BONES: a Benchmark fOr Neural Estimation of Shapley values

BONES: a Benchmark fOr Neural Estimation of Shapley values

1st Davide Napolitano
Department of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
davide.napolitano@polito.it

2nd Luca Cagliero
Department of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
luca.cagliero@polito.it

***Abstract*—Shapley Values are concepts established for eXplainable AI. They are used to explain black-box predictive models by quantifying the features' contributions to the model's outcomes. Since computing the exact Shapley Values is known to be computationally intractable on real-world datasets, neural estimators have emerged as alternative, more scalable approaches to get approximated Shapley Values estimates. However, experiments with neural estimators are currently hard to replicate as algorithm implementations, explainer evaluators and results visualizations are neither standardized nor promptly usable. To bridge this gap, we present BONES, a new benchmark focused on neural estimation of Shapley Value. It provides researchers with a suite of state-of-the-art neural and traditional estimators, a set of commonly used benchmark datasets, ad hoc modules for training black-box models, as well as specific functions to easily compute the most popular evaluation metrics and visualize results. The purpose is to simplify XAI model usage, evaluation, and comparison. In this paper, we showcase BONES results and visualizations for XAI model benchmarking on both tabular and image data. The open-source library is available at the following link: https://github.com/DavideNapolitano/BONES.**

***Index Terms*—Explainable AI, Shapley Values, Neural Shapley Values Estimation, Benchmarking**

I. Introduction

EXplainable Artificial Intelligence (XAI) aims to make AI models more transparent to end-users [1]. Given a black-box predictive model, XAI solutions focus on providing explanations for the decision-making process. Shapley Values (SVs) [2] are concepts rooted in cooperative game theory, which have become established for XAI. SVs provide end-users with a deep understanding of each feature's contribution to the model's prediction, thereby enhancing the interpretability and trustworthiness of complex predictors.

The exact computation of SVs from real-world data is known to be computationally intractable [3] as the number of feature combinations is exponential with the dimensionality of the input dataset. Hence, a number of heuristic methods (e.g., KernelSHAP [4]) have been proposed to generate approximated SV estimates.

The increasing availability of GPU-equipped hardware and the evolution of Deep Learning techniques has fostered the study of Neural Network-based approaches to compute approximated SVs. During the training process, these networks learn the functional mapping between the input data features and their SVs attributions. State-of-the-art neural approaches (e.g., FastSHAP [5]) are currently able to efficiently generate accurate SVs estimates.

As a drawback, neural SVs estimators are currently neither promptly accessible nor easy to use. Actually, the most popular XAI projects (e.g., Quantus [6], OpenXAI [7], ComparexAI [8]) lack neural solutions. Furthermore, there is a lack of standardization in model testing, evaluation, and comparison. This limits the applicability of neural approaches compared to more popular approximated methods such as Monte Carlo sampling [9] and regression techniques [10].

We present **BONES**, a **B**enchmark f**O**r **N**eural **E**stimation of **S**hapley values, aimed to foster XAI applications that mainly rely on neural SVs estimators. BONES consists of

- A suite of Shapley Values estimators, mainly neural and some traditional, tightly integrated for easy comparison and use;
- A set of benchmark datasets, both in tabular form and images, that are commonly used for XAI model benchmarking;
- Ad hoc modules to train black-box models and generate reliable ground truth SVs, whenever not already available, either exact or approximated.
- A set of testing functions implementing the most popular performance evaluation metrics;
- A set of promptly interactive plots that can be used to visually explore the main results and compare models' performance with each other.

BONES simplifies and expedites the use of state-of-the-art neural approaches and allows end-users to perform accurate model comparisons considering aspects such as computational efficiency, attribution accuracy, model robustness to data cardinality and dimensionality. We hope BONES could effectively support XAI researchers interested in exploring the strengths and limitations of neural solutions.

