAI, densely populated in recent years, are methods derived explicitly to counter the black-box nature of Deep Neural Networks [11, 12, 13], particularly Convolutional Neural Networks (CNNs) used for computer vision applications. [8, 14] are seen as more generalized means of attributing explanations to model predictions, but LIME focuses on the case of local, linear approximations and SHapley Additive exPlanations (Shap) [1]offers a more comprehensive effort towards exploring local and global model explanations. Our focus in this paper is on Shap, which follows from its longstanding impact in literature and applications, the attractiveness of the properties of additive feature attributions, as well as the user studies that noted consistency between model explanations and human explanations.

Shapley Regression Values (SRGs), EQ 1, generate FIs in the presence of multicollinearity. Generating SRGs is an iterative, expensive process that involves training a model on every possible combination of feature subsets and measuring the overall effect of the model of occluding a specific feature [15]. ϕ is the output SRGs as a weighted average of differences between all such possible subsets of features $S \subseteq F$.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \cdot \left[f_{S \cup \{i\}} \left(x_{S \cup \{i\}} \right) - f_S \left(x_S \right) \right] \tag{1}$$

A turn from its original use case in cooperative game theory, where marginal contributions, or losses, are distributed between coalitions, there is a naturalness to applying it to predictive models, where input

[1]https://github.com/slundberg/shap

feature subsets are considered as coalitions and the Shapley value is the contribution (gain, loss) against the explanatory power of the model [16]. We show an example of Shap usage for global explanations in Figure 1, where we adopt the term 'feature importance graph' for end users. We partly rely on this formulation for dynamic perturbation, but do not emphasize the presence, or lack thereof, of features, instead focusing on their magnitudinal effect given a previously trained model f.

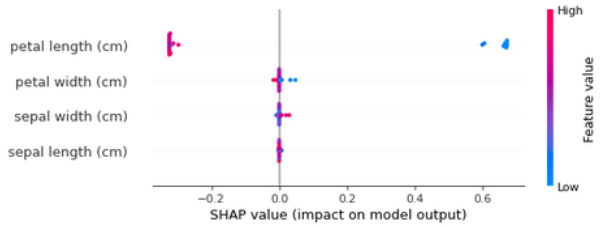

Figure 1: Shap can be used to generate local or global explanations about each label $y_i \in Y$. The global explanation for the 0^{th} class of the Iris Flower Dataset is shown above. The graph shows us that when looking at predictions, globally, made for the $0th$ class, our model is generally not using petal width, sepal width, or sepal length to draw a decision boundary. Petal length SRGs are more dispersed and are generally showing that a high value is more indicative of a lower $p(y_i = 0)$, and a lower value has a higher influence on mapping it to the 0^{th} class $p(y_i = 1)$.

3. Methodology

We note our algorithm for dynamic perturbation in algorithm 1 in the Appendix. We let $X \in R^{nxm}$ denote a matrix where $X_{i,j}$ represents the jth feature value of the ith datapoint in X. Let the cardinality of the column dimension be represented by $|m|$, e.g., let $X(j \in |m|)$ represent the jth column of X. $p_i \in P$ is a perturbation parameter.

The Continuous Case: Given a trained model f, a continuous feature j, and a perturbation parameter p_i, apply the perturbation parameter to X^{test}: $X^{test'} = X^{test}(j) * p_i$, to arrive at a perturbed testing set. Perform inference on $X^{test'}$ using the original f to generate predictions: $\hat{y} = f(X^{test'})$. Apply *any* metric m to measure predictions against actuals $m(\hat{y}, y)$

The Categorical Case: The Categorical case has the same underlying mechanics, but p_i is no longer a multiplicative scaling factor. In this case p_i is a factor influencing the presence of a categorical variable. $|X(j)| = n$, and $|X(j = 1)|$ is the length of the set where the categorical feature is active. Applying the perturbation parameter p_i to X^{test}: $X^{test'} = X^{test}(j) * p_i$, we scale the presence of data observations with an activated (boolean) categorical feature. When $p_i = 2$ we have doubled the size of $|X(j = 1)|$. We make the assumption that all categorical features are boolean, as even ordinal variables can be transformed into one-hot representations with binary responses.

