A Trustworthy View on XAI Method Evaluation

Ding Li $^{1,1}$, Yan Liu $^2$, Jun Huang $^2$, and Zerui Wang $^2$

$^1$Concordia University
$^2$Affiliation not available

October 31, 2023

Abstract

As the demand grows to develop end-user trust in AI models, practitioners start to build and configure customized XAI (Explainable Artificial Intelligence) methods. The challenge is the lack of systematic evaluation of the newly proposed XAI method. As a result, it limits the confidence of XAI explanation in practice. In this paper, we follow a process of XAI method development and define two metrics in terms of consistency and efficiency in guiding the evaluation of trustworthy explanations. We demonstrate the development of a new XAI method in feature interactions called Mean-Centroid Preddiff, which analyzes and explains the feature importance order by a clustering algorithm. Following the process, we perform cross-validation on Mean-Centroid Preddiff with existing XAI methods. They show comparable consistency and gain in computation efficiency. The practice helps to adopt the core activities in the trustworthy evaluation of a new XAI method with rigorous cross-validation on consistency and efficiency.
A Trustworthy View on XAI Method Evaluation

Ding Li, Yan Liu, Jun Huang, Zerui Wang
Department of Electrical and Computer Engineering, Concordia University, Montreal, CA

**Abstract** As the demand grows to develop responsible AI models, practitioners begin to configure customized XAI (Explainable Artificial Intelligence) methods to some extent for explaining the prediction. The challenge is the lack of systematic evaluation of the newly proposed XAI method. As a result, it limits the confidence in XAI explanation in practice. In this paper, we follow a process of XAI method development and define two metrics in terms of consistency and efficiency in guiding the evaluation of XAI explanations. We demonstrate the development of a new XAI method in feature interactions called Mean-Centroid Preddiff, which analyzes and explains the feature contribution in terms of the distribution density of prediction changes. Following the process, we perform cross-validation on Mean-Centroid Preddiff with existing XAI methods. They show consistency in explanation summaries and gain in computation efficiency. The practice highlights the core activities and metrics applicable to developing and evaluating XAI methods.

**Introduction**

Explainable AI (XAI) practices perform an interpretive analysis of the input data. They produce an approximation summary as an explanation relating inputs and outputs without engaging the internal representations, attributes, and structures of the learning models [1]. XAI is emerging as a common goal for data scientists, engineers and AI practitioners to deal with the problem of AI opacity on the purpose, server and how they work. Critically, Babic et al. [2] have highlighted the need for an explainable method to be trusted in the healthcare domain. In particular, a trustworthy XAI method should exhibit some robustness, which means the XAI method should ordinarily produce similar explanations for similar inputs [2].

The discrepancy in the explanation summary leads to ambiguity in understanding the machine learning prediction. This becomes the question of Are the explanation from different XAI methods trustworthy? Therefore, the consistency of these summaries becomes essential for the trustworthy and accountable assessment of machine learning models.

One source of variation of explanation summary originates from the XAI operations. The major entities involved in XAI operations are data sets, a trained model and XAI methods. An XAI method can be applied to the model prediction on each data sample. As a result, a set of explanation summaries are collected for the same prediction model. One XAI method may demonstrate different levels of similarity among the explanation of each data sample. In addition, multiple XAI methods may yield explanations in variation from each other on the same data set and the model. We describe one example scenario of the XAI explanation variation in the case of code vulnerability analysis in section Observing Explanation Consistency.

A conventional approach is to compare several XAI methods’ explanation summaries and make decisions among the available results based on how consistent the explanation summaries are. If the decision cannot be reached with majority
voting, developing a new XAI algorithm becomes necessary. A proposed XAI algorithm needs to define new metrics that measure the feature contributions in perspective not fully addressed by state-of-the-art XAI methods.

In addition to explanation summary consistency, state-of-the-art XAI methods vary in an extensive range of run-time delays inherently due to their intrinsic algorithms of computing the features’ contribution. One target of a new XAI algorithm is to reduce the time complexity with a comparable consistency level to state-of-the-art XAI methods. Coherently, the problem of evaluating existing XAI methods and developing an XAI alternative converges to the question as What are the criteria and core activities in evaluating XAI methods towards consistent and robust explanation?

In this article, we demonstrate the XAI engineering on evaluating different XAI methods with well-defined metrics, namely consistency and run-time efficiency. We further define the measure of consistency using the distances among explanation summaries. The run-time efficiency is based on asymptotic analysis in terms of the number of features and the size of data samples. We present two evaluation techniques, (1) comparing with a baseline model and (2) performing cross-validation among multiple models. Based on the consistency evaluation of state-of-the-art XAI methods, we develop a new model-agnostic method for the XAI taxonomy called Mean-Centroid Preddiff. Together with the other nine XAI methods, we evaluate the consistency and time efficiency of Mean-Centroid Preddiff on three domain examples, including image classification, code vulnerability detection and search-based ranking. We demonstrate a working path of systematically evaluating and making decisions on an XAI method under the trustworthy view.