II. Related Works

A. XAI tools

As discussed in [11], [12], the attention of the research community to the eXplainable AI (XAI) field is ever-increasing. To actively support related research activities, several XAI

benchmarks and tools have been released, e.g., SHAP [4], Quantus [6], OpenXAI [7], Compare-xAI [8], and Ferret [13]. The purpose is to allow fair and transparent comparisons among different XAI methods by making available suites of state-of-the-art algorithms, datasets, evaluation metrics, and visualization techniques. However, existing libraries do not incorporate the latest neural approaches. Thus, comparing neural Shapley Value estimators with each other or with traditional approaches requires additional effort. BONES addresses the above limitation by providing researchers with promptly usable implementations of state-of-the-art Shapley Values estimators, both neural and traditional, as well as a testing suite including benchmark tabular datasets, evaluation metrics, and visualization tools. Our solution welcomes future extensions towards the integration of new algorithms, datasets, and standardized evaluation procedures.

B. XAI models

Understanding how AI models make decisions is crucial for augmenting their transparency and interpretability, especially because most AI predictors act as black boxes. Feature importance attribution measures how much each input feature contributes to a model's predictions. Among existing methods, Shapley Values are popular due to their solid mathematical basis, as they fairly distribute the model's output among the input features based on their contributions and interactions. However, computing Shapley Values is often impractical. To address this issue, researchers have developed methods to approximate Shapley Values and, as alternatives, other techniques to compute feature relevance, like permutation importance [14], LIME [15], and DeepLIFT [16]. Empirical studies have compared various Shapley Values approximation methods, highlighting the trade-offs between accuracy, computational efficiency, and robustness. Existing analysis [17] provides a comprehensive evaluation of different Shapley Value approximation methods, showing that while neural approaches can significantly reduce computation time, their approximation accuracy varies depending upon the model architecture and dataset characteristics.

a) Traditional Approaches: Classical methods to compute Shapley Values involve exact computation [4], Monte Carlo sampling [9], and regression techniques [4], [10]. Exact computation evaluates the model on all possible subsets of features, providing precise Shapley Values but at a computationally prohibitive cost. This approach is not applicable to models trained on many features due to the combinatorial explosion in the number of candidate subsets. Monte Carlo sampling methods approximate Shapley Values by averaging over random subsets of features, reducing computational burden but often requiring a large number of sampling iterations to achieve accurate results. Regression techniques, such as KernelSHAP [4] and Unbiased KernelSHAP [10], are also used to approximate Shapley Values using linear regressions, allowing for improved computational efficiency.

b) Neural Approaches: Although traditional methods are accurate, they often have computational problems when scaling up the dataset size, making them impractical, especially at inference time. Regarding existing neural approaches, DeepExplainer is part of the SHAP library [4]. It consists of an enhanced version of DeepLift [16], which recursively attributes the difference in the model's output between each input sample and the corresponding background sample back to the input features, significantly improving the computational efficiency over traditional methods. GradientExplainer leverages integrated gradient-based attributions [18] with SHAP values, utilizing the gradients of the output with respect to the inputs to approximate feature contributions more efficiently. FastSHAP [5] employs a neural network to learn a mapping from model inputs to Shapley Values, reducing the computation time, especially on large datasets, by approximating the complex Shapley Value calculations through a learned function. DASP (Differentiable Approximation of Shapley values) [19] introduces a polynomial-time algorithm that leverages neural network architectures to approximate Shapley Values, enhancing the scalability and efficiency of the computation process. ViT-Shapley [20], designed mainly for Vision Transformers (ViTs) [21], adapts the Shapley Value computation to the unique architecture of ViTs, providing interpretable explanations for image classification tasks by learning to estimate the contribution of image patches to the model's predictions. Other techniques, such as ShapNet [22], focus on computing Shapley Values from ground truth data, making them unsuitable for explaining black-box models.

III. The BONES Benchmark

BONES is a benchmark for neural Shapley Values estimation. It consists of the following modules:

- **Black-Box Models**: it generates post-hoc explanations of arbitrary classification of various types and with various settings.
- **XAI Models**: it integrates a variety of approaches to approximated SVs estimations, both neural and not.
- **Datasets**: it provides access to several benchmark datasets, both tabular and image data.
- **Ground Truth**: it supports the computation of both exact SVs [4] and regression-based estimations [10] that can be used as alternative ground truths.
- **Evaluation Functions**: it allows quantifying the accuracy of the SVs' estimates against the ground truth and the efficiency of the estimation process, as well as comparing different models with each other.
- **Visualization**: it natively supports the generation of various plots useful to perform exploratory analysis of the models' results and of their accuracy-efficiency ratio.