3.1. Absolute Normalized Shap

To compute the Relative Feature Importance (RFIs) for each feature $j \in |m|$ given Shap values, we propose measuring each feature's absolute contribution to the absolute total contribution of all Shap values for a datapoint's correct class. Formally, we represent this as:

$$S(j) = \frac{|\ \text{shap}\ (X(j))\ |}{\sum_j^m |\text{shap}\,(X(j))|} \tag{2}$$

3.2. Absolute Normalized Weighted Average (ANWA)

Let EQ 3 represent the weighted average given a feature $x_i \in X$ and a scaling factor (perturbation parameter) $p_i \in P$, where $P \in (0, 2)$ to negate the effect of vanishing weights. Let $f(X^{test'}, P)$ be the ensuing model output from applying the scaling factor p_i to the feature column $X(j)$.

$$W\,(j, P) = \frac{\sum_i^{|P|} w_i \cdot f\,(X(j), p_i)}{\sum_i^{|P|} w_i} \tag{3}$$

The term under the summation in the numerator $w_i = (1 - |1 - p_i|)$ places more weight on perturbations closer to the unperturbed base case, i.e., when $p_i = 0$, we have our base case $f(X^{test})$. We present a working example of this in Table 1.

p_i	$w_(p_i)$	$F(X')$	Contribution
0.01	0.01	0.5	0.005
0.5	0.5	0.83	0.42
1	1	0.96	0.96
1.5	0.5	0.7	0.35
1.99	0.01	0.36	0.004
		ANWA	**0.863**

Table 1: The table can be read as follows: Column p_i represents the perturbation parameter, $w(p_i)$ is the weight we place on the predictions on the dataset perturbed by p_i, $F(X')$ is the models output (given a metric) on the perturbed dataset, and the ANWA is in the lower right cell as a linear combination of the previous two columns. More weight is placed on predictions where X' is closer to the original, unperturbed dataset X.

Let $u(\cdot) = |f(X^{test}) - f(X^{test'}, P)|$ represent the absolute difference between the base case and the weighted average W induced from perturbations $p_i \in P$

on x_i. We compute the relative importance of feature x_i on the model's output (accuracy/precision/recall/f1) as

$$I\left(x_i, P\right) = \frac{u\left(x_i, P\right)}{\sum_i^n u\left(x_i, P\right)} \tag{4}$$

This outputs the contribution of a model's performance on a perturbed dataset given p_i against the total contribution generated via $\forall p \in P$, or the absolute difference between the base case and the weighted average over perturbing a feature $|P|$ times, over the sum of absolute differences from repeating this for $X(j), j \in |m|$.

3.3. Metric Comparison

To quantify similarity between RFI vectors associated with the outputs from EQs 2 and 3, we leverage two popular distance metrics used in literature for a variety of tasks. The proposed usage of varying distance metrics helps to shape the underlying reference narrative of the model artifacts, where some end users may want to drill down on the full explanation model, and others may side with *exclusively* focusing on the most important features.

Cosine Similarity Cosine similarity is a distance metric bounded between $[0, 1]$, where a cosine value of 0 means that two vectors are orthogonal to one another and have no intrinsic similarity, while a cosine value approaching 1 indicates a greater likeness between vectors. In our case, a higher cosine value represents a greater harmony between explanations under the proposed taxonomy of static and dynamic explainability.

$$\text{dist}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{5}$$

Let $\vec{i}$ be a vector corresponding to the RFIs computed using EQ 4, and let $\vec{j}$ be a vector containing the proportion of summed absolute Shap values against the whole population (all features).

$$\begin{aligned} \vec{i} &=< I\left(x_i, X, P\right), \ldots, I\left(x_n, X, P\right) > \\ \vec{j} &=< S(x_i), ..., S(x_n) > \end{aligned} \tag{6}$$

We superpose these vectors into EQ 5 to arrive at the cosine similarity between the RFIs generated under the static and dynamic cases.

$$\cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \mid \vec{j}\|} \tag{7}$$

Jaccard Similarity Jaccard Similarity (JS) is a popular metric when looking at top-k performance commonly used to measure efficacy in Recommender System [17, 18] or Multi-label classification [19, 20] tasks. We rationalize our inspection of similarity under JS with the assumption that the magnitude of the RFIs is not as important as their ranking in some cases, which is often the case when generating reason codes in regulated financial applications. We ask the question: "How often do FI rankings align under static and dynamic scenarios?". Like Cosine Similarity, this metric is also bounded, i.e., $0 \leq J(A, B) \leq 1$, making it an attractive option for analysis.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{8}$$

In EQ 8, we see that J measures the cardinality of the intersection of two sets against the union of the two sets. Here, we consider the two sets, $A(k)$ and $B(k)$ to be the top-k ranked RFIs given EQs 2 and 3.

