Feature Importance for Deep Neural Networks: A Comparison of Predictive Power, Infidelity and Sensitivity

Lars Fluri[a]
Department of Computational Economics and Finance, Department of Management Accounting
University of Basel, Peter Merian-Weg 6, Basel, Switzerland
fl

Keywords: XAI, XML, SHAP, Shapley Value Sampling, DeepLIFT, LIME, Integrated Gradients, GradientSHAP, Deep Learning, Neural Networks, Finance.

Abstract: This paper evaluates the effectiveness of different feature importance algorithms employed on a neural network, focused on target prediction tasks with varying data complexities. The study reveals that the feature importance algorithms excel with data featuring minimal correlation between the attributes. However, their determination considerably decreases with escalating levels of correlation, while the inclusion of irrelevant features has minimal impact on determination. In terms of predictive power, DeepLIFT surpasses other methods for most data cases, but falls short in total infidelity. For more complex cases, Shapley Value Sampling outperforms DeepLIFT. In an empirical application, Integrated Gradients and DeepLIFT demonstrate lower sensitivity and lower infidelity, respectively. this paper highlights interesting dynamics between predictive power and fidelity in feature importance algorithms and offers key insights for their application in complex data scenarios.

1 INTRODUCTION

While machine learning techniques, particularly neural networks, have demonstrated tremendous potential across various applications, their adoption in academic research has been hindered by their ”black box” nature (Castelvecchi, 2016). Despite their considerable predictive power, these models often have non-transparent functional forms. This is especially challenging in fields like economics, finance, and social sciences, where understanding the relationships between variables is crucial (Molnar, 2020). This study aims to address this gap by investigating the effectiveness of feature importance methods in providing interpretable insights for machine learning models. Specifically, it contributes to the literature by assessing these methods in a controlled simulation with known ground truth and comparing their performance in an empirical case study. By doing so, it seeks to enable both academic researchers and industry practitioners to interpret machine learning models and explain their predictions in understandable terms. The opacity of machine learning models has driven research in explainable machine learning (XML) and explainable artificial intelligence (XAI). Feature importance methods, developed to calculate the significance of individual features, have gained popularity, especially in image analysis and pattern recognition. However, a thorough examination of their explanatory power in economics and finance is lacking. This study's findings are crucial for advancing the interpretability of machine learning in these domains by highlighting strengths, weaknesses, and possible pitfalls of various feature importance methods..

2 LITERATURE REVIEW

Using methods for explainability and interpretability is widespread in the current research literature — this subsection mentions the most relevant works. There is a plethora of feature importance methods for neural networks and machine learning models in general. Examples include *Integrated Gradients* (Sundararajan et al., 2017), *Shapley Additive Explanations* (Lundberg and Lee, 2017), *Local Interpretable Model-Agnostic Explanations* (Ribeiro et al., 2016), and *Deep Learning Important FeaTures* (Shrikumar et al., 2017). Additional measures include *RelATive cEntrality* to prioritise candidate genetic variants (Crawford et al., 2019). This framework was also ex-

[a] https://orcid.org/0009-0005-0031-8355

Fluri, L.
Feature Importance for Deep Neural Networks: A Comparison of Predictive Power, Infidelity and Sensitivity.
DOI: 10.5220/0012903300003886

In *Proceedings of the 1st International Conference on Explainable AI for Neural and Symbolic Methods (EXPLAINS 2024)*, pages 15-26
ISBN: 978-989-758-720-7

tended for highly collinear predictors (Ish-Horowicz et al., 2019). Applications of feature importance methods and neural networks in the fields of economics and finance are widespread. Convolutional neural networks (CNN) have been shown to be effective methods when it comes to financial time-series prediction tasks (Chen et al., 2016). In the field of feature importance, comparisons of feature selection methods based on importance calculations for solving classification problems in finance exist (Xiaomao et al., 2019). Research in finance also started to incorporate feature selection for financial stress predictions (Liang et al., 2015). However, the current literature is suffering from an interesting knowledge gap. Specifically, it is assumed that the feature importance attribution appropriately reflects the true underlying causal link between the input features and the output prediction. However, in empirical data, the true underlying relationships between the target and the features are generally not observable, and most of the time completely unknown. Therefore, feature importance attributions are necessarily only approximations of the underlying relationships. To gauge the relative determination and predictive power of feature importance attribution, further research is therefore necessary. In this paper, a synthetic data generating process (DGP) is used to test various feature importance methods in terms of predictive power, infidelity, and sensitivity. This paper thus provides a comprehensive analysis of the predictive power of various feature importance methods. This paper also closes a methodological gap by offering a comparative setup to justify the chosen importance methods.

3 METHODS FOR FEATURE IMPORTANCE ATTRIBUTION

This section provides an overview over some of the most important feature importance attributions currently in use. Subsection 3.1 provides a general introduction to the methodology, while Subsection 3.2 highlights the specific methods used in this study. Subsection 3.3 briefly discusses limitations and drawbacks.

3.1 General Overview

Feature importance methods are tools used to measure how much influence each feature in the data has on the model's predictions. This subsection briefly highlights general concepts. More in-depth analyses and reviews can be found in the surrounding literature (Gevrey et al., 2003), which reviews and compares different feature selection methods, including those based on neural networks, decision trees, and regression models. Since this study is concerned with (deep) neural networks, neural-network specific as well as model-agnostic importance attribution methods are introduced. Feature importance methods are divided into backward-based and forward-based methods. Forward-based methods move from the input to the output through the neural network, while backward-based methods move backwards from the output to input to compute importance. Forward-based methods compute the importance of a feature as the difference in the output between a trained model where the feature is present versus a trained model where the feature is missing. This can be understood as a leave-one-out calculation of importance. Backward-based methods measure the importance of a feature through the gradient (or a gradient-similar measure) of the output with respect to the inputs. Subsection 3.2 briefly introduces the relevant methods and their theoretical frameworks.