The design of BONES ensures maximal usability, portability, and extendability. The key properties are summarized below.

- **Modality-Agnostic.** A core strength of our framework is its modality agnosticism by design. Shapley Values are potentially applicable across various data modalities such as image, tabular, and text data. Our framework

is designed to support a wide range of approaches and data types, ensuring its applicability in different input types domains. This broad applicability is crucial for researchers and practitioners who deal with data in different modalities and require reliable explainability standards. Currently, BONES supports tabular and image data. The extension to other modalities is already planned as a future work.

- **Post-Hoc Explanations.** Our benchmark allows end-users to explain predictions of already trained models. This aspect of the framework is particularly valuable for practical applications, where models are often trained in a production environment, and explainability needs to be retrofitted to provide insights into model behavior and decision-making processes.
- **Opensource, modular framework.** To foster collaboration, reproducibility, and extensibility, our framework is designed with an open and modular architecture. The open BONES benchmark fosters contributions from the broader research community, facilitating the integration of new methods, datasets, and evaluation metrics. Modularity ensures that components of the framework can be independently developed, tested, and replaced. This flexibility allows users to customize the framework to suit their specific needs, whether that involves incorporating new neural architectures, experimenting with alternative Shapley value estimation techniques, or adapting the benchmark to novel interpretability challenges.

In the following we detail the characteristics of the BONES components.

A. Datasets

BONES is currently suited to both tabular and image data. The benchmark is designed to facilitate the seamless integration and utilization of both proprietary and benchmark datasets such as those available in the UCI repository [23]. The current list of integrated datasets is given in Table I.

For tabular data, we choose a subset of datasets that are representative of different cardinality, dimensionality, and density distributions. For image data we include datasets covering different aspects of visual information and model explainability. In detail, we integrate ImageNette [24] and Pet [25] by adopting the same configuration as in Vit-Shapley [20].

TABLE I: Benchmark datasets.

	Dataset	Source	Train samples	Validation samples	Num. features
Tabular	Monks	UCI	302	130	6
	WBC	UCI	436	110	9
	Census	SHAP	20838	5210	12
	Credit	UCI	19200	4800	23
	Magic	UCI	12172	3044	10
Image	ImageNette	ViT-Shapley	9469	1963	224x224
	Pet	ViT-Shapley	5879	735	224x224

B. Explainers

BONES provides a comprehensive suite of SVs estimators, both neural and not. The list of currently available XAI models is reported in Table II, where column *Type* differentiates between traditional and neural estimators. We standardize the integration process to make the module easily extensible with newly proposed approaches. The implementation currently relies on TensorFlow and PyTorch for tabular data and on PyTorch for images.

For tabular data, our framework supports several approaches, including SHAP [4], i.e., the Exact, GradientSHAP, and DeepSHAP versions, ShapleyRegression [10] with Unbiased KernelSHAP and KernelSHAP, Monte Carlo Sampling [9], DASP [19][1], and FastSHAP [5]. For image data, the framework currently includes SHAP [4] (i.e., DeepSHAP and GradientSHAP variants), FastSHAP [5], and ViT-Shapley [20] for Vision Transformers.

TABLE II: Explainers

	Model	Type	Supported Framework	Black-Box Type
Tabular	Exact	Traditional	All	All
	Unbiased KS	Traditional	All	All
	KernelSHAP	Traditional	All	All
	MonteCarlo	Traditional	All	All
	DeepExplainer	Neural	PT/TF	Neural
	GradientExplainer	Neural	PT/TF	Neural
	DASP	Neural	TF 1	Neural
	FastSHAP	Neural	PT/TF	All
Image	DeepExplainer	Neural	PT/TF	Neural
	GradientExplainer	Neural	PT/TF	Neural
	FastSHAP	Neural	PT/TF	All
	ViT-Shapley	Neural	PT	ResNet/DeepNet/ViT

C. Black-Box Models

Most neural explainers are suited to explain neural Network-based models only (see Column *Black-Box type* in Table II). However, the latest approaches (e.g., FastSHAP) are compatible with non-neural classifiers as well.