An Example: Suppose we are looking at the Jaccard Similarity for the top-2 highest RFI features given an explanation model's output. Let A be the set containing the top-2 most important features under EQ 2 and let B be the top-2 ranked RFIs under EQ 3 (refer to Figure 2). We compute JS here:

$$\begin{aligned} &A(k=2) = \{\text{petal length, petal width}\} \\ &B(k=2) = \{\text{petal width, sepal length}\} \\ &J(A, B) = \tfrac{1}{2+2-1} = \tfrac{1}{3} \end{aligned} \tag{9}$$

4. Experiments

Our focus in this paper is to systematically analyze XAI under static and dynamic scenarios. Analysis of model performance is beyond the scope of this paper, and as such we utilize known classification datasets in the literature that *most* existing methods can solve. We run our experiments on a variety of linear and nonlinear classifiers, all available through the ScikitLearn API (SKlearn) [22]. We point the reader to the Table 7 in the Appendix for relevant literature discussing the origin and mathematics of each. We aim to perform the following:

1. Generate RFI under the static case using Shap and EQ 2.
2. Generate RFI under the dynamic case using algorithm 1 and EQ 3.

Dataset	Reference	Size	Ind. Features	Continuous	Categorical	Classification	Classes
Iris	[21]	150	4	X		X	3
Wine	[21]	177	13	X		X	10
Breast Cancer	[21]	568	30	X		X	2
UCI Census	[21]	48842	14	X	X	X	2
Synthetic Fraud	J.P. Morgan Chase	3430	8	X	X	X	2

Table 2: Summary of the Datasets we explore. Each falls under the bucket of a classification task, where a data point consists of an (x, y) tuple and where y is a discrete variable. NumClasses is a value corresponding to the number of discrete values in the target variable. All datasets are split 80/20 into train and test sets with randomized shuffling. Datasets with categorical features are run through a label encoder to generate one hot representations.

3. Identify the similarity between (1) and (2) using EQs 5 and 4.

4. Offer analyses around noted differences in explanations under contrasting model specifications and classification metrics, differing feature sets/types, and elasticity of similarity as a function of dataset size.

4.1. Datasets

We note several popular datasets used in Table 2. Most datasets are available via SKlearn, while the others can be sourced directly from the UCI Machine Learning Repository [21], and were chosen due to their widespread use as benchmarking datasets in ML and DL to evaluate efficacy on a variety of performance metrics, as well as the variability in independent variables ranging from continuous only to mixtures of ordinal, nominal and continuous.

4.2. Performance Metrics

Our measure to compute feature importance under dynamic scenarios requires an arbitrary performance metric m that measures $\hat{Y}^{test}$ against Y^{test} before computing the ANWA via EQ 3. Accuracy is an oft-used metric in classification tasks but falters when dealing with datasets containing class imbalance (non-uniform distribution of the target variable), which resembles most real-world problems. We use Accuracy, Precision, Recall and F1-Measure throughout our analyses, defining and formalizing these in Table 3.

Metric	Definition	Notation
Accuracy	Prop. of true preds amongst all preds	$\alpha = \frac{TP+TN}{TP+TN+FP+FN}$
Precision	Prop. of true positive preds amongst all positive preds	$p = \frac{TP}{TP+FP}$
Recall	Prop. of true positive preds amongst all actual positive instances	$r = \frac{TP}{TP+FN}$
F1 Measure	Harmonic Mean of Precision and Recall	$f1 = 2 \cdot \frac{p \cdot r}{p+r}$

Table 3: Commonly used metrics when dealing with a classification task $f : x \rightarrow y$, where y is a discrete target variable pertaining to a particular class. In the cases where the number of target variables exceeds 2, a weighted metric is required. The Iris and the Wine: datasets noted above require weighted, macro or micro averages for Precision, Recall and F1 measure calculations.

4.3. Shap vs ANWA - A Drill Down

Due to page limitations, we omit a complete, detailed drill-down on the measured RFIs for Shap against our methodology. We do, however, include a sample comparison displaying RFIs for a set of models on the Iris and Cancer datasets.