Table 1: Importance attribution method overview can generally be categorised by model-agnosticism and mechanism of importance computation.

method	agnosticism	mechanism
IG	no	gradient
GSHAP	yes	perturbation (gradient-boosted)
LIME	yes	perturbation
DeepLIFT	no	gradient-like
SVS	yes	perturbation

3.2 Method Overview

The selected methods for this study are Integrated Gradients (IG), GradientSHAP (GSHAP), Local Interpretable Model-Agnostic Explanations (LIME), ShapleyValueSampling (SVS), and DeepLIFT. They were chosen due to their concept heterogeneity and popularity in previous applications. The following paragraphs briefly discuss their methodological frameworks.

IG. IG was proposed as a framework for interpreting the predictions of deep neural networks by assigning attribution scores to input features (Sundararajan et al., 2017). IG back-propagates and calculates importance through gradients. Assume that $f : \mathbb{R}^d \to \mathbb{R}$ is a function that represents a neural network, x is the input at hand, and x^0 is a baseline input. In terms of a single observation and considering the straight line (in $\mathbb{R}^n$) from the baseline x^0 to the input x, one can compute the gradients at all points along the path. Integrated gradients are obtained by integrating over the computed gradients. The integrated gradient $\mathrm{IG}_i(x)$ along the i^{th} dimension for a single observation input x and baseline x^0 is defined as

$$\left(x_i - x_i^0\right) \cdot \int_0^1 \frac{\partial f\left(x^0 + \alpha \cdot \left(x - x^0\right)\right)}{\partial x_i} d\alpha, \qquad (1)$$

where $\frac{\partial F(x)}{\partial X_i}$ is the gradient of $f(x)$ along the i^{th} feature dimension (Sundararajan et al., 2017).

LIME. LIME is a model-agnostic method that offers local explanations for the prediction of any classifier (Ribeiro et al., 2016). LIME learns an interpretable model around the prediction with the goal of faithfully replicating the classifier's behaviour in the local region. The attribution method defines an explanation model $g \in G$ in order to explain the function f, where G is a class of potentially interpretable models. LIME then produces a local explanation obtained by

$$\xi(x_i) = \underset{g \in G}{\operatorname{argmin}} \quad \mathcal{L}(f, g, \pi_{x_i}) + \Omega(g). \qquad (2)$$

$\mathcal{L}$ is a measure of how unfaithful a simplified explanation model g is in explaining the function f in the neighbourhood π_{x_i} and $\Omega(g)$ is a measure of complexity of the explanation model. LIME can be used with different model classes G, fidelity functions $\mathcal{L}$, and complexity measures Ω. LIME is designed to be model-agnostic and therefore does not make any assumptions about f. Furthermore, it is a local method, because it computes the importance of a feature with respect to a prediction on a single observation. LIME can also be aggregated to create a global understanding of feature importance, as will be shown in Section 4.6.

DeepLIFT. DeepLIFT was introduced as a technique to decompose the output prediction of a neural network by back-propagating the contributions of all neurons in the network to every feature of the input (Shrikumar et al., 2017). It assigns contribution $C_{\Delta x_i \Delta t}$ scores based on difference-from-reference neuron activations. As described previously, reference activations are synonymous to baseline activations. For the feature along the i^{th} feature dimension, the difference-from-reference between observation x and the reference x^0 is denoted by by Δx_i. The difference in the neuron output is given as

$$\Delta t = t - t^0, \qquad (3)$$

where t is the activation of a neuron at x and t^0 is the reference activation of the neuron at x^0. DeepLIFT requires completeness of the form

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t, \qquad (4)$$

indicating that the sum of attributions must be equivalent to the difference-from-reference activation. DeepLIFT is not model-agnostic because it is only applicable to neural networks.

SHAP. SHAP is a forward-based importance method that computes importance using perturbations and differences in model outputs (Lundberg and Lee, 2017). SHAP is based on Shapley values from cooperative game theory, and it has several desirable theoretical properties. SHAP assigns importance by training two different models, one with the feature present $f_{S \cup \{d\}}$ and one with the feature withheld f_S. It then compares predictions from the two models

$$f_{S \cup \{d\}}\left(x_{S \cup \{d\}}\right) - f_S\left(x_S\right). \qquad (5)$$

Since the effect of one feature depends on other features in the set S, the difference described beforehand is computed for all possible subsets $S \subseteq F \backslash \{d\}$, where F is the complete set of all features. The Shapley values ϕ_i are a weighted average of all possible differences

$$\phi_i = \sum_{S \subseteq F \backslash \{d\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \cdot \left[f_{S \cup \{d\}}\left(x_{S \cup \{d\}}\right) - f_S\left(x_S\right)\right] \qquad (6)$$

and used as a proxy for feature importance. Due to the computational complexity of Shapley values, multiple approximations and boosted methods have been introduced. *Shapley Value Sampling* (SVS) (Strumbelj and Kononenko, 2010) is a sampling-based approach to the calculation of SHAP that reduces computational burden (Castro et al., 2009). In this study, it is used as a proxy for SHAP values. *GradientSHAP* (GSHAP) is an extension of the original SHAP method. It calculates the gradient of outputs with respect to a chosen baseline and input observation. The SHAP values can then be estimated using the expected values of these gradients times the difference between inputs and baselines.