As default black-box models, BONES exploits:

- For **tabular data**, a Multi-Layer Perceptron classifier with two intermediate dense layers, each containing 64 units and ReLU activation, interspersed with dropout layers. The final dense layer has an output corresponding to the number of classes, followed by a softmax activation. This implementation relies on Tensorflow;
- For **image data**, a pre-trained Vision Transformer (ViT) in its tiny version [21], followed by a linear layer corresponding to the number of classes, is used for classification.

Thanks to its modularity and extensibility, BONES straightforwardly supports the integration of traditional non-neural classifiers as well (e.g., the classifiers available in the Scikit-Learn library [26]).

[1]BONES currently integrates the authors' implementation relying on TensorFlow version 1.

D. Evaluation functions

a) Estimation error: BONES natively supports evaluation functions suited to quantify the prediction error made by a SVs estimator against a ground truth. It integrates the L1 and L2 distances. Furthermore, for tabular data only, it also supports the Kendall correlation coefficient, which evaluates the consistency in the SVs feature ranking.

b) Computational cost: To evaluate the efficiency of the XAI models, BONES keeps track of the explainers' training and inference times.

c) Comparative analysis: To compare the performance of different explainers with each other, BONES supports the following performance metric P:

$$P = 1 - \frac{d_i - d_{min}}{d_{max} - d_{min}} \tag{1}$$

where d_i is the distance metric of the $i-th$ explainer, whereas d_{min} and d_{max} are the minimum and maximum values on the same distance metric across all analyzed explainers, respectively. This metric provides a value from 0 to 1, where a higher value highlights better performances.

To compare the performance of image explainers, BONES also supports the Inclusion and Exclusion AUC (Area Under the Curve) [5]. Inclusion evaluates how well an image explainer identifies important regions by measuring the increase in the model's prediction score as these regions are progressively included. Exclusion assesses the impact of removing important regions identified by the explainer on the model's prediction score. Their combined use allows end-users to identify the best-performing image explainer, i.e., the model with maximal Inclusion and minimal Exclusion [5].

E. Visualization

BONES offers the following options to visualize the performance results of SVs estimators and to compare them with each other graphically:

- **Bar plot:** It displays the local or global per-feature SVs computed by different explainers. This visualization allows for a direct comparison of the feature importance assigned by each explainer. This visualization is mainly intended for tabular data.
- **Image plot:** For image data only, it graphically shows the mask of Shapley Values retrieved by different explainer pairs overlaid on the input processed image. In detail, a 14x14 pixels mask is used for all approaches, interpolating ones providing pixel-wise explanations.
- **AUC curves:** It plots the Inclusion and Exclusion AUC for image data only. AUC shows the predictor accuracy by varying the percentage of Inclusion/Exclusion.
- **Quadrant plot:** The quadrant plot is computed based on overall times and our performance metric P (1). It offers a comprehensive view of the computational efficiency and performance of different explainers, aiding in selecting the most suitable method for a given task.
- **Computational times vs. number of samples plot:** This plot visualizes the model's computational times by varying the number of samples processed on the chosen dataset. End-users can vary the total number of samples, the interval between the tested values, and the sampling techniques applied to the input dataset. It provides end-users with insights into the explainers' scalability of explainers with the dataset cardinality.
- **Computational times vs. number of features plot:** This plots show the inference times spent by the XAI model by varying the number of input features.

IV. Case Study

We showcase the usability and flexibility of BONES using two example case studies, one on a tabular dataset, i.e., Monks [23], and on an image dataset, i.e., ImageNette [24].