Related Work

Explainable AI (XAI) is an emerging research topic in recent years, aiming to explain AI models’ logic and decision-making processes for users in the goodness of safety and fairness. Conventionally, post-hoc XAI methods are categorized as model-agnostic and model-specific [1]. Model-specific methods probe and extract the model gradients or neuron activation states from the neural network models. Examples are the family of XAI methods based on Class Activation Mapping (CAM) [3] including EigenCAM [4], GradCAMElementWise [5], Grad-CAM++ [6], XGrad-CAM [7], and HiResCAM [8]. They have been applied to explain feature contributions to image classification algorithms and tasks. Existing XAI work has applications in various domains. Related to the case studies in this article, the work [9] introduces model-agnostic LIME and SHAP to compare ranking models. The study [9] defines completeness and validity measurements of ranking models.


The robustness and consistency of XAI outputs are essential for adopting XAI to enhance the trust and accountability of artificial intelligence [15]. The trust can be based on an individual prediction by users. The XAI explanation summary related to this level of trust means the robustness of the explanation for each data sample prediction. Based on the prediction trust, a user may accept the trust at the whole model level. Accordingly, the consistency across multiple XAI methods on the same model relates to the accountability of AI models. At both levels, existing work [16] highlights quantified metrics that have been defined to measure XAI explanation results toward the progress of developing trustworthy artificial intelligence.

Observing Explanation Consistency

We introduce an observation of explanation consistency across multiple XAI methods. In software code vulnerability detection, understanding how code features affect the accuracy of vulnerable code classification helps reduce the vulnerability risks [17] and enables automated corrections on vulnerable code [18]. We have trained an XLNet model [19] as a classifier to classify vulnerable software code at the method level to different CWE (Common Weakness Enumer-
ation) types. The feature masking is configured in three kinds: namely (1) code-only, program code without comments and import statements; (2) comment-only, and (3) import-only, with only import statements. An XAI method explains the feature importance of code, comment, and import statement in terms of their contributions to code vulnerability classification. We eliminate the CWE label tokens in the code comments to avoid the training of the machine learning model may "remember" the CWE labels.

Three model-agnostic XAI methods are applied to the above feature masking scenarios, namely Prediff [10], Shapley Value [14] and KernelSHAP [13]. Our results show that both Shapley and KernelSHAP rank the feature importance in descending order as comment, code, and import statement on public data sets of Juliet1 and OWASP2. However, Prediff ranks in the order of code, import, and comment on the OWASP data set. We observe the difference in the explanation summary among the XAI methods and data sets. Furthermore, we measure the time consumption of running three XAI methods against the same XLNet model on two data sets. Shapley value and KernelSHAP consume approximately three times and twenty times as Prediff, respectively. The complete results are available in github3.

In this case, one can select one of the three XAI methods as the baseline model to further conclude on the XAI explanation consistency. One option is selecting the Shapley value as the baseline. Shapley values are initially created to assign attributions to specific participants in coalition games. Shapley values have been adopted for explaining to the machine learning models since it has the properties of efficiency, symmetry, dummy and linearities [13]. Alternatively, a new XAI method can be developed to cross-validate the existing explanation summary.

**Explanation on Features (Sidebar)**

XAI methods that derive explanation by features include masking-based methods and mutation-based methods. Feature masking-based methods remove certain features or set the features with default values [20]. Then the output prediction is evaluated. On the other hand, mutation-based methods assign possible input values to the model and then obtain prediction [21]. The feature influence is measured by inputting the black-box models with feature masking or mutation. These methods then measure prediction changes compared to the original model and inputs. These methods vary from each other in terms of (1) feature masking or mutation techniques and (2) summary techniques to compute the feature importance [20]. Other XAI methods, such as explain-by-visualization, apply digital patterns, plots, or heatmaps to explain the feature classification and localisation [22].

**Evaluate XAI Explanations**

The above discussion motivates a unified process of evaluating the existing XAI methods or guiding the development of a new XAI method. Both activities share the core activities as presented in Figure 1.

The process starts with the task of setting XAI goals and criteria. Survey works [1], [20] provide taxonomy and classification to select candidate XAI methods that target the same goals. Based on selected XAI methods, the next step is to define the measurement of trustworthy attributes. In this article, we focus on the consistency of the explanation summary under feature masking or feature removal.