3.3 Limitations, Drawbacks and Extensions

Both forward-based and backward-based attribution methods suffer from drawbacks and limitations. Generally speaking, all methods require baselines to calculate the importance attribution of features, where the baseline represents the normal or typical behaviour of the model. These baselines are crucial for the calculation of importance attribution and must be chosen carefully. Previous research has explored possible (dis-)advantages of different baselines for image analysis (Sturmfels et al., 2020). Backward-based methods are limited by multiple issues. First, they struggle with some activation functions of neural networks, for example the ReLu. ReLu zeroes out gradients which makes it hard to approximate importance. Additionally, gradient-based methods often struggle with modelling saturation, meaning that once a feature reaches a certain threshold where its importance to the target stays the same, it may incorrectly be given an importance score of zero. Forward-based methods can be computationally expensive because they require training of the model for every possible subset of features. Sampling mechanisms, i.e. SVS, or boosting procedures such as GradientSHAP may be reduce the computational burden. However, the analysis of high-dimensional still introduces significant computational burden. Additionally, forward-based methods generate an out-of-distribution (OOD) problem for the neural network because they force the model to extrapolate it to a point of the multivariate distribution that does not naturally occur in the data. Finally, forward-based methods have from theoretical constraints. For example, SHAP assumes that features are independent for the calculation of feature importance. This is a strong assumption in the case of a real-world application that is criticised in (Kumar et al., 2020).

4 EXPERIMENTAL SETUP

This section explains the setup for the simulation study. Subsection 4.1 elaborates on the general setup, while Subsections 4.2, 4.3, and 4.4 discuss specific details of the DGP, the neural network, and the importance attribution methods. Subsection 4.6 describes the performance measurement for the synthetic and empirical data.

4.1 General Setup

The setup of the simulation study in this paper consists of three steps. In the first step, synthetic data is generated from a pre-specified DGP. In the second step, the neural network is trained on the synthetic data. The third and last step consists of the computation of the importance attribution and the analysis of determination. The goal of the data generation is to create a data set where the underlying characteristics are known in order to evaluate how well the importance attribution performs. Table 2 shows the different cases of the DGP. The process for generating synthetic data starts by creating a sample of features $\mathbf{X}$ and a discriminator D as the underlying causal relationship between $\mathbf{X}$ and Y. A more detailed explanation of the feature creation is found in Subsection 4.2. The target variable is created as a noisy linear combination of the discriminator D and the features sample $\mathbf{X}$, formulated as

$$T = D \cdot \mathbf{X} + \varepsilon, \tag{7}$$

where D is a vector of random integers and

$$\varepsilon \sim N\left(\hat{\mu}_T, c^2 \hat{\sigma}_T^2\right),$$

where c is a scaling factor for the variance of the Gaussian noise term. The scaling factor is set to $c = 0.4$. This level of noise in the DGP leads to a true R^2 of between 0.8 and 0.9, which is an appropriate level of disturbance in the data, especially considering that additional disturbances such as marginal transformations as well as spurious and irrelevant features are added. The discriminant D is sampled from a discrete uniform distribution $U\{-10, 10\}$. This means that the features with a negative (positive) discriminant coefficient have a negative (positive) influence on the target T. To create regression targets, T is transformed using a logit transformation of the form

$$f_{\text{logit}}(t_i) = y_i = \frac{e^{t_i}}{1 + e^{t_i}} \quad \forall i \in \{1, \ldots, n\}, \tag{8}$$

to map the target to $[0, 1]$. In order to introduce nonlinearity to the marginal distributions of the features, the generated sample are transformed using a quantile transformation of an arbitrary distribution. After the target is created using the combination mentioned in Equation (8), the features are transformed using a column-wise quantile transformation

$$F_{X_j}^{-1}(X_j) = U_j \quad \forall j \in \{1, \ldots, m\} \tag{9}$$

where $F_{X_i}^{-1}$ is the inverse of the row-wise cumulative distribution function. This maps the normally-distributed variables to $[0, 1]$. Afterwards, the standard uniform sample is transformed by applying a transformation on U such that

$$X_j = F_{X_j}(U_j) \quad \forall j \in \{1, \ldots, m\}. \tag{10}$$

These two steps are a common procedure in simulation and data generation that allow an arbitrary marginal distribution of the features while still preserving the multivariate dependency between features. This increases the data complexity and makes the training process as well as the importance attribution more challenging.

4.2 Feature Creation

Three distinct types of features can be created in the simulation study: Relevant, spurious, and irrelevant features. Relevant features are either continuously or discretely distributed, while spurious and irrelevant features are always continuously distributed. The continuously distributed features are sampled from a multivariate normal distribution given by

$$p\left(\mathbf{X},\mu,\Sigma\right) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{X}-\mu)'\Sigma^{-1}(\mathbf{X}-\mu)}, \quad (11)$$

where is Σ a randomly generated covariance matrix and μ is a mean vector. The covariance matrix Σ is used to control the correlation between the synthetic features. The constraints for the maximum level of allowed covariance depend on the specification of the DGP. These features are at a later stage transformed via a quantile transformation of a χ_1^2 distribution to introduce non-linearity to the classification task. Categorical features are created through converting some numerical variables into distinct categories. The case probabilities convert the numerical features into categorical ones and preserve the underlying dependence structure of the data. To keep the DGP as close to a real-world application as possible, these features are encoded as dummy variables for use in the neural network and are also multiplied with the discriminator D in dummy form. The rationale is that these features should represent an indicator, for example group or sector membership, not linearly additive factors. In addition to the relevant features described above, the data-generating process also includes spurious features in certain settings. Spurious features are correlated to the relevant features through the multivariate normal distribution described in Equation (11), but do not influence the target variable via the discriminant D. This increases the complexity of the prediction or classification task for the neural network and the feature importance attribution. Irrelevant features are sampled from an independent distribution and therefore are statistically independent from the relevant and spurious features. They do not influence the target variables Y and can therefore be considered completely irrelevant. This differentiates them from spurious features which do not influence the target directly but may not be statistically independent from it. In theory, uncorrelated feature with an importance score of zero should be easy to detect.