A. Tabular Data

Code 1: Example of code snippet for tabular data.

```
from bones.sv.tabular.explainers import FastSHAPModel,
↪  ShapRegModel, DASPModel, ExactModel
from bones.sv.tabular.datasets import Monks, Census
from bones.sv.tabular.metrics import L1, L2, Kendal
from bones.sv.tabular.evaluation import Banckmark
from bones.sv.tabular.display import TimeSamplePlot,
↪  TimeFeaturePlot, BarPlot, QuadrantPlot

benchmark=Benchmark(explainers=[FastSHAP, ShapReg,
↪  DASP], ground_truth=Exact, dataset=[Monks, Census,
↪  Credit], metrics=[L1, L2, Kendal],
↪  num_samples=100).run()

benchmark.print_results(Monks) # table results

TimeSamplePlot(benchmark, dataset=Monks,
↪  number_sample=100000, interval=10000,
↪  sample_method="random").plot()

TimeFeaturePlot(benchmark).plot()

BarPlot(bechmark, dataset=Monks).plot()

QuadrantPlot(benchmark, dataset=Monks).plot()
```

The snippet of Code 1 shows how to efficiently import SVs estimators and datasets, select the preferred evaluation metric, and generate the corresponding plots.

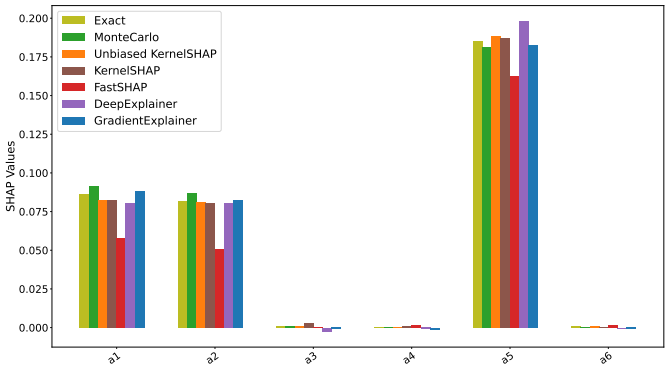

Fig. 1: Example of bar plot comparing the global Shapley Values estimated by six explainers on the Monks dataset against the ground truth (i.e., Exact).

Figure 1 shows the resulting bar plot, which compares the outcomes of several approximated SVs estimators with the Exact SVs estimates [4]. Specifically, for each explainer, the bars correspond to the Global Shapley Value relative to a specific feature computed over the whole dataset. The more similar to the Exact SVs, the better. Besides global evaluations, this visualization supports local inspection when a dataset sample is specified. Figure 3 shows the inference times by varying the number of features (upper image) and samples (lower image), respectively. Unlike all the other approaches, FastSHAP [5] has an inference time per sample negligible compared to its training time. Hence, the per-sample variation is flattened. Finally, the quadrant plot in Figure 2 allows a graphical comparison between the tested models in terms of L2 distance-Computational time ratio. In this dataset analysis, Neural approaches have shown to be consistently better than traditional models (e.g., Monte Carlo sampling [9], KernelSHAP [10]).

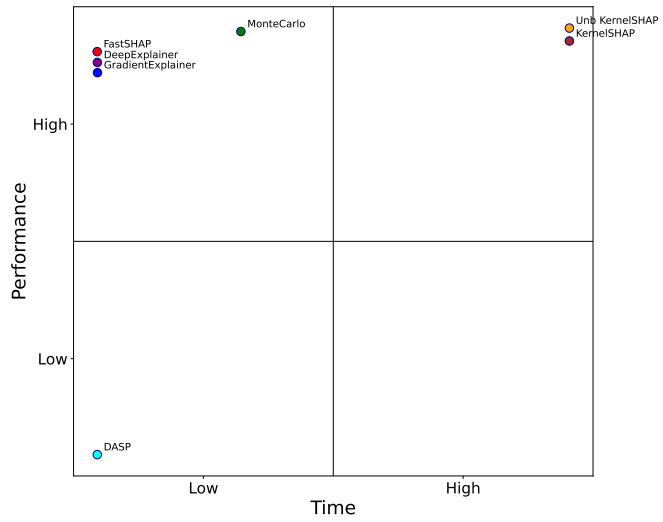

Fig. 2: Quadrant plot combining computational times and a L2 distance metric.