**Define Consistency for Feature-based XAI Explanation**

Consider \( \hat{f}(x[i]) \) is the model prediction on instance \( x[i] = \langle x_1[i], x_2[i], ..., x_p[i] > \), where \( p \) is the number of features. Suppose \( S \) is the subset of all the features by masking or removing a feature \( j \) that is \( S \subseteq \{1, 2, 3, ..., p\} \setminus \{j\} \) and \( P \) contains the whole features, \( P = S \cup \{j\} \).

Under feature masking, the prediction on the masked feature set \( S \) and on the whole feature set \( P \) for each instance \( x \) has the difference as \( \delta^X_j = \hat{f}_S(x[i]) - \hat{f}_P(x[i]). \) Hence the feature contribution to the payout by masking feature \( j \) on the prediction of instance \( x[i] \) is defined as a function as \( \phi_j(\delta^X_j) \). An XAI method develops the aggregation of \( \phi_j(\delta^X_j) \) on all the data samples differently. Finally, by masking the features one by one, the feature importance order is derived

---

1 https://samate.nist.gov/SARD/test-suites/111

2 https://owasp.org/www-project-benchmark/

3 https://github.com/DataCentricClassificationofSmartCity/Mean_Centroid-Preddiff
by ranking the feature contribution values.

After the transformation from feature contribution values to the feature importance order, Kendall Tau Ranking Distance \cite{23} is applied to measure the distance of any two pairs of the XAI method’s explanation results.

**Analyze Time Complexity**

Asymptotic analysis for $\phi_j(\delta^X_j)$ depends on the size of data instances number $N$ and the number of features $P$. Shapley value computes the feature value difference under feature masking $\delta^X_j$ for the whole data set for each masked feature. Shapley value considers the permutation when selecting one feature to mask and makes the reverse value of permutation as the weight to sum the feature contribution value $\phi_j$. Overall, we derive that the Shapley value has the complexity as $\Theta(N \times P \times 2^P)$. KernelSHAP \cite{13} uses the linear LIME explanation model and the classical Shapley value. According to the definition, KernelSHAP depends on the LIME loss function \cite{12}, weighting Kernel and the regularization term. Therefore, Kernel SHAP has the complexity of $\Theta(N \times (2^P + P^5))$. Preddiff removes each feature individually and measures the difference between each instance’s prediction and the feature removal prediction. The time complexity of Preddiff is $\Theta(N \times P)$. In section Develop a New XAI Method, we present a newly proposed XAI method with the time complexity of $\Theta(N \times P^2)$.

**Collect Explanation Summary**

The evaluation of the XAI explanation summary begins with the configuration of the test data set into several versions, including the whole features and each subset of masked features. The machine learning model runs on the configured test data set to output $\delta_j^{x^{(i)}}$ the prediction. Next, each XAI method computes the feature contribution $\phi_j(\delta^X_j)$ by aggregating all the data samples.

**Measure Explanation Consistency**

The evaluation process in Figure 1 provides a guideline concentrating on the consistency of the explanation summary as a primary attribute to decide the selection of the XAI method or the development of a new XAI method. The computing complexity of an XAI method is another additional attribute of decision-making. At the condition checking point, the XAI practitioners decide to use the existing XAI method or build a new one. In both cases, the consistency definition enables a unified understanding of XAI methods in three phases, including (1) computing the difference of prediction under feature configuration, (2) computing the feature contribution value based on the prediction difference, and (3) converting the contribution values to the explanation. A distance metric such as Kendall Tau Ranking Distance is applied to measure the distance between two explanation summaries. The larger the distance value, the less consistent the two explanations are.

**Develop a New XAI Method**

We aim to explain the effects of feature masking by the relative difference in the ratio to the prediction without feature masking. The state-of-the-art methods consider the absolute prediction difference. The objective of this new XAI method should achieve comparable consistency of explanation summary to the state-of-the-art methods and reduce computing time consumption. Figure 2 shows the core tasks of computing the
prediction difference under feature masking and feature contribution value for each masked feature in three phases.

**Phase 1: Compute Prediction Difference under Feature Masking.**

Algorithm 1 presents that the prediction difference $\delta_j^{(i)}$ (as x-coordinate) and its corresponding prediction $\hat{f}_p(x^{(i)})$ (as y-coordinate) form a data point in two dimensional Euclid plane. Hence, $N$ numbers of two-dimensional points are created for each masking feature $j$.