What we see from these feature importance graphs (Iris), measuring the static (Shap) and dynamic (perturbation-based) scenarios, is the following:

- Shap deems the feature 'sepal length' to be negligible in its effect on the models' ability to partition the classes. Most of the predictive power comes from 'petal length' and 'petal width'.
- Using our algorithm 1 to perturb testing instances and compute RFIs that way, we show that 'sepal length' has a more recognizable influence on the model's outputs, which can be construed as contrasting explanations. This is especially noticeable as we see the respective values drastically shift for the shallow neural network, where under the static case the global explanations emphasized petal length as being important, and under testing conditions, petal length was shown to have *no* effect at all on the sensitivity of the model ('sepal length' took most of this difference in contributions).

We notice this as a recurring theme across our analyses of other datasets, where the explanations shift under the two scenarios, which raises questions about the trustworthiness of the model artifacts, particularly within highly regulated industries. If the

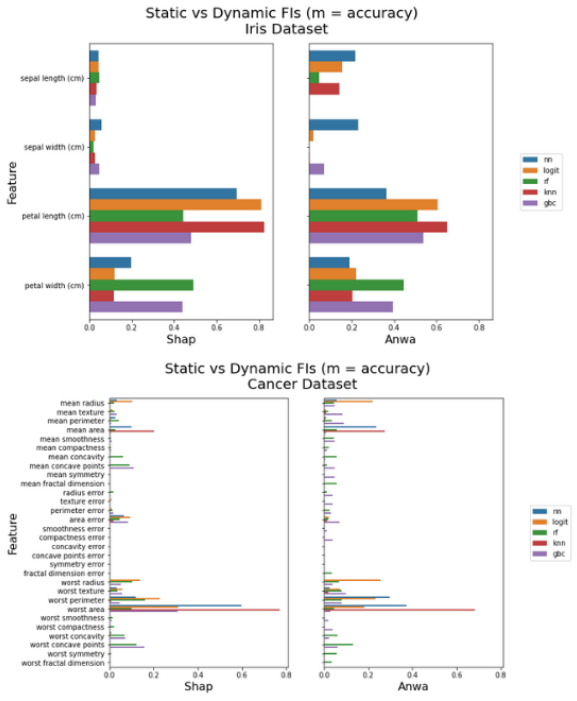

*Figure 2: Analysis conducted on the Iris **(TOP)** and Cancer **(Bottom)** datasets [21]. **Left:** Absolute Normalized Shap Values for each feature. **Right:** Absolute Normalized Weighted Averages for each feature following our methodology for dynamic perturbation. We fix the arbitrary metric per EQ 3 as m = accuracy for simplification. The legend denotes abbreviations that follow the mapping: **'nn'** : Multilayer Perceptron Neural Network, **'svm'** : Support Vector Machine Classifier, **'logit'** : Logistic Regression, **'rf'** : Random Forest Classifier, **'knn'** : K-Nearest Neighbors Classifier, **'gbc'** : Gradient Boosted Trees Classifier.*

models are not interpretable or harmonized in their interpretations under each scenario, there is the potential for unintentional, systematic bias propagating through the decision-making process. We also include feature importance graphs for the Cancer dataset in Figure 2, where it becomes easier to notice how challenging a task creating 'reference narratives' can be in high dimensional spaces.

4.4. Systematic Comparison

While model artifact drill-downs may be necessary in some cases, our goal is to systematically quantify the similarity that we highlighted in the case is presented in 4.3. We dissect these results in Table 4, where we show the similarity induced using various classification metrics on our algorithm across a suite of predictive models.

In general, we notice that the choice of performance metric m that we use to compute the feature importance via Eq 3 has a marginal impact on the outcome, except in a few cases. Across all datasets, the average similarity across all metrics range from [0.79, 0.84] with a standard deviation between [0.15, 0.22]. This variation reduces drastically if we remove the results from the Census dataset ($\mu = 0.88$, $\sigma = 0.13$).

The three datasets where the explanations align well under both static and dynamic forecasting are low dimensional, and singular in their independent variable types (continuous). The Adult Census dataset is relatively high dimensional and features a large number of boolean response variables after one-hot encoding. In that case, the RFIs for the continuous features are less emphasized and more diverse, i.e., when there are more explanatory variables, there is a lesser amount of unanimous consensus. This would raise concern in a deployed setting, particularly for fraud detection, where the feature space is large and varies in datatypes. We note that this is not a problem isolated to Deep Neural Networks, as it appears in similarity inconsistency across all four of the nonlinear estimators. The logistic regression model displays the highest average similarity, as well as the lowest variance. These results are further visualized in Figure 3, and speak to a consensus between the two scenarios.