B. Image Data

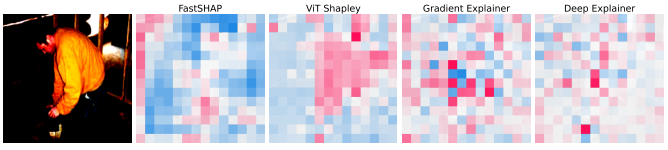

Fig. 4: Image plot: comparison of the Shapley Values' masks computed by the different explainers on an ImageNette sample.

The snippet of Code 2 shows a similar example tailored to image data. BONES offers the opportunity to visualize the Inclusion and Exclusion AUC plot (see Figure 5). For example, the experiments on ImageNette confirm the better capabilities

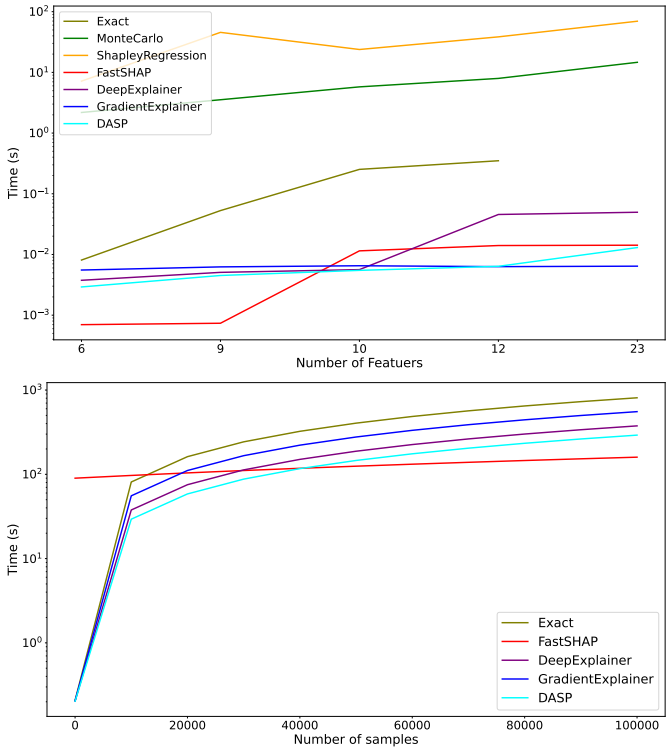

Fig. 3: Examplse of visualizations plots showing the variations of the computational times with the number of dataset features (upper plot) and the number of dataset samples (bottom).

of FastSHAP to avoid excluding discriminating regions. The Image plot in Figure 4 allows end-users to select a sample and view the masks of the Shapley Values calculated by the various methods. This is particularly interesting for users who would like to perform qualitative analysis quickly.

Code 2: Example of code snippet for image data.

```
from bones.sv.image.explainers import FastSHAP,
↪  ViTShapley, DeepExplainer, GradientExplainer
from bones.sv.image.datasets import ImageNette
from bones.sv.image.metrics import L1, L2, AUC
from bones.sv.image.evaluation import Banckmark
from bones.sv.image.display import ImagePlot, AUC

benchmark=Benchmark(explainers=[ViTShapley,
↪  DeepExplainer, GradientExplainer],
↪  ground_truth=FastSHAP, dataset=[ImageNette],
↪  metrics=[L1, L2, AUC], num_samples=100).run()

# results, TimeSample and Quadrant as for Tabular data

ImagePlot(bechmark, datset=ImageNette, sample=0).plot()

AUC(benchmark, dataset=ImageNette,
↪  num_sample=100).plot()
```

V. Conclusions and Future Work

The paper presented BONES, a benchmarking library for neural Shapley Values estimation. The main purpose is to make neural estimators available and easy-to-use to researchers of

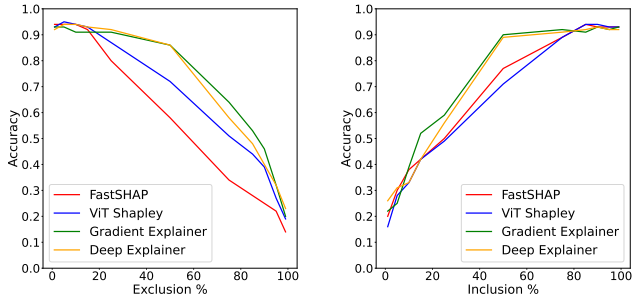

Fig. 5: AUC Exclusion (left) and Inclusion (right) computed on ImageNette.

the XAI community, leveraging applications to real-world data and quantitative models' comparisons. As future work, we plan to extend BONES to support a broader range of modalities, models, and datasets. Additionally, we aim to integrate variants of Shapley Values into BONES, including Shapley Residuals and Interval Shapley Values.