4.3 Neural Network Hyperparameterisation

The neural network trained on the data is a three-layer, fully-connected neural network. It consists of three hidden layers with 5,4,2 nodes respectively, and a batch normalisation layer before every hidden layer. It is trained on 80% of the data set. The remaining 20% are used for testing and as input for the feature importance algorithms. The network is trained for 200 epochs per experiment. The sigmoid activation function is used in every layer. As described in Section 3, the sigmoid function is chosen because other activation functions may not be suited for all feature importance methods . An overview of the hyperparameterisation is provided in Table 3.

Table 2: The DGP can reflect six different cases with increasing data complexity. The columns specify the number of total, spurious, and irrelevant features as well as the correlation bounds.

case	features	spurious	irrelevant	correlation
I	20	0	0	[-0.3,0.3]
II	20	0	0	[-1,1]
III	30	10	0	[-0.3,0.3]
IV	30	10	0	[-1,1]
V	35	10	5	[-0.3,0.3]
VI	35	10	5	[-1,1]

4.4 Importance Attribution Methods

Most of the feature importance methods require some hyperparameterisation, for example the setting of baseline values against which the importance attribution is calculated. In this example, the baseline is set to 1, because a χ_k^2 distribution has mean k (in this case, $k = 1$). Additional possible configurations include using the training data as baselines for importance attribution in the test set. The perturbation function passed to the infidelity and sensitivity calculation is given as

$$f_{\text{pert}}(x_i) = \tilde{x}_i = x_i - \eta \quad \forall i \in \{1,\ldots,n\} \qquad (12)$$

where x_i is one observation of the input and

$$\eta \sim N(0, 0.0009). \qquad (13)$$

The perturbation function has been chosen in accordance with the usage tutorials of the software package used for feature importance calculation. Numerical experiments show that the choice of perturbation function does not significantly affect the computations and results.

Table 3: Hyperparameters for the neural network trained on the synthetic and empirical data.

hyperparameter	parameterisation
hidden layers and nodes	5,4,2
learning rate	0.003
batch size	50
epochs	200
dropout probability	0.2
activation function	sigmoid

4.5 Data Cases

There are six different scenarios for data creation and the underlying DGP. The general idea is to replicate data structures and anomalies that are most prominent in real-world data with the two most frequent anomalies being high levels of correlation and spurious features. Table 2 shows an overview of the relevant characteristics. Case I data is weakly correlated with correlation bounds given through the covariance matrix Σ with every off-diagonal entry being $\in [-0.3, 0.3]$. The total number of features is 20 with 17 discretely distributed features, three categorical ones and no redundancies. Case II data has 20 variables with no spurious features and correlation bounded by $[-1, 1]$. Case III data has 30 features, 17 relevant continuous ones, 3 relevant discrete ones, and 10 spurious features. Because of the constraints of positive semi-definite matrices for the covariance, the maximum correlation in the off-diagonals is lower than the numerical bounds. Case IV data combines complications and characteristics from case II and III such that the data is correlated and the some of the numerical features are spurious. This should imitate the presence of both data anomalies, as would most likely be expected in a real-world application. The last data anomaly — as represented in case V and VI — is the presence of irrelevant features. These features are statistically independent from the relevant and spurious $\mathbf{X}$ and the target Y. The features are drawn from a separate multivariate distribution.

4.6 Performance Evaluation

This section briefly explains how the performance of the feature importance methods is measured in the synthetic and the empirical setup. In the study of synthetic data with known ground truth, the performance of the feature importance methods can be evaluated directly. In the empirical application, the importance attribution performance cannot be measured directly because the causal link between Y and $\mathbf{X}$ is unknown.

Synthetic Data. This paragraph summarises how performance is measured when the neural network is trained on synthetic data. Four different components are of interest: The first one is the performance of the neural network in the prediction task. The second component is the determination of the feature importance methods with respect to the true underlying causal link D. The two remaining components are the infidelity and sensitivity of the importance attribution. To ensure the robustness of results, the experiment described in the previous section is conducted repeatedly. The neural network is retrained and the importance attributions are calculated 100 times on a resampled or newly generated data set. The first component can be measured using R^2 — as one usually would in the context of any regression task. The second component (feature importance determination) is evaluated by regressing the discriminant D on $\Theta = \{\theta_1, \dots, \theta_m\}$, which is a vector containing the importance attributions of every feature. These attributions can be calculated as

$$\theta_i = \frac{\sum_{i=0}^{n} \theta_{i,d}}{\sum_{i=0}^{n} |\theta_{i,d}|} \quad \forall i \in \{1, \dots, D\} \tag{14}$$

which is an aggregation of individual attributions to obtain a global impression of the data. The numerator is the sum of the attributions, while the denominator is the L1 norm of the attributions. This gives the normalised sum of attributions. Given that D represents the underlying causal structure of the DGP, the score or the coefficient of determination

$$R_i^2 = 1 - \frac{\sum_{i=1}^{m} (\theta_i - d_i)^2}{\sum_{i=1}^{m} \left(d_i - \bar{d}_i\right)^2}, \tag{15}$$

where ϕ_i is the attribution of an individual feature, d_i is the true importance of the synthetic feature, and $\bar{d}_i$ is the mean importance. This approximates how well the average importance attribution serves as a proxy of true global importance if ϕ_i is the importance attribution for an individual feature, whereas d_i is the true causal importance of the feature. Ideally, the R^2 of the regression on the determinant should be as close to 1 as possible, assuming that the feature importance coefficients describe the underlying data structure sufficiently well. To determine the reliability of the importance attribution, two components have to be studied further. First, the fidelity of the attributions with respect to the original model. Second, the sensitivity of the attributions with respect to input perturbations. Two appropriate measures, infidelity and sensitivity of explanations, have been introduced in previous research (Yeh et al., 2019). Infidelity is the expected mean-squared error between the explanation multiplied by a meaningful input perturbation and the differences between the predictor function at its input and its perturbed input. Infidelity is derived from the completeness property, a property or axiom all importance methods share. It requires that the difference between the output of f at input x and x^0 must be equivalent to the importance attributions. The infidelity $\mathrm{INFD}(\Phi, f, x)$ is formally defined as

$$E_{\eta \sim N}\left[\left(\eta^T \Phi(f, x) - (f(x) - f(x - \eta))\right)^2\right] \tag{16}$$

for a black-box function f, an explanation function Φ, a random variable η distributed as described in