4.5. Sample Size Analysis

We are also interested in how elastic similarity between Shap and ANWA is, as a function of sample size. In other words, we want to see how well explanations align when models are allotted varying input sizes. For this, we isolate the Census dataset, as it has the largest total number of samples, and we fix $m = accuracy$ to be our metric of choice for computing the RFIs under a dynamic scenario. We set the size of the full dataset to be $|X| \in \{1000, 2000, 4000, 8000, 16000, 32000\}$ and follow the same cleaning, splitting, and perturbation protocols as the previous step.

Table 5 shows results from this experiment. We notice a trend towards unified explanations (in cosine similarity between RFIs) as the size of the sample pool grows, but variation in the similarities is too profound to assume a purely linear relationship ($R^2 = 0.58$). While increasing the size of the dataset is likely to have a corresponding effect on the model's performance, we do not see enough evidence to support that it implies more similar explanations. More work is required to understand the relationship between the two, as well as the performance-interpretability trade-off that comes when imposing likeness as a constraint on the training process.

	NN				Logit				RF				KNN				GBC			
	α	p	r	f	α	p	r	f	α	p	r	f	α	p	r	f	α	p	r	f
Adult	.51	.49	.51	.49	.80	.55	.83	.55	.58	.51	.59	.49	.90	.26	.90	.34	.69	.80	.77	.63
Cancer	.85	.92	.85	.86	.88	.82	.88	.87	.80	.72	.80	.80	.99	.98	.99	1.00	.50	.41	.50	.50
Iris	.83	.90	.83	.85	.96	.99	.96	.97	.99	1.00	.99	.99	.97	1.00	.97	.98	.99	.97	.99	.99
Wine	.86	.91	.86	.87	.83	.94	.83	.85	.86	.90	.86	.89	.94	.88	.94	.95	.97	.96	.97	.96

Table 4: For each of the models we experiment with , we generate a cosine similarity measure (EQ 5) between the RFIs generated via Shap and by ANWA. This table shows the results when using accuracy, precision, recall or F1 measure as an input m into Algorithm 1. $sim \rightarrow 1$ means the explanations are similar, and as $sim \rightarrow 0$ there exists disharmony between the static and dynamic cases.

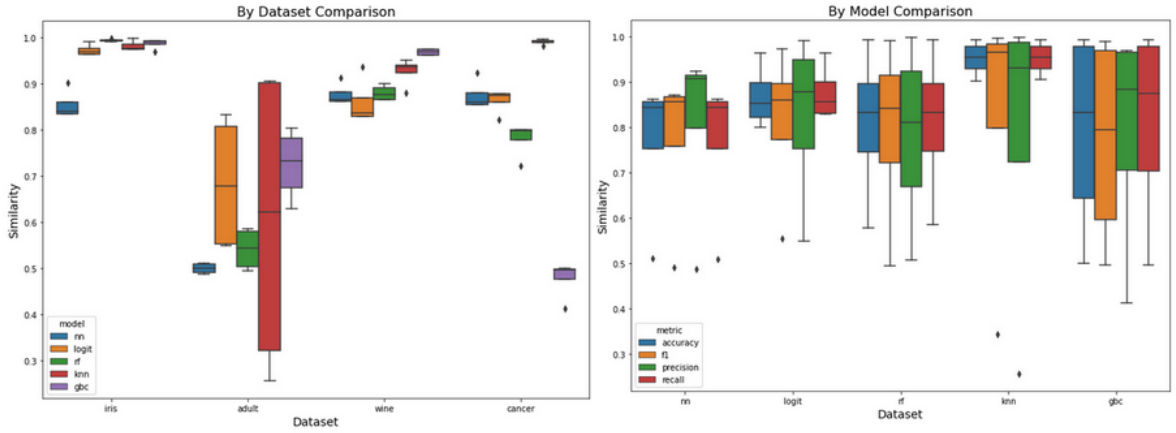

*Figure 3: **Left:** Boxplot displaying cosine similarity between Shap and ANWA by dataset. The colors map to specific predictive modeling techniques. **Right:** Boxplot displaying similarity between Shap and ANWA across all models, with the color mapping to the specific performance metric m used.*

$\lvert X \rvert$	1k	2k	4k	8k	16k	32k
Similarity	0.70	0.56	0.49	0.74	0.86	0.87

Table 5: Subsample analysis on UCI Census Data using a Neural Network and $m = accuracy$.

4.6. Ranked Similarity

Following from EQ 8, we aim to quantify how often the top-k features resultant from the two explanation methods align. In highly regulated areas, like finance, we may not be as concerned with uniform harmony under dynamic forecasting scenarios, but rather with alignment amongst top features that the model identifies as being important.