VI. Acknowledgements

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PNRR M4C2, INVESTIMENTO 1.3 D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. The research leading to these results has been partly funded by the SmartData@PoliTO center for Big Data and Machine Learning technologies.

References

[1] W. Saeed and C. Omlin, "Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities," *Knowledge-Based Systems*, vol. 263, p. 110273, 2023.

[2] L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games II*, H. W. Kuhn and A. W. Tucker, Eds. Princeton: Princeton University Press, 1953, pp. 307–317.

[3] G. V. den Broeck, A. Lykov, M. Schleich, and D. Suciu, "On the tractability of SHAP explanations," *J. Artif. Intell. Res.*, vol. 74, pp. 851–886, 2022.

[4] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.

[5] N. Jethani, M. Sudarshan, I. C. Covert, S. Lee, and R. Ranganath, "Fastshap: Real-time shapley value estimation," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[6] A. Hedström, L. Weber, D. Krakowczyk, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, and M. M.-C. Höhne, "Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond," *Journal of Machine Learning Research*, vol. 24, no. 34, pp. 1–11, 2023.

[7] C. Agarwal, S. Krishna, E. Saxena, M. Pawelczyk, N. Johnson, I. Puri, M. Zitnik, and H. Lakkaraju, "Openxai: Towards a transparent evaluation of model explanations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 784–15 799, 2022.

[8] M. K. Belaid, R. Bornemann, M. Rabus, R. Krestel, and E. Hüllermeier, "Compare-xai: Toward unifying functional testing methods for post-hoc xai algorithms into a multi-dimensional benchmark," in *World Conference on Explainable Artificial Intelligence*. Springer, 2023, pp. 88–109.

[9] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, pp. 647–665, 2014.

[10] I. Covert and S.-I. Lee, "Improving kernelshap: Practical shapley value estimation via linear regression," *arXiv preprint arXiv:2012.01536*, 2020.

[11] P. Q. Le, M. Nauta, V. B. Nguyen, S. Pathak, J. Schlötterer, and C. Seifert, "Benchmarking explainable ai: a survey on available toolkits and open challenges," in *International Joint Conference on Artificial Intelligence*, 2023.

[12] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, and R. Ranjan, "Explainable ai (xai): Core ideas, techniques, and solutions," *ACM Comput. Surv.*, vol. 55, no. 9, jan 2023.

[13] G. Attanasio, E. Pastor, C. Di Bonaventura, and D. Nozza, "ferret: a framework for benchmarking explainers on transformers," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, D. Croce and L. Soldaini, Eds. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 256–266.

[14] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[15] 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*, 2016, pp. 1135–1144.

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

[17] H. Chen, I. C. Covert, S. M. Lundberg, and S.-I. Lee, "Algorithms to estimate shapley value feature attributions," *Nature Machine Intelligence*, vol. 5, no. 6, pp. 590–601, 2023.

[18] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3319–3328.

[19] M. Ancona, C. Oztireli, and M. Gross, "Explaining deep neural networks with a polynomial time algorithm for shapley value approximation," in *International Conference on Machine Learning*. PMLR, 2019, pp. 272–281.

[20] I. Covert, C. Kim, and S.-I. Lee, "Learning to estimate shapley values with vision transformers," 2023.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[22] R. Wang, X. Wang, and D. I. Inouye, "Shapley explanation networks," *arXiv preprint arXiv:2104.02297*, 2021.

[23] K. Bache and M. Lichman, "UCI machine learning repository," 2013.

[24] J. Howard and S. Gugger, "Fastai: a layered api for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020.

[25] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3498–3505.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.