Explain Yourself

Expanding and optimizing models to enable fast Shapley value approximations

Holger Ziekow, Peter Schanbacher, Valentin Göttisheim
Faculty of Business Information Systems
Furtwangen University
Furtwangen, Germany
email: {Holger.Ziekow, Peter.Schanbacher, Valentin.Goettisheim}@hs-furtwangen.de

Abstract — **This paper addresses the problem of providing fast and accurate approximations of Shapley values for neural networks by embedding the approximation directly into the network architecture. The approach is tested on a synthetic and a real world dataset. The results demonstrate that integrating Shapley value approximations into the loss function enables making a trade-off between explainability and prediction accuracy, optimizing both aspects. This method yields accurate approximations while improving the model's explainability, making it more stable and easier to explain in practical applications.**

Keywords - Explainable AI; Machine Learning; Neural Networks; Shapley value approximation.

I. Introduction

In various applications, understanding and explaining the behavior of neural networks is crucial for both internal management decision-making and meeting the requirements of regulators and external stakeholders. As neural networks are increasingly deployed in critical areas such as finance, healthcare, and autonomous systems, the need for transparency and explainability becomes paramount. Stakeholders need to trust that the models are making decisions based on relevant and understandable factors, and they must be able to justify these decisions to regulatory bodies and customers alike [1].

A powerful tool for gaining insights into the relevance of attributes in these models is the use of Shapley values [2]. Originating from cooperative game theory, Shapley values provide a fair distribution of the total gain generated by a coalition of players, attributing a value to each player's contribution. When applied to neural networks, Shapley values help users understand how each input feature contributes to the model's prediction. They are prized for their desirable properties, such as fairness, efficiency, and consistency, making them an ideal choice for feature attribution. A significant challenge with Shapley values is that their exact evaluation is computationally expensive, with the complexity growing exponentially with the number of input features [3]. This computational burden makes them impractical for large-scale applications involving high-dimensional data. To mitigate this, researchers have developed various approximation methods. Notably, the authors in [4] introduced polynomial-time approximations, which significantly reduce the computational load while still providing useful insights into feature importance.

This work advances this field by demonstrating that the approximation of Shapley values can be seamlessly integrated into the training process of neural networks. Specifically, a method is proposed where the outputs of interest from the neural network are extended to include these approximated Shapley values. This integration occurs during the training phase, ensuring that the model not only learns to make accurate predictions but also provides explanations for these predictions concurrently.

A key benefit of this integration is that it enables a direct trade-off between model accuracy and Shapley value approximation. In addition, this approach enables improved explainability of the model as well as the immediate availability of explanations.

By integrating Shapley value approximations during training, the neural network converges to a state that is inherently easier to explain. For instance, the network's responses to changes in input features become more stable. This smoothing effect is often a desirable property, especially in domains where stakeholders need to understand the model's behavior in intuitive terms. It prevents scenarios where minor changes in input result in disproportionately large and unexpected changes in the output, which can be challenging to justify to customers and regulators [1]. An explainable model enhances trust and facilitates better decision-making.

Additionally, the approximated Shapley values are produced as a direct result of the model's predictions. This means that for every prediction the model makes, an accompanying explanation is immediately available. This capability is appealing in applications requiring high-frequency predictions and where each decision needs to be justified on the spot.

The approach is particularly valuable in applications where the model undergoes a single training phase followed by numerous predictions, each requiring an explanation. This ensures that the model not only performs well in terms of predictive accuracy but also remains transparent and explainable throughout its operational lifecycle. By embedding the approximation of Shapley values into the training process, the approach strikes a balance between computational efficiency and the need for clear,

ISBN: 978-1-68558-215-9

understandable explanations, meeting the demands of both operational efficiency and regulatory compliance.

The remainder of the paper is structured as follows: Section 2 discusses the related work and the inclusion of Shapley values into the model's prediction is laid out in Section 3. Section 4 presents an analysis with the data and model applied. The results are discussed in section 4. Section 5 summarizes and concludes.