Using Shap as our baseline, we compute the rank of each feature in the feature set by absolute normalized Shap value, and compare them to the rankings pursuant from our method, ANWA, with varying $k \in K$, where k is the size of the slice over the sorted set.

Iris only contains four features, so we exclude that dataset from this analysis. We see a general monotonic trend across all estimators, for all datasets. At the

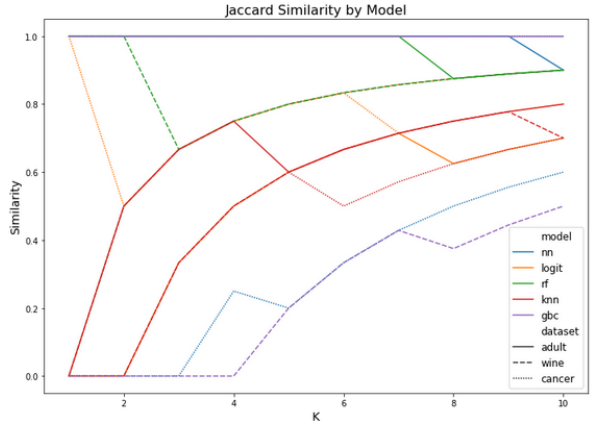

Figure 4: Jaccard Similarity by Model, by Dataset. The similarity is measured by the intersection of two sets against their union. The two sets contain the top-k features for each method under static and dynamic XAI.

$k = 10$ step, all models have a Jaccard Similarity greater than 0.5 ($\mu = 0.84$, $\sigma = 0.16$). While a large k can artificially inflate the similarity measure as $k \rightarrow |m|$, we find the general trend to be true for the

datasets with a high feature dimension (Breast Cancer and UCI Census). Results are visualized in Fig 4, and pseudocode for the heuristic we use to generate this measure is included in algorithm 2 in the Appendix.

4.7. Explainability in the Wild

To this point, our focus has been exclusively on datasets where trained estimators achieve moderate to high levels of accuracy without tuning. We turn our attention to a synthetic dataset provided by J.P. Morgan. The details of the dataset can be found in Table 2, with the underlying scope of the task being to classify fraudulent transactions from genuine ones (all data cleaning notes can be found in the Appendix). We note as a general case that the testing suite of models outperform a majority vote baseline (50%) by $\sim 20\%$ without hyperparameter tuning, whereas the full dataset had an accuracy baseline of 98%. The Cosine Similarity between the two explanation vectors, along with the model accuracy on the testing set, are shown in Table 6.

Model	Accuracy	Cos Similarity
Neural Network	0.58	0.43
Logistic Regression	0.6	0.9
Random Forest	0.65	0.88
K-Nearest Neighbors	0.58	0.99
Gradient Boosting	0.66	0.97

Table 6: Model Accuracy and Cosine Similarity between explanation techniques on the J.P. Morgan synthetic dataset.

While the models are not fully tuned, there are notable similarities to the SRGs reported from Shap against our method, which does not require the full retraining of models on the $S \subseteq F$ possible feature subsets. The lone outlier is in the case of the Deep Neural Network, which is under-powered by limited amount of positive (fraudulent) examples in the dataset. Using JS as a proxy for the harmony of explanations, we note that a lower Cosine Similarity between explanation vectors does not necessarily imply low congruence. For $k \in \{1, \ldots, 10\}$ we see the mean JS, $\frac{1}{|k|}\sum_i^{|k|} J(k)$, near 80% ($\sigma = 0.14$) with $k \leq \frac{1}{4}|X(j)|$. Disharmony between the explanation vectors can be subjective based on the intended use-case of understanding the model artifacts. While using JS may allow us to understand the most important features, a model that exploits $\frac{1}{|X(j)|}$ of the feature space for more than half of its predictive power (as was the case for Transaction_Amount to detect fraud here) should be monitored, or further tuned.

We believe that these experiments work to further validate the explanation models resultant from the Shap attribution generation process by showing a consensus between methods. Further, we believe this to be a novel framework for dissecting general predictive models and their ensuing explanations, with a lesser computational burden than experienced with generating SRGs.

5. Conclusion

XAI offers a paradigm shift towards interpretability and explainability required in many fields utilizing ML and AI. We have introduced a taxonomy classifying two unique instances of XAI, 'dynamic' and 'static' cases [10], formulated harmony as a measure of the distance between explanations of these cases (Cosine and Jaccard Similarity), and employed a perturbation-based algorithm 1 to systematically quantify it on several models, metrics, and datasets.