**Phase 2: Compute Feature Contribution Values.**

We observe from Phase 1 output that the data points form clusters. We further group the data points into $k_j$ numbers of clusters by agglomerative clustering algorithm [24]. We then estimate the centroid data point of these clusters using the Gaussian mixture model [25]. For each masked feature $j$, we define its feature contribution value $\phi_j$ aggregated for all the input data samples as the slope or tangent of the centroid data point to the origin point in a two-dimensional plane. An example in Figure 3 depicts how the Gaussian mixture clusters aggregate the contribution values of two feature markings. Data points are grouped into two clusters for each feature. The centroid data point is derived as the weighted average of the clusters’ density points generated by the Gaussian Mixture model. This algorithm has considered the distribution density of the prediction changes of the whole data samples.

**Phase 3: Convert to Feature Importance Order**

The conversion is simply ranking the features in ascending order according to their feature contribution value. The consistency of the two explanations is then measured as the distance between two orders.

**Asymptotic Analysis on Time Complexity**

Given the number of features $P$ and the number of instances $N$, computing the prediction difference is of the time complexity $\Theta(N \times P)$ in phase one. In phase two, computing the clusters takes $\Theta(N \times P^2)$. Overall, the time complexity is $\Theta(N \times P^2)$.

**Algorithm 1** Mean-Centroid Prediction Difference (Preddiff) Explanation

- $f_{agg}$, agglomerative clustering algorithm [24]
- $f_{gmm}$, gaussian mixture model [25]
- $k_j$, the number of clusters under feature masking $j$
- $\text{Centroid}$ as cluster centroid point
- $S \subseteq \{1, 2, 3, \ldots, p\} \setminus \{j\}$, the subset of all the features by masking or removing a feature $j$
- $P$, the whole features, $P = S \cup \{j\}$

**Input:** Input data set $X$, full feature set $P$, masking feature set $S$, model prediction $\hat{f}(x^{[i]})$

```plaintext
/* Phase 1: compute the difference of prediction under feature configuration*/
for all $j \in P$ do
  for all $x_i \in X$ do
    $\delta_j^{(i)} \leftarrow |\hat{f}_g(x^{[i]}) - \hat{f}_p(x^{[i]})|$
    $\nu_j^{(i)} \leftarrow <\delta_j^{(i)}, \hat{f}_p(x^{[i]})>$
  end for
end for

/* Phase 2: compute the feature contribution value*/
/* group $V_j$ to $k_j$ clusters */
k_j \leftarrow f_{agg}(V_j)
/* derive the centroid of $k_j$ clusters */
centroid_j \leftarrow f_{gmm}(k_j, V_j)
/* compute the contribution value as tangent of the centroid data point in two dimensional coordinates */
$\phi_j(\delta_j^X) = \tanh(\text{centroid}_j)$

/* Phase 3: convert the contribution values to the feature importance orders */
order = sort(abs($\phi_j(\delta_j^X)$))
```

**Output:** $\phi_j(\delta_j^X)$, order

**Case Study on Image Classification**

The first case study evaluates the explanation summaries of the image classification on face mask detection. The Mean-Centroid Preddiff method is cross-validated with six state-of-the-art model-specific XAI methods. The open-source pre-trained ResNet50 [26] is applied to detect the face mask categories from images.
Figure 2: The dataflow of Mean-Centroid Prediction Difference explanation summary

Figure 3: An example of deriving two features’ contribution values by Gaussian Mixture clusters.


Applying Mean-Centroid Preddiff to Image Explanation

As illustrated in Figure 4, we generate a kernel masking matrix to mask the pixels iteratively by filling in zeros. We then obtain \((l \times l)/(n \times n)\) masked images for the model prediction where the image size is \((l \times l)\) and the kernel masking matrix has a size \((n \times n)\). The Mean-Centroid Preddiff summarizes the pixel feature contributions from the prediction difference between the original image and the masked ones. In the experiment, the \(l\) is 256, and we take kernel masking matrix size \(n = 8\).

Explanation Evaluation Analysis

Six XAI methods are selected for explaining the saliency map of the input images, including Grad-CAM [27], EigenCAM [4], GradCAMElementWise [5], Grad-CAM++ [6], XGrad-CAM [7], and HiResCAM [8]. The saliency map explanation shows the active area of the image that contributes to the model’s prediction.

Consistency Observation. CAM-based methods compare the prediction change between the original image and the masked image by saliency map. Mean-Centroid Preddiff summarizes the prediction change from kernel-based image masking.

Figure 5 displays the prediction change distance distribution of 2,630 images. EigenCAM has the longest distributed range compared to other methods. This indicates that EigenCAM varies the most in explaining the feature contributions of 2,630 images. In contrast, the Mean-Centroid Preddiff plot has the lowest range. This shows that the Mean-Centroid Preddiff method is more consistent across all the images.