Equation (13). The explanation function Φ is one of the feature importance methods introduced beforehand. The calculation of infidelity requires the trained model, the attribution, a perturbation function for the inputs, and the attributions calculated by the importance method. Another interesting measure is sensitivity, which measures the extent of explanation change when the input is slightly perturbed. The relevant metric here is so-called maximum sensitivity which is computed using a black-box function f, an explanation function Φ, and a given input neighbourhood radius r. The maximum sensitivity $\mathrm{SENS}_{\mathrm{MAX}}(\Phi, f, x, r)$ is defined as

$$\max_{\|\tilde{x}-x\| \leqslant r} \| (\Phi(f,\tilde{x}) - \Phi(f,x)) \|, \tag{17}$$

where $\tilde{x}$ is a slightly perturbed variation of the input x in the neighbourhood of x with radius r. $\|\ldots\|$ is the Frobenius norm. In this application, the perturbation $\Delta(\tilde{x}, x)$ is distributed as the random variable described in Equation (13). Ideally, attribution methods should exhibit low amounts of infidelity and sensitivity. A lower infidelity score means that the explanation provided by the feature importance method closely aligns with the actual behaviour of the model, which is desirable. Sensitivity measures how much the explanation changes when small changes are made to the input. A lower sensitivity score means that the explanation is stable and does not drastically change due to minor changes in input, which is also desirable. For the discussion of results in Section 5, the results are reported as total infidelity and total sensitivity. Total infidelity and maximum sensitivity are computed for every feature of every observation and to make results comparable, these results are summed up and summarised as total infidelity and total sensitivity.

Empirical Data. This subsection summarises how performance is measured when the neural network is trained on empirical data. In the empirical application, a measure of predictive power is not available. Since the underlying causal link D cannot be measured and is unknown, the determination of the importance attribution cannot be computed. The other three measures (neural network performance, importance attribution infidelity and sensitivity) can be computed nonetheless. The latter two are used to compare the performance of the attribution methods in an empirical application.

5 SIMULATION STUDY RESULTS

This simulation study with known ground truth serves as a benchmarking tool for the capabilities of the importance attribution methods. The performance of the feature importance methods is measured as described in Subsection 4.6, while the neural network performance is summarised in Table 4. The column for the true R^2 shows the R^2 achieved by *Maximum Likelihood Estimation* (MLE) when the underlying functional form is known and the determinant D is estimated from the data. This estimated optimal R^2 should theoretically be smaller than the training R^2 (due to overfitting) and larger than the test R^2 (due to bias). The methods should ideally achieve high coefficients of determination and low levels of infidelity and sensitivity. The coefficients of determination for the feature importance calculation are presented hereafter for the different setups of the data-generating process. To ensure robustness of results, the experiment as described in Section 4 is repeated 100 times. For every experiment, a sample of synthetic data with 1000 observations is generated. The parameter specifications for the individual data cases can be found in Table 2.

Table 4: R^2 of neural network for 100 experiments including true R^2 shows satisfying in- and out-of-sample performance, as compared to the true R^2 of the DGP.

R^2	training	validation	test	true
case I	0.759	0.480	0.500	0.858
case II	0.811	0.605	0.610	0.856
case III	0.802	0.433	0.437	0.859
case IV	0.825	0.527	0.533	0.852
case V	0.804	0.405	0.423	0.866
case VI	0.835	0.509	0.501	0.862

Case I. For the first configuration, the correlation levels are low. This should make it easier for a correct importance attribution since multicollinearity is not an issue. The network's performance in terms of training, validation, and testing can be assessed from Table 4. The data appears to be well comprehended by the feature importance methods used. As can be inferred from Table 7, DeepLIFT shows the highest determination, closely followed by SVS. In terms of total infidelity, LIME stands out with the best performance among the methods. DeepLIFT, despite its strong determination, also presents noticeable infidelity. Regarding sensitivity, IG is best-in-class.

Case II. The second configuration does not generate any spurious features and allows for high correlation levels between the features in the data. As observed from Table 8, the network performance is not significantly impacted by the strong correlation in training, validation, and testing phases. Reviewing the results from Table 8, introducing correlation has a discernible impact on attribution performance. DeepLIFT maintains its lead in terms of determination. IG presents the lowest sensitivity, while LIME shows the lowest infidelity. Notably, there is a tangible drop in performance across all methods when compared to non-correlated data configurations. Furthermore, infidelity levels rise considerably for all methods. Total sensitivity, as well, sees an uptick. Backward-based methods such as DeepLIFT and IG remain less sensitive to perturbations, while forward-based methods, including GradientSHAP, LIME, and SVS, show increased sensitivity.

Case III. For the third configuration, 10 additional features are added. These continuously distributed features have an importance score in the discriminator vector D of 0 and therefore do not influence the target directly. Since the features are weakly correlated, it is possible that there is non-zero correlation between the spurious features and the target. As seen in Table 4, the neural network performance for the training data slightly decreases from case II, and the validation and test performance decrease more significantly compared to case II. This decrease shows a wider performance gap between the training, validation, and testing stages. Despite this performance decrease, when comparing Table 8 and Table 9, the determination score does not decrease significantly. Most notably, DeepLIFT still outperforms the other methods in this category. Interestingly, determination increases slightly for all methods except LIME. This is most likely due to the fact that all methods can sufficiently well differentiate spurious features from important ones. In terms of infidelity, LIME exhibits the best performance, followed by IG. GradientSHAP has the highest sensitivity.