II. Related Work

Shapley values, originating from cooperative game theory, have become a fundamental tool for feature attribution in machine learning models [2]. They offer a fair distribution of the total gain generated by a coalition of players, attributing a value to each player's contribution [3][5]. However, their exact computation is computationally expensive, leading to the development of various approximation methods [6]. This section reviews these methods, highlighting the limitations they present, and the gaps the proposed approach aims to address.

Feature-removal approaches are central to feature contributions in Shapley value calculations [6]. They involve systematically removing features and assessing the impact on the model's output. The primary types are: (1) Baseline Shapley values where missing features are replaced with values from a baseline sample, such as zeros, means, or medians. This approach is simple to implement and interpret; however, the choice of baseline can be arbitrary and may not accurately represent the data distribution [5][7]. (2) Marginal Shapley values calculate the marginal expectation of the model output by treating absent features as random variables following their marginal distribution. It involves evaluating the model with subsets of features including and excluding the feature of interest. It provides a more accurate estimate of feature importance by considering the marginal distribution of features. However, it is computationally more expensive as it requires multiple model evaluations for different subsets of features [7]. (3) Conditional Shapley values which define the game by the conditional expectation of the model output, where absent features are treated as following a conditional distribution given the observed features. It considers the interdependencies between features [7]. This most accurately accounts for the conditional dependencies between features, providing a realistic assessment of feature importance. However, it is highly complex and computationally intensive due to the need for estimating conditional distributions, which can be challenging, especially in high-dimensional data.

To address the computational challenges of exact Shapley value calculations, various approximation strategies have been developed. These strategies can be broadly categorized into model-agnostic approximations, which are applicable to any model type, and model-specific approximations, which are tailored to specific model structures. Model-agnostic approximations include methods such as interactions-based method for explanation (IME) [9] and KernelSHAP [5][10]. IME utilizes stochastic sampling to provide unbiased estimates of Shapley values. While broadly applicable to various models, it is computationally intensive. KernelSHAP also employs a sampling-based approach, reducing computational load but still requiring significant resources.

In contrast, model-specific approximations are tailored to particular model structures. TreeSHAP [7] leverages the inherent structure of decision trees to compute exact Shapley values efficiently. It offers faster and more precise calculations but is limited to tree-based models. Similarly, LinearSHAP [11] computes Shapley values exactly for linear models with linear time complexity. It performs well for linear relationships, however, is not suitable for other models. While approximation methods like KernelSHAP and IME provide useful insights with reduced computational demands, they suffer from high variance and are still resource intensive. Assumption-based methods like TreeSHAP and LinearSHAP offer solutions with lower computational costs but are restricted to specific model types.

Some research has focused on considering Shapley value approximations into the model architecture itself to balance accuracy and computational efficiency. For instance, ShapNets [12] are designed to facilitate easier estimation of Shapley values through specific network architectures, enhancing both explainability and performance. Deep Approximate Shapley Propagation [4] leverages uncertainty propagation to estimate Shapley values, providing deterministic results with moderate computational requirements.

The proposed approach distinguishes itself by embedding Shapley value approximations directly into the neural network training process. This integration ensures that the model's predictions are inherently more explainable due to more stable responses to input feature changes. Additionally, it allows for the immediate availability of explanations with each prediction, a crucial advantage in settings requiring frequent and justifiable decisions. By embedding the Shapley value approximation into the network architecture, the proposed method achieves a balance between computational load and the need for clear, understandable explanations. It also enables an explicit trade-off between model performance and quality of Shapley value approximations. The proposed integrated approach offers a novel solution that enhances both explainability and efficiency, meeting the demands of real-world applications requiring transparency and accountability.

