

# Time to Focus: A Comprehensive Benchmark using Time Series Attribution Methods

Dominique Mercier<sup>1,2</sup><sup>a</sup>, Jwalin Bhatt<sup>2</sup>, Andreas Dengel<sup>1,2</sup><sup>b</sup> and Sheraz Ahmed<sup>1</sup><sup>c</sup>

<sup>1</sup>German Research Center for Artificial Intelligence GmbH (DFKI), Kaiserslautern, Germany

<sup>2</sup>Technical University Kaiserslautern (TUK), Kaiserslautern, Germany

**Keywords:** Deep Learning, Time Series, Interpretability, Attribution, Benchmarking, Convolutional Neural Network, Artificial Intelligence, Survey.

**Abstract:** In the last decade neural network have made huge impact both in industry and research due to their ability to extract meaningful features from imprecise or complex data, and by achieving super human performance in several domains. However, due to the lack of transparency the use of these networks is hampered in the areas with safety critical areas. In safety-critical areas, this is necessary by law. Recently several methods have been proposed to uncover this black box by providing interpretation of predictions made by these models. The paper focuses on time series analysis and benchmark several state-of-the-art attribution methods which compute explanations for convolutional classifiers. The presented experiments involve gradient-based and perturbation-based attribution methods. A detailed analysis shows that perturbation-based approaches are superior concerning the Sensitivity and occlusion game. These methods tend to produce explanations with higher continuity. Contrarily, the gradient-based techniques are superb in runtime and Infidelity. In addition, a validation the dependence of the methods on the trained model, feasible application domains, and individual characteristics is attached. The findings accentuate that choosing the best-suited attribution method is strongly correlated with the desired use case. Neither category of attribution methods nor a single approach has shown outstanding performance across all aspects.


## 1 INTRODUCTION


For several years, the field of artificial intelligence has shown a growing interest in both research and industry (Allam and Dhunny