Case IV. Case IV data introduces 10 spurious features as well as possibly strongly correlated data. As is shown in Table 4, the delta between training and test performance is smaller compared to previous cases. When correlation is high and 10 of the 30 original features are spurious, determination — as measured by feature importance R^2 — decreases significantly. DeepLIFT still performs best, closely followed by SVS. Total infidelity does not change significantly from case III (Table 9) to case IV (Table 10). Total sensitivity significantly increases from case III to case IV data for GradientSHAP and LIME. Integrated Gradients and DeepLIFT show the smallest sensitivity, while the sensitivity of LIME and GradientSHAP is largest.

Case V. In case V, the data set is expanded to additionally include 5 irrelevant features. This change leads to diminution in the model's test and validation performance, compared to both cases II and III, as per Table 4. For determination of feature importance, SVS overtakes DeepLIFT and now performs best. When analysing total infidelity, DeepLIFT and GradientSHAP show the highest levels of infidelity, while SVS and LIME are the methods with the highest fidelity. Sensitivity shows comparable results to previous cases: Integrated Gradients and DeepLIFT perform well, while GradientSHAP and LIME are extremely sensitive.

Case VI. Case VI data is comparable to case IV data since both sets are created using comparable levels of correlation between relevant and irrelevant features. Case VI data also introduces irrelevant features. Determination, as shown in Table 12, decreases for all methods except LIME. SVS performs best with DeepLIFT as a close second. Integrated Gradients and GradientSHAP are close, while LIME performs worst. Total infidelity and total sensitivity remain unchanged in relative terms:

Summary. Table 5 shows the results for all experiments on synthetic and empirical data. For synthetic data, it is clear that neural network-specific methods, namely DeepLIFT and IG, perform best in terms for determination and sensitivity. LIME shows the lowest infidelity out of all the methods, even though the difference to other methods is not that big. Tables 7 to 12 show that spurious features, with zero causal importance, are less problematic than highly correlated data. Correlated data cause a significant drop in determination coefficients, as seen in Table 8, even when all features are relevant predictors. In contrast, case III, despite having one-third spurious features, still produced higher determination scores than case II. A similar pattern was found in case V (Table 11), where irrelevant features had little effect on the determination score. However, a mix of spurious features and high correlation in case IV caused a drop in importance scores (Table 8 IV). This suggests the methods used are robust against spurious features but struggle with correlated data. Irrelevant features, while not significantly impacting R^2, do affect the sensitivity of some methods, highlighting the need for preprocessing and dimensionality reduction to improve feature importance methods in practice.

Table 5: Best-in-class performance for synthetic and empirical data.

case	determination	infidelity	sensitivity
I	DeepLIFT	LIME	IG
II	DeepLIFT	LIME	IG
III	DeepLIFT	LIME	IG
IV	DeepLIFT	LIME	IG
V	SVS	LIME	IG
VI	DeepLIFT	LIME	IG
Ames		DeepLIFT	IG
California		DeepLIFT	IG
FF5		GSHAP	LIME

Table 6: R^2 of neural network for 100 iterations shows satisfying performance and no significant overfitting.

dataset	training	validation	test
Ames	0.699	0.693	0.698
California	0.603	0.570	0.592
FF5	0.195	0.131	0.133

6 EMPIRICAL RESULTS

The importance attribution methods previously used on synthetic data are applied to empirical data to provide further insights. The data sets used are the California house price dataset (Dheeru and Casey, 2017), the Ames house price data set (Cock, 2010), and the daily Fama-French-5 (FF5) factor dataset (Fama and French, 2024). Since Section 5 showed that importance attribution is challenged by high levels of correlation, it is worthwhile to analyse correlation in the data at hand. As can be seen in the supplementary documents to this study, most of the features do not exhibit significant levels of correlation. However, some combinations of features do show high levels of correlation. Table 6 summarises the neural network performance for all datasets in training, validation, and testing. To compute and compare infidelity and sensitivity, the data from the corresponding data sets is resampled for 100 repeated experiments. The model is retrained for every resampling.

The results for 100 experiments on a sample with 1000 observations are shown in Tables 13 to 15. The table shows that while the infidelity is almost identical for all methods, the total sensitivity varies substantially. While the maximum sensitivity of DeepLIFT and IG is small, GSHAP and SVS show considerably higher total sensitivity. This is again in line with theoretical findings: Forward-based methods are much more sensitive to small input perturbations.

7 SUMMARY

This study shows that the five importance attribution methods that were reviewed in the paper generally perform sufficiently well at identifying relevant and spurious features as well as their magnitude. In a simulation study with underlying ground truth, the methods are able to approximate the underlying connection between the features and the target. Nonetheless, the simulation study shows that even in a simple neural network, the computed importance scores cannot perfectly explain the underlying causal structure. Additionally, not all methods are alike, and DeepLIFT predictive performance outperforms the other methods for most data cases with SVS coming in as a close second. For more complex cases, SVS sometimes outperforms DeepLIFT. Total infidelity of DeepLIFT and Integrated Gradients is usually higher, while the other methods show a somewhat lower infidelity. It is especially noteworthy that infidelity of Integrated Gradients and GSHAP sharply increased for more complicated data structures. Total sensitivity of LIME, GSHAP, and SVS is extremely high for all classes under observation. On the other hand, DeepLIFT and Integrated Gradients show low sensitivity for all four data cases. This means that the attributed importance of the features varies less when the inputs are slightly perturbed. This is in line with the current literature. Forward-based methods are more sensitive to input perturbations than backward-based methods. Adding irrelevant features to the data substantially increases sensitivity of GSHAP and LIME. In the empirical application, these results are partly replicated. While neural network-specific methods, namely IG and DeepLIFT perform well for the California house price and the Ames house price data set, they are not best-in-class for the FF5 data. On the one hand, this shows that model-specific attribution methods can provide substantial advantages over model-agnostic methods. On the other hand, the suitability of the method heavily depends on the specific task at hand as well as the neural network hyperparameters. The results discussed in Section 5 and 6 suggest that the feature importance methods work well in understanding the relationships between variables in synthetic data. However, complex patterns in the data make it difficult to accurately determine which variables are most important, especially in real-world data. This is particularly problematic for data with a high degree of correlation. In more general terms, it is important to observe that feature importance methods are not a one-size-fits all toolbox. Before deciding on which feature importance algorithms to use, it is necessary to consider the underlying model specifications.