III. Modeling Shapley Values

Shapley values are a well-established method to understand the impact of an attribute on the outcome [5]. Consider a data set of N attributes and a model f mapping each subset S of the attributes to real numbers (i.e., a prediction). The Shapley value quantifies the importance of attribute i to the prediction. To determine the effect, a model $f_{S\cup\{i\}}$ using data $x_{S\cup\{i\}}$ for a subset S of features including feature i and a model f_S using data x_S without feature i. Now for all possible subsets $S \subseteq F \backslash \{i\}$ the impact of withholding feature i is calculated. The Shapley values are calculated based on the weighted average of all possible differences.

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

Computing Shapley values requires the evaluation of all possible feature subsets, which makes it infeasible for common practical applications with many features to consider. Shapley values sampling is most frequently used to approximate the Shapley values [13]. Despite the approximation, it still requires considerable calculation time. The standard approach to predict outcome y based on input x, is to minimize the objective $f = \arg\min_f E\left[\left(y - f(x)\right)^2\right]$. This work aims to predict the Shapley values of the features as well, hence optimizing function $g: \mathbb{R}^N \rightarrow \mathbb{R}^{N+1}, g(x) \rightarrow \left(y, \phi_1(x), .., \phi_n(x)\right)$ such that we minimize:

$$\mathrm{g} = \arg\min_{\mathrm{g}} \mathrm{E}\left[\left(\mathrm{y} - \mathrm{g}_0(\mathrm{x})\right)^2 + \lambda \sum_{\mathrm{i}=1}^{\mathrm{N}} \left(\phi_{\mathrm{i}}(\mathrm{x}) - \mathrm{g}_{\mathrm{i}}(\mathrm{x})\right)^2\right]$$

The hyperparameter λ can be used for a trade-off between the standard approach (λ=0) and a joint prediction of outcome y and the Shapley values ϕ_i (λ>0). The hyperparameter λ controls the balance between prediction accuracy and Shapley value approximation. At λ = 0, the model optimizes accuracy, while increasing λ improves explainability by incorporating Shapley values, albeit with some loss in accuracy. Higher λ values shift the focus more toward generating accurate Shapley values.

IV. Experimental Setup

As a test model, a neural network with a three-node input layer, a hidden layer of 16 neurons, another hidden layer of 8 neurons and a four-neurons output layer (see Figure 1) is built. The output contains y_j as well as the three Shapley values $\phi_1(x_j)$, $\phi_2(x_j)$, $\phi_3(x_j)$ for $x_j = (x_{0j}, x_{1j}, x_{2j})$. For the hidden layers, a leaky ReLu is used ($\alpha = 0.1$). The MSE is optimized using the ADAM [14] optimizer.

The model is trained once with minimizing the MSE of the output of interest y_j only and no weight on accurate Shapley value approximations (λ=0). A second model is trained for the joint prediction of the output of interest as well as the Shapley values (λ=1). A third model is trained with joint prediction of the output of interest and a very high weight on Shapley value approximations (λ=1000). A batch size of one was chosen for pragmatic reasons. In each forward pass we compute the target Shapley values of the model with an existing technique. In our tests we used KernelExplainer from the SHAP library [5]. However, this may be replaced with any other method. We use these values to compute the error for $\phi_1(x_j)$, $\phi_2(x_j)$, $\phi_3(x_j)$.

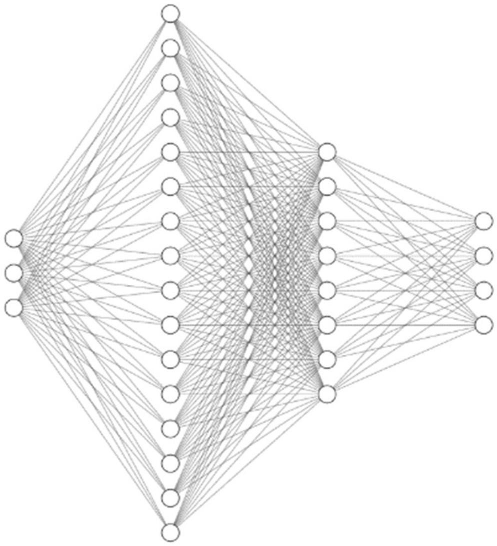

Figure 1. Architecture for the neuronal network (created with https://alexlenail.me/NN-SVG/index.html)

For bigger λ values we expect an increase of the MSE based on the outcome y_j, as the introduction of the Shapley values leads to a biased prediction. We also expect reduced errors for the Shapley value approximations as λ increases. Furthermore, we expect simpler relations between feature values and their corresponding Shapley values, which are easier to approximate. This should be apparent when plotting the feature values against the targeted Shapley values (in our tests computed with KernelExplainer from the SHAP library [5].