We show, in general, a moderate to a high level of consensus among methodological views, one looking towards attribution values generated from the training data, and one which looks towards the testing data to validate it on *potentially* out of sample datapoints. Our proposed framework *begins* to shine a light on questions sparsely asked in XAI, such as "How well does the explanation model hold up in production?", and "Can I trust this explanation model in a deployed setting, without unintentionally amplifying bias and unfairness?". Our proposed method is also flexible, intuitive, and easy to implement.

We believe that this work can be catalyzed to answering those questions, and towards facilitating the generation of reference narratives as a way to answer the question "what's going on here?" in any particular model and decision-making setting. Also, we believe this work to be essential to the overall model lifecycle, where checks-and-balances are needed to show shortcomings of over-exposed bias from the model artifacts.

5.1. Future Work

XAI is very much an open research problem, stemming from widespread industrial adoption of sophisticated algorithms, and the importance for them to pass tests concerning bias, fairness, and transparency. In the future, we intend to explore the framework that we have specified here for datasets under highly regulated industries, like finance, where mitigating bias in explanations is of the utmost importance. We believe the ideas expressed here could be relevant to non-tabular, unstructured data, where it could be a useful addition to the growing literature on XAI in its quest to dispel explanation uncertainty in tasks like Natural Language Processing, Computer Vision, and Graph

Machine Learning.

References

[1] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[2] C. Garvie and J. Frankle, "Facial-recognition software might have a racial bias problem," *The Atlantic*, vol. 7, 2016.

[3] K. Miller, "A matter of perspective: Discrimination, bias, and inequality in ai," in *Legal Regulations, Implications, and Issues Surrounding Digital Data*, pp. 182–202, IGI Global, 2020.

[4] M. Georgopoulos, Y. Panagakis, and M. Pantic, "Investigating bias in deep face analysis: The kanface dataset and empirical study," *Image and Vision Computing*, vol. 102, p. 103954, 2020.

[5] K. Johnson, F. Pasquale, and J. Chapman, "Artificial intelligence, machine learning, and bias in finance: toward responsible innovation," *Fordham L. Rev.*, vol. 88, p. 499, 2019.

[6] S. Corbett-Davies and S. Goel, "The measure and mismeasure of fairness: A critical review of fair machine learning," *arXiv preprint arXiv:1808.00023*, 2018.

[7] S. Arduin, "John kay and mervyn king, radical uncertainty: Decision-making for an unknowable future, london: The bridge street press, 2020, 528 pp, hb, £25. 00," *The Modern Law Review*, vol. n/a, no. n/a.

[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[9] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, pp. 151–178, 2020.

[10] D. Dolk, D. Kridel, J. Dineen, and D. Castillo, "Model interpretation and explainability towards creating transparency in prediction models," in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

[11] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*, pp. 3319–3328, PMLR, 2017.

[12] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2017.

[13] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *International Conference on Artificial Neural Networks*, pp. 63–71, Springer, 2016.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[15] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, pp. 319–330, 2001.

[16] B. Iooss and C. Prieur, "Shapley effects for sensitivity analysis with correlated inputs: comparisons with sobol'indices, numerical estimation and applications," *International Journal for Uncertainty Quantification*, vol. 9, no. 5, 2019.

[17] M. Y. H. Al-Shamri, "Power coefficient as a similarity measure for memory-based collaborative recommender systems," *Expert Systems with Applications*, vol. 41, no. 13, p. 5680–5688, 2014.

[18] M. Ayub, M. A. Ghazanfar, M. Maqsood, and A. Saleem, "A jaccard base similarity measure to improve performance of cf based recommender systems," in *2018 International Conference on Information Networking (ICOIN)*, pp. 1–6, IEEE, 2018.

[19] H. Gouk, B. Pfahringer, and M. Cree, "Learning distance metrics for multi-label classification," in *Asian Conference on Machine Learning*, pp. 318–333, PMLR, 2016.

[20] E. Montanes, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz, and E. Hüllermeier, "Dependent binary relevance models for multi-label classification," *Pattern Recognition*, vol. 47, no. 3, pp. 1494–1508, 2014.

[21] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[22] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[23] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," *American Journal of Psychology*, vol. 76, p. 705, 1963.

[24] P. D. Wasserman and T. Schwartz, "Neural networks. ii. what are they and why is everybody so interested in them now?," *IEEE Expert*, vol. 3, no. 1, pp. 10–15, 1988.