7.1 Limitations

This research paper is only a limited analysis of the aspect of feature importance in deep neural networks. Multiple things should be considered. First, the choice of baseline values significantly influences the feature importance calculation. In the case of image analysis, research has shown how heavily the impact of a feature attribution baseline influences the computed importance scores (Sturmfels et al., 2020). This is no different for data such as the one used in this paper. Additionally, the data generation process used in this paper leads to limited generalisability - some of the results may be different for differing setups. Finally, the neural network architecture heavily influences the predictive power of a neural network. For example, choosing a ReLu activation function (instead of the sigmoid function in this paper) can lead to significant problems for the utilised gradient-based attribution methods. Another component to consider is the computational burden implied by feature importance attributions. The computations in this paper were done on small-scale model. For bigger models, the computational complexity increases drastically. For large-scale, industry-standard applications, this is an important factor to consider.

7.2 Extensions and Practical Implications

This paper has given valuable insights into the predictive power, infidelity, and sensitivity of various feature importance methods. Future research could implement the insights gathered into practical research in economics and finance. For example, adding metrics of uncertainty to feature importance can enhance the insights gathered. Furthermore, seeing the robustness of feature importance calculations over repeated experiments can also be beneficial in the realm of variable selection. Additionally, it creates a baselines for researchers to understand that feature importance methods are not one-size-fits-all solutions and should therefore be chosen and calibrated carefully. Additionally, It is important to note that most current studies of feature importance focus mainly on image data. In these cases, it is easy to visually check if the important areas identified by the network make sense. But this kind of validation does not work for purely numerical data. This means that feature importance in finance and economics requires additional sanity checks. Paths for new research in these fields could concentrate on feature importance validation using domain-specific knowledge. Looking forward, the field of user-friendly and explainable machine learning can additionally benefit from studying the balance between performance, complexity, and explainability. As neural networks get more complex with deeper architecture, they become harder to understand. This is not in the best interest of the researchers and general stakeholders if model predictions should also stay explainable. This implies that future research could focus on smaller, sparse models, that still offer a certain level of interpretability. Looking beyond feature-based attribution methods, researchers should also focus on more high-level, human-friendly XAI and XML methods. For example, *Testing with Concept Activation Vectors* (TCAV) (Kim et al., 2018), which explain importance not by features, but by so-called concepts. Instead of focusing on single features, TCAV analyses higher level ideas in the data. In TCAV, these concepts are typically identified using a separate model. This model is often a different neural network which is trained specifically to recognise the concepts in the model's internal representations of the data. This can bring XAI closer to explanations that are easily understandable and therefore human-centric. Especially for high-dimensional input data, the importance of a single feature may not be important for humans to understand the prediction model. They would probably be much more interested in the importance of a concept for predictions. Since this methodology is not yet popular in research in finance and economics, it may be worthwhile to pursue research concerning concept importance.

ACKNOWLEDGEMENTS

I thank Prof. Dr. Dietmar Maringer and Prof. Dr. Sabine Böckem for their valuable feedback and suggestions on the paper.

REFERENCES

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature News*, 538(7623):20–23.

Castro, J., Gómez, D., and Tejada, J. (2009). Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730.

Chen, J.-F., Chen, W.-L., Huang, C.-P., Huang, S.-H., and Chen, A.-P. (2016). Financial time-series data analysis using deep convolutional neural networks. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 87–92. IEEE.

Cock, D. D. (2010). Housing prices in Ames, Iowa. https://www.openintro.org/data/index.php?data=ames. Accessed: 2024-09-12.

Crawford, L., Flaxman, S. R., Runcie, D. E., and West, M. (2019). Variable prioritization in nonlinear black box methods: A genetic association case study. *The Annals of Applied Statistics*, 13(2):958.

Dheeru, D. and Casey, G. (2017). UCI machine learning repository. Accessed: 2024-09-12.

Fama, E. F. and French, K. R. (2024). Fama/French 5 factors. https://mba.tuck.dartmouth.edu. Accessed: 2024-09-12.

Gevrey, M., Dimopoulos, I., and Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling*, 160(3):249–264.

Ish-Horowicz, J., Udwin, D., Flaxman, S., Filippi, S., and Crawford, L. (2019). Interpreting deep neural networks through variable importance. *arXiv preprint arXiv:1901.09839*.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677. PMLR.

Kumar, E. I., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. (2020). Problems with Shapley-value-based explanations as feature importance measures. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5491–5500. PMLR.

Liang, D., Tsai, C.-F., and Wu, H.-T. (2015). The effect of feature selection on financial distress prediction. *Knowledge-Based Systems*, 73:289–297.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

Molnar, C. (2020). *Interpretable Machine Learning*. Lulu.com.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). 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*, pages 1135–1144.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3145–3153. PMLR.

Strumbelj, E. and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18.

Sturmfels, P., Lundberg, S., and Lee, S.-I. (2020). Visualizing the impact of feature attribution baselines. *Distill*. Published: 2020-01-10, Accessed: 2024-09-12.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR.

Xiaomao, X., Xudong, Z., and Yuanfang, W. (2019). A comparison of feature selection methodology for solving classification problems in finance. In *Journal of Physics: Conference Series*, volume 1284, pages 12–16. IOP Publishing.