A. Experiments with synthetic data

For illustration and initial analysis, we use a synthetic dataset generated as follows. The target variable y_j is created using the linear relationship:

$$y_j = 2 \cdot x_{0j} + \frac{1}{2}\epsilon_j, j \in \{1, \dots, 1000\}$$

where x_{0j} is the first feature, and ε_j represents independent and identically distributed (i.i.d.) noise drawn uniformly from the interval [0, 1]. The second feature x_{1j} is also i.i.d. and uniformly distributed, generated independently from the same interval. The third feature x_{2j} is then derived from a non-linear transformation of x_{1j} and y_j as follows:

$$x_{2j} = \left(x_{1j} + y_j\right)^{\frac{1}{4}}, j \in \{1, \dots, 1000\}$$

We use 80% of the generated data as the training set and 20% as the test set. The synthetic data was designed to exhibit both simple and complex relationships between the features (x_{0j}, x_{1j}, x_{2j}) and the target variable y_j. This setup allows us to demonstrate the desired trade-off between prediction

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-215-9

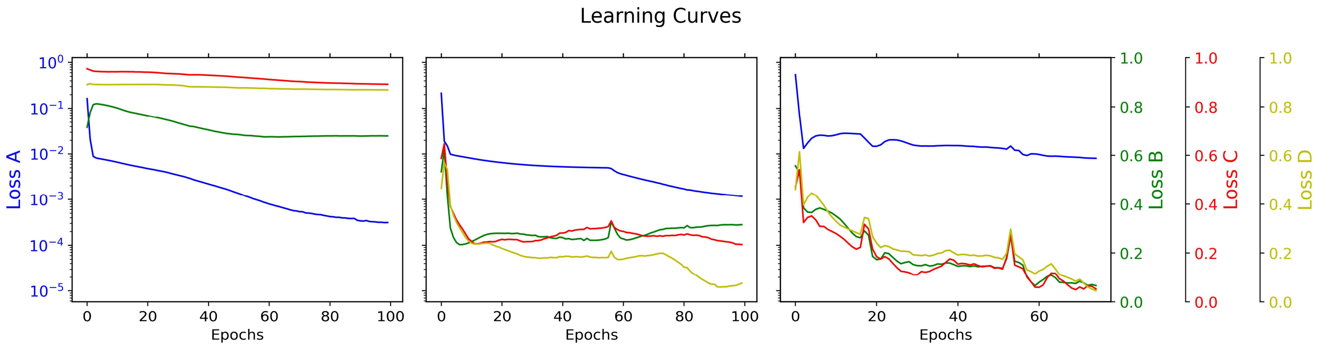

Figure 2. MSE for $\lambda \in \{0,1,100\}$ (left to right) models for the outcome of interest y (blue) and the corresponding Shapley values (green, red, orange).

accuracy and the approximability of Shapley values. The model and training procedure are implemented as described above. The number of epochs was chosen based on the learning curves observed across all tests, ensuring that training did not stop prematurely due to a sudden error spike in any model. The resulting learning cures are shown in Figure 2. The model outputs A, B, C, D, represent the model prediction y (i.e., y=A) and the predicted Shapley values for the features 1, 2, 3.

As expected, higher λ values drive down the errors for Shapley value predictions and increase the prediction error for the target A. In detail the MSE for the model λ=0 is 0.0003, while the MSE for the λ=100 model is 0.001. It is also observed – as expected – that the partial dependency plots show increasingly simpler structures (see Figure 3). The resulting curves become more smooth and less scattered. This makes them easier to approximate and easier to interpret by humans.