Time Complexity Analysis. Mean-Centroid Preddiff has the time complexity of \(\Theta(N \times P^2)\), given the number of images \(N\) and the number of features \(P\). In this case, an image with a masking kernel matrix is counted as one feature. Hence \(P = (l \times l)/(n \times n)\) is the number of features.

[^4]: https://github.com/youyinnn/ai_face_mask_detection_project.git
Case Study on Code Vulnerability Detection

Referring back to the discussion in section Observing Explanation Consistency, we observe that the explanation produces different feature importance order from three state-of-the-art methods. We re-evaluate the XAI methods by adding the Mean-Centroid Preddiff method. Table 1 shows that Preddiff and Mean-Centroid Preddiff have the same feature importance order. In both values, the code is the most important feature. The importance order of comment and import statement varies from the Juliet and OWASP data sets. Shapley value and KernelSHAP share consistent results but value comment more than code and import statement. Security experts can make further decisions on XAI methods based on the above explanation.

Figure 4: The process of Mean-Centroid Preddiff on Image Explanation

Figure 5: Explanation consistency between Mean-Centroid Preddiff and other CAM-based methods. The green line indicates the mean value.

Case Study on Scholar Searching Ranking System

This case study attempts to explain the feature influence of an open-source semantic scholar search ranking (S2Search) model [28]. S2Search provides a prediction tool to output a ranking score for each scholarly article given a query keyword and a list of features. The arXiv data set selected from Kaggle\(^5\) has forty categories and a total of 542,877 articles. An article contains six relevant features title, authors, abstracts, citation numbers, venue, and published year. Each category is performed as a single data set.

Cross Validation of Consistency

**Across Data Sets Comparison.** The median contribution values sort the feature importance order of an XAI method from forty data sets. We measure the KTRD distance between this aggregated feature importance order with the orders of forty data sets as the across data sets comparison. Figure 6a shows the median value of KTRD distances across data sets. It demonstrates that Mean-Centroid Preddiff, Shapley Value and KernelSHAP are more consistent than Preddiff.

**Across XAI Methods Comparison.** The baseline method is selected in rotation out of the four methods. Figure 6b plots the 50th percentile of KTRD distances. In summary, Mean-Centroid Preddiff is more consistent than Preddiff but less than the other two methods.

Computing Time Consumption

Figure 7 shows the time consumption curve increases as the number of data samples grows. Preddiff and Mean-Centroid Preddiff are more time-efficient than KernelSHAP and Shapley.

\(^5\)https://www.kaggle.com/data sets/Cornell-University/arxiv
### Table 1: Feature importance order summary of code vulnerability detection case study

<table>
<thead>
<tr>
<th>XAI Methods</th>
<th>Juliet test cases</th>
<th>OWASP test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preddiff</td>
<td>comment &gt; code &gt; import</td>
<td>code &gt; import &gt; comment</td>
</tr>
<tr>
<td>Mean-Centroid Preddiff</td>
<td>comment &gt; code &gt; import</td>
<td>code &gt; import &gt; comment</td>
</tr>
<tr>
<td>Shapley value</td>
<td>comment &gt; code &gt; import</td>
<td>comment &gt; code &gt; import</td>
</tr>
<tr>
<td>KernelSHAP</td>
<td>comment &gt; code &gt; import</td>
<td>comment &gt; code &gt; import</td>
</tr>
</tbody>
</table>

Value. Mean-Centroid Preddiff spends approximately 10% more time than the Preddiff due to the clustering computation.

### Conclusion

This article discusses the trustworthy view of XAI methods by defining consistency and efficiency metrics. Two metrics, consistency and time efficiency, provide a trade-off view to evaluate XAI methods. In the case that higher consistency and faster time efficiency cannot be achieved simultaneously, users are left to prioritize the metrics for decision-making. Through case studies, we observe that state-of-the-art XAI methods may produce explanation summaries that vary at the data set level and across methods. Hence motivates, this work to develop a unified evaluation method that helps to assess the explanation consistency of existing XAI methods, as well as guides the development of a new XAI method. This evaluation method is the base for constructing the service pipeline of XAI operations.

### REFERENCES


15. K. de Bie, A. Lucic, and H. Haned, “To trust or not
Figure 6: Consistency comparison across data sets and XAI methods. A shorter link edge indicates a consistenter XAI method. a: Across data sets. b: Across XAI methods.

Figure 7: Time consumption between XAI methods along with the data set size increasing in scholar searching rank system case study to trust a regressor: Estimating and explaining trustworthiness of regression predictions, "arXiv preprint arXiv:2104.06982, 2021.