[25] W. S. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[26] L. Wang, *Support vector machines: theory and applications*, vol. 177. Springer Science & Business Media, 2005.

[27] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The journal of educational research*, vol. 96, no. 1, pp. 3–14, 2002.

[28] R. E. Wright, "Logistic regression.," 1995.

[29] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[30] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[31] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics*, vol. 7, p. 21, 2013.

6. Appendix

6.1. Algorithm

We note pseudocode for our dynamic perturbation algorithm here and include code written in python linked to our repository. Additionally, we note pseudocode for our means of comparing static and dynamic RFIs using Jaccard Similarity.

Algorithm 1: Perturbation(f, m, X, Y, j, P)

W = weighted average
m = performance metric
$X \in R^{nxm}$
$Y \in R^{n}$
$p_i \in P$
$j \in |m|$
$(X^{train}, X^{test}, Y^{train}, Y^{test}) \in X, Y$
$X^{Cont}, X^{Cat} \in X^{test}$
$W = 0$
$f \leftarrow f(X^{train}, Y^{train})$
for $p_i \in P$ **do**
  //Outer loop
  for $j \in |m|$ **do**
    //Inner loop
    //Perturb jth feature by p_i
    if $x \in X^{Cont}$:
      $X^{test'}(j, p_i) = X^{test}(j) * p_i$
    if $x \in X^{Cat}$:
      if $p_i \geq 1$:
    //Activate inactive observations for feature x
      $X^{test'}(j, p_i) = X^{test}(j = 1) * (2 - p_i)$
    //Deactivate active observations for feature x
      else:
      $X^{test'}(j, p_i) = X^{test}(j = 1) * (1 - p_i)$
    //apply trained model f
    $\hat{y} = f(X^{test'})$
    //generate performance of model
    $w_i = (1 - |1 - p_i|)$
    $W(j) += \frac{w_i \cdot m(\hat{y}, y)}{\sum_i^P w_i}$
  End for
End for
Return: Weighted average W for perturbing feature j for each $p_i \in P$

Algorithm 2: Jaccard($dataset, model, k$)

k =slice range
svs = generate_shap_values(·)
anwa = generate_anwa_values(·)
top_k_shap = rank(sort(svs))[:k]
top_k_anwa = rank(sort(anwa))[:k]
intersection = $top\_k\_shap \cap top\_k\_anwa$
union = $top\_k\_shap \cup top\_k\_anwa$
Jaccard = $\frac{intersection}{union}$
Return Jaccard

Model Name	Abbrev.	Refs
Deep Neural Network	nn	[23, 24]
Support Vector Machine	svm	[25, 26]
Logistic Regression	logit	[27, 28]
K-Nearest Neighbors	knn	[29]
Random Forest	rf	[30]
Gradient Boosted Trees	gbc	[31]

Table 7: Abbreviation to Model map used in this paper. References to seminal work, or surveys, on each can be found here. All models were trained, validated and tested using the Sklearn API, and all data cleaning, splitting, and modification followed suit.

6.2. Models

We include some cursory information on each of the predictive models used in our experiments in Table 7. Models were **not** trained with extensive hyperparameter tuning aimed at driving marginal performance improvements. SVM was excluded from our analysis as Shap value generation was a bottleneck on the hyperplane-deriving algorithm. Specific configurations can be found in our source code (available later).

6.3. Datasets

In general, there was little in the way of data cleaning. The J.P. Morgan Fraud dataset, being a real-world, synthetic set, was downsampled to have a uniform class distribution to deal with imbalance, and certain nominal features with little predictive power were omitted to limit exploding dimensionality. The transaction timestamp feature was parsed and hour, day of week, and month were used as categorical features. Iris, Wine, and Cancer are all continuous datasets. UCI Census contains categorical variables, and those were preprocessed using pandas dataframe operations. Each categorical variable was transformed into a one-hot boolean response vector. All datasets were split into train and test sets using the Sklearn API, and generally using an 80-20 split, except for specific analysis. These datasets were randomly shuffled before splitting.

6.4. Shap

Our work is dependent on Shap [8] to derive baseline RFIs. RFIs in the static case can be dependent on parameter configurations. As computing Shap values over a full dataset with high dimensionality (in the case of binary expansion) can be expensive, we summarize the training/reference dataset using the shap.kmeans call. For reproducibility, a link to our code is included in our final version of this paper (after review).