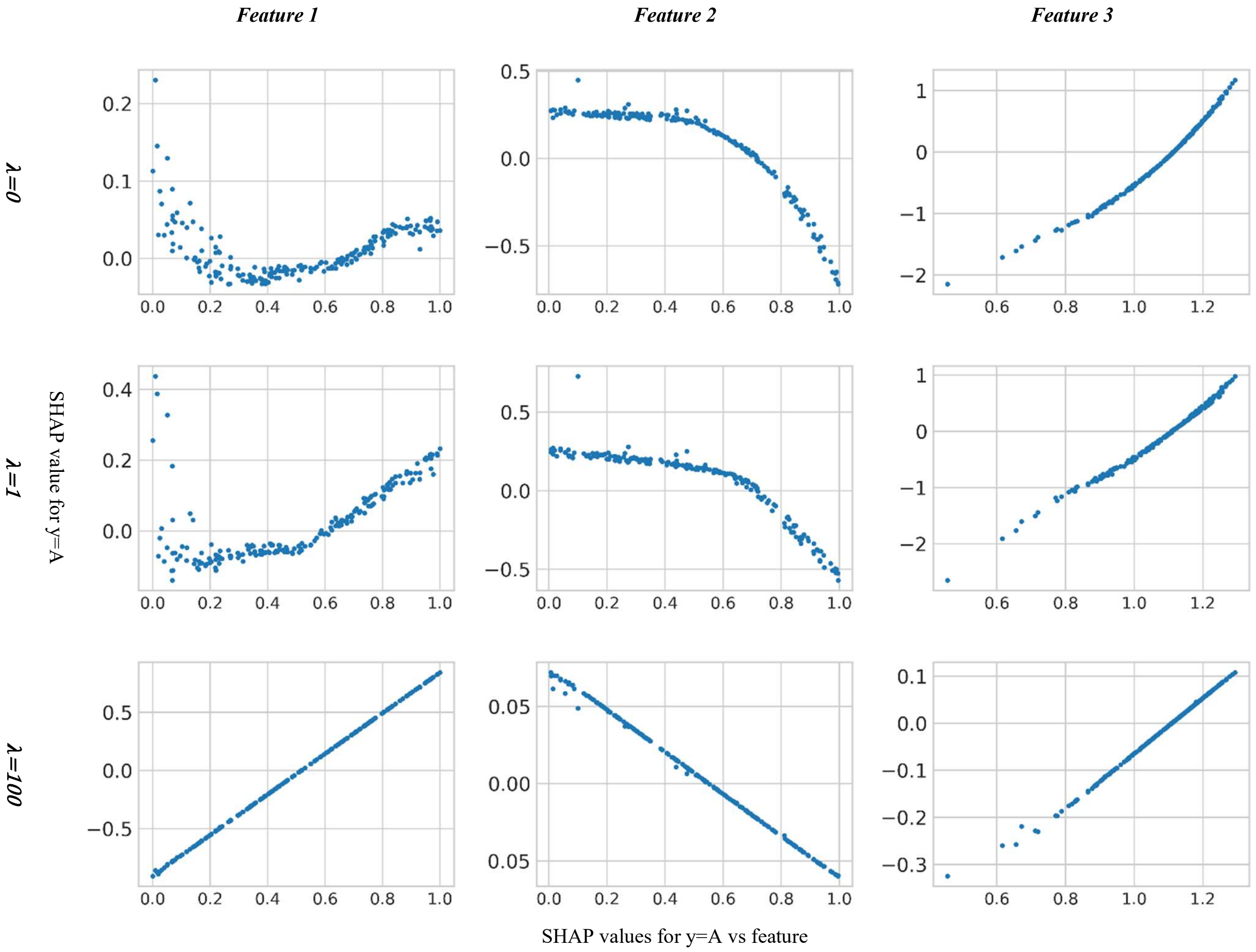

Figure 3. Shapley values of features 1,2,3 (left to right) of models $\lambda \in$ 0,1,10 (top to bottom)