Yeh, C.-K., Hsieh, C.-Y., Suggala, A. S., Inouye, D. I., and Ravikumar, P. (2019). On the (in)fidelity and sensitivity for explanations. *arXiv preprint arXiv:1901.09392*.

APPENDIX

The appendix contains the experiment results described in the paper. Note that the results are rounded to three decimals unless a higher grade of granularity is required for comparisons (most often for the infidelity metrics).

Table 7: Mean results (Standard Deviation) for 100 repeated experiments on case I data. Best-in-class values are marked with *.

	determination	infidelity	sensitivity
IG	0.186	0.0035	2.782*
	(0.118)	(0.0020)	(0.822)
GSHAP	0.183	0.0037	21808.902
	(0.118)	(0.0021)	(41842.360)
LIME	0.107	0.0025*	921.865
	(0.093)	(0.0012)	(5259.562)
SVS	0.223	0.0028	142.402
	(0.144)	(0.0013)	(87.815)
DeepLIFT	0.236*	0.0040	7.512
	(0.114)	(0.0019)	(1.200)

Table 8: Mean results (Standard Deviation) for 100 repeated experiments on the case II data. Best-in-class values are marked with *.

	determination	infidelity	sensitivity
IG	0.129	0.0033	2.872*
	(0.108)	(0.0018)	(0.911)
GSHAP	0.126	0.0036	24776.439
	(0.106)	(0.0018)	(118520.80)
LIME	0.101	0.0026*	434.697
	(0.089)	(0.0012)	(545.340)
SVS	0.146	0.0029	144.623
	(0.119)	(0.0014)	(79.661)
DeepLIFT	0.174*	0.0041	7.537
	(0.108)	(0.0020)	(1.441)

Table 9: Mean results (Standard Deviation) for 100 repeated experiments on case III data. Best-in-class values are marked with *.

	determination	infidelity	sensitivity
IG	0.153	0.0030	2.816*
	(0.117)	(0.0010)	(0.812)
GSHAP	0.148	0.0030	30712.477
	(0.116)	(0.0010)	(77256.110)
LIME	0.066	0.0020*	446.402
	(0.071)	(0.0010)	(451.300)
SVS	0.174	0.0030	176.937
	(0.136)	(0.0010)	(135.055)
DeepLIFT	0.183*	0.0040	7.269
	(0.111)	(0.0010)	(1.217)

Table 10: Mean results (Standard Deviation) for 100 repeated experiments on case IV data. Best-in-class values are marked with *.

	determination	infidelity	sensitivity
IG	0.116	0.0030	2.591*
	(0.104)	(0.0014)	(0.674)
GSHAP	0.118	0.0032	71441.851
	(0.103)	(0.0014)	(492030.40)
LIME	0.069	0.0022*	970.022
	(0.066)	(0.0009)	(4515.176)
SVS	0.137	0.0025	173.937
	(0.112)	(0.0010)	(120.617)
DeepLIFT	0.142*	0.0035	6.835
	(0.101)	(0.0015)	(0.996)

Table 11: Mean results (Standard Deviation) for 100 repeated experiments on case V data. Best-in-class values are marked with *.

	determination	infidelity	sensitivity
IG	0.141	0.0029	2.517*
	(0.112)	(0.0013)	(0.752)
GSHAP	0.139	0.0031	93285.070
	(0.114)	(0.0015)	(599034.40)
LIME	0.052	0.0020*	549.692
	(0.058)	(0.0009)	(1091.150)
SVS	0.179*	0.0023	165.701
	(0.137)	(0.0010)	(119.806)
DeepLIFT	0.154	0.0032	6.700
	(0.086)	(0.0016)	(1.035)

Table 12: Mean results (Standard Deviation) for 100 repeated experiments on case VI data. Best-in-class values are marked with *.

	determination	infidelity	sensitivity
IG	0.153	0.0030	2.816*
	(0.117)	(0.0015)	(0.812)
GSHAP	0.148	0.0032	30712.477
	(0.116)	(0.0015)	(77256.110)
LIME	0.066	0.0023*	446.402
	(0.071)	(0.0009)	(451.300)
SVS	0.174	0.0025	176.937
	(0.136)	(0.0010)	(135.055)
DeepLIFT	0.183*	0.0036	7.269
	(0.111)	(0.0015)	(1.217)

Table 13: Mean results (Standard Deviation) for 100 repeated experiments on the Ames housing price data. Best-in-class values are marked with *.

	infidelity	infidelity
IG	0.000591	10.586*
	(0.00020)	(1.751)
GSHAP	0.000597	309.495
	(0.00020)	(230.412)
LIME	0.000591	321.872
	(0.00020)	(1171.460)
SVS	0.000596	32.218
	(0.00020)	(5.283)
DeepLIFT	0.000587*	17.458
	(0.00021)	(3.197)

Table 14: Mean results (standard deviation) for 100 repeated experiments on the California housing price data. Best-in-class values are marked by *.

	infidelity	sensitivity
IG	0.093	26.609*
	(0.053)	(14.956)
GSHAP	0.092	655.637
	(0.054)	(885.336)
LIME	0.092	260.244
	(0.053)	(252.150)
SVS	0.093	62.229
	(0.055)	(21.680)
DeepLIFT	0.091*	64.089
	(0.055)	(28.282)

Table 15: Mean results (Standard Deviation) for 100 repeated experiments on the FF5 data. Best-in-class values are marked with *.

	infidelity	infidelity
IG	0.000151	23.564
	(0.00006)	(2.209)
GSHAP	0.000149*	312.104
	(0.00006)	(78.938)
LIME	0.000150	0.887*
	(0.00006)	(3.714)
SVS	0.000152	51.652
	(0.00006)	(12.362)
DeepLIFT	0.000152	33.935
	(0.00006)	(4.978)