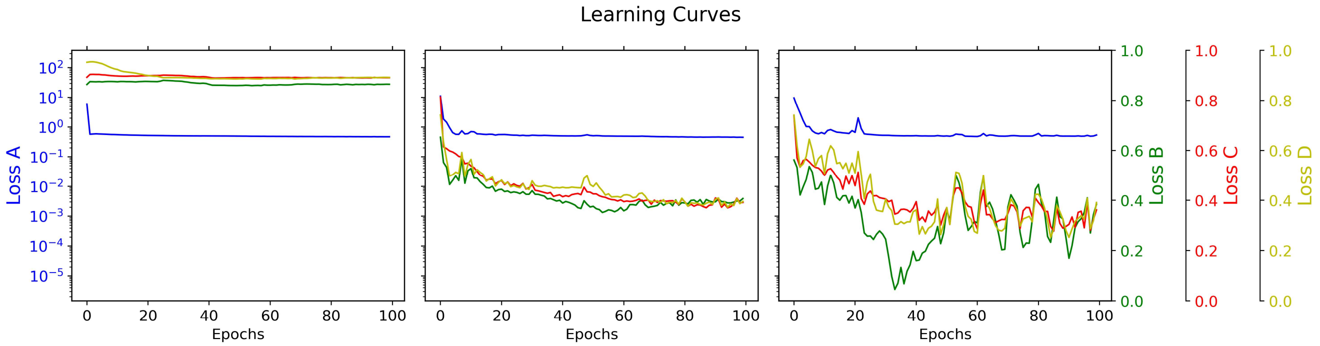

Figure 4. MSE for $\lambda \in \{0, 10, 1000\}$(left to right) models for the outcome of interest y (blue) and the corresponding Shapley values (green, red, orange).

The results demonstrate the desired tendency towards more explainable models with higher values for λ. Overall, the tests verify the feasibility of the proposed approach and demonstrate the desired effects.

B. Experiments with real data

A publicly available data set from openml.org was chosen to verify the applicability of the approach on real data. Specifically, the data set named wine-quality-red was used to predict wine quality [15]. The network structure remained the same as described above, with three features for predicting the target. The target variable includes 6 levels of quality, and the learning problem is treated as a regression problem. The selected features are 'sulphates', 'alcohol', and 'total_sulfur_dioxide'. Feature selection was done based on exploratory analysis for identifying features with non-linear relations to the target. This was done to give room for a trade-off between model accuracy and simplicity of the Shapley value approximation.

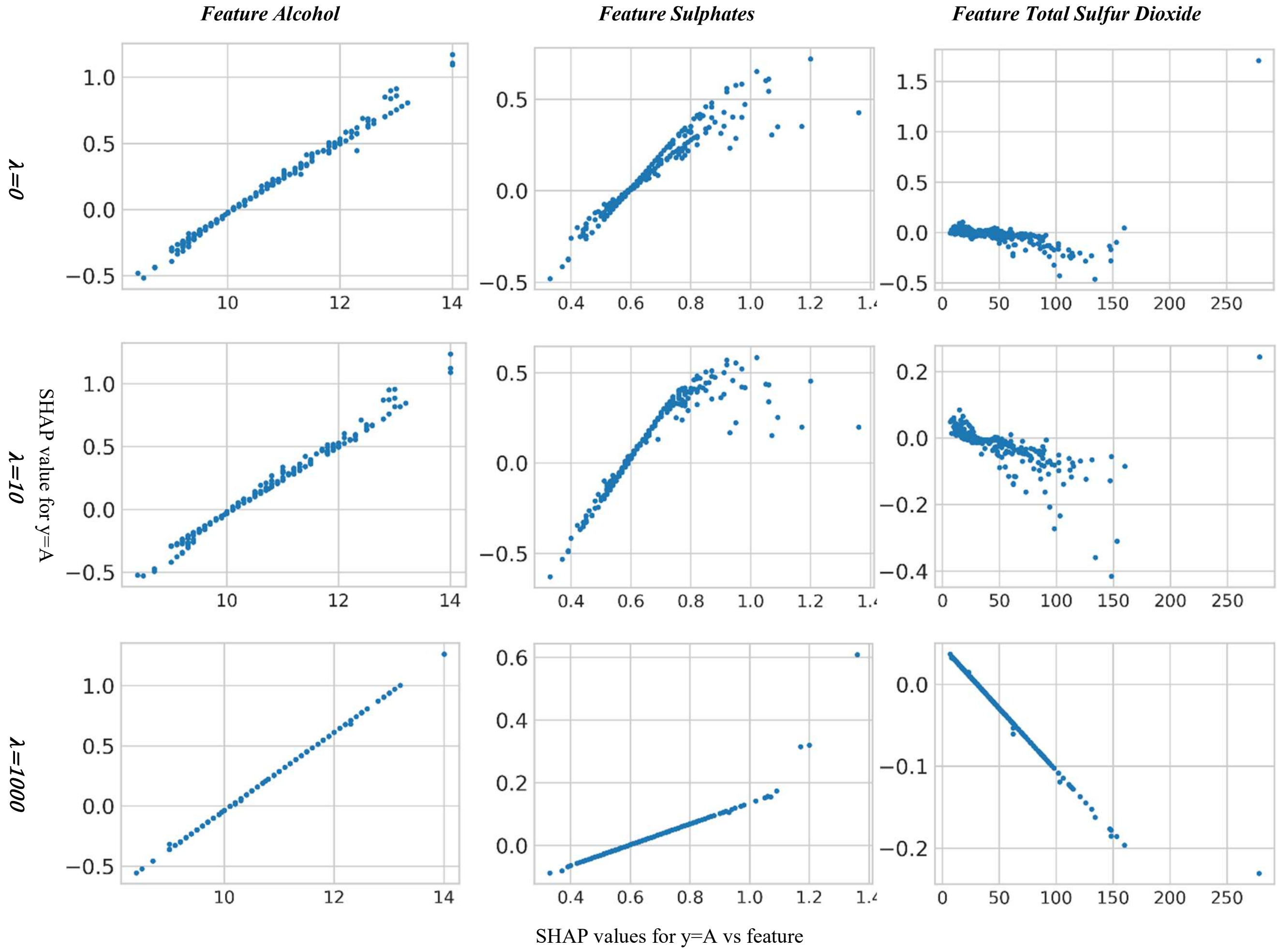

Figure 5. Shapley values of features 1,2,3 (left to right) of models $\lambda \in 0,1,10$ (top to bottom)

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-215-9

Model and training procedure followed the procedure described above, with 100 training epochs. Again, the number of epochs was chosen based on the learning curves of all tests (i.e., ensuring not to stop at a sudden error spike for any model). The resulting learning curves are shown in Figure 4 The model outputs A, B, C, D, represent the model prediction y (i.e., y=A) and the predicted Shapley values for the features 'sulphates', 'alcohol', and 'total_sulfur_dioxide'.

The experiment with real data show the same general effects as the experiments with synthetic data. Specifically, higher λ values reduce the errors in Shapley value predictions but increase the prediction error for the target variable y. For instance, for epoch 100 the MSE the for model . λ=0 is 0.460, while for the model with. λ=10, the MSE decreases to 0.45. However, for . λ=1000, an increase of the MSE to 0.54 can be observed. The learning curves for all models exhibit similar behavior, while the Shapley value approximation shows significant improvement in smoothness and explainability. Again, we observe that the partial dependency plots show increasingly simpler structures (see Figure 5). These findings confirm the applicability of the approach with real data.

V. CONCLUSION AND FUTURE WORK

Explaining neural networks remains a challenging task, often due to the complexity and non-linear nature of these models. Often minor changes in input data can lead to significantly different model outcomes, which complicates explaining these changes to users. It was found that training models with a focus on Shapley values results in more stable and explainable outputs. This approach enhances the consistency of explanations derived from Shapley values, making the model's behavior more predictable and understandable. Contrary to initial expectations, incorporating Shapley values into the training process did not lead to a significant decline in predictive performance, as measured by the mean squared error of the outcome of interest. This suggests that it is possible to maintain accuracy while improving explainability.

Future work could explore adjusting batch sizes to balance convergence and estimation accuracy, as larger batch sizes, while smoothing convergence, may reduce the precision of Shapley value approximations. Moreover, increasing λ improves explainability, it may reduce sensitivity to rare or extreme cases. And scaling to high-dimensional data poses challenges, suggesting more efficient methods for Shapley approximations should be developed. Additionally, expanding experiments to include more diverse datasets could further validate the approach and confirm its generalizability across different domains.

Nevertheless, the proposed model offers the advantage of providing direct explanations for its predictions. This feature is particularly valuable for internal stakeholders, such as management, and external stakeholders, such as regulators, who often require transparent and understandable model explanations.

Based on these findings, this paper recommends adopting our approach for AI models that have to be rarely updated but are frequently used for prediction tasks. This methodology ensures that the model not only performs well but also delivers reliable explanations in the form of Shapley values, thereby meeting the growing demand for transparency in AI systems.

REFERENCES

[1] European regulation on artificial intelligence, “EU AI Act”, [Online], https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138-FNL-COR01_EN.pdf, [retrieved: 10, 2024].

[2] L. Shapley, “Notes on the n-Person Game -- II: The Value of an n-Person Game.”, Santa Monica, Calif.: RAND Corporation, 1951.

[3] L. Merrick and A. Taly, “The Explanation Game: Explaining Machine Learning Models Using Shapley Values” *Machine Learning and Knowledge Extraction*, Springer International Publishing, CD-MAKE 2020, pp. 17-38, Dublin, Ireland, August 25–28, 2020.

[4] M. Ancona, C. Oztireli, and M. Gross, “Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation”, *Proceedings of the 36th International Conference on Machine Learning*, PMLR, pp. 272-281, 2019.

[5] S. Lundberg and S. Lee, “A Unified Approach to Interpreting Model Predictions”, *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

[6] H. Chen, I. Covert, and S. Lundberg, “Algorithms to estimate Shapley value feature attributions”, *Nature Machine Intelligence 5*, pp. 590–601, 2023.

[7] M. Sundararajan and A. Najmi, “The Many Shapley Values for Model Explanation”, *Proceedings of the 37th International Conference on Machine Learning*, in Proceedings of Machine Learning Research, pp. 9269-9278, 2020.

[8] S. Lundberg, G. Erion, H. Chen, et al., “From local explanations to global understanding with explainable AI for trees”, *Nature Machine Intelligence 2*, pp. 56–67, 2020.

[9] E. Strumbelj and I. Kononenko, “An Efficient Explanation of Individual Classifications using Game Theory”, *Journal of Machine Learning Research 11*, pp. 1–18, 2010.

[10] I. Covert and S. Lee, “Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression”, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, in Proceedings of Machine Learning Research, pp. 3457-3465, 2021.

[11] H. Chen, J. Janizek, S. Lundberg, and S. Lee, “True to the Model or True to the Data?”, *ArXiv, abs/2006.16234,* 2020.

[12] R. Wang, X. Wang, and D. Inouye, “Shapley explanation networks”, *In Proc. International Conference on Learning Representations*, ICLR, 2021.

[13] J. Castro, D. Gamez, and J. Tejada, “Polynomial calculation of the shapley value based on sampling”, *Computers and Operations Research*, pp. 1726 – 1730, 2009.

[14] D. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *ICLR: international conference on learning representations,* pp. 1-15, 2014.

[15] OpenML, “Red Wine Quality Dataset. Dataset”, [Online] https://openml.org/search?type=data&status=active&id=40691, [retrieved: 10, 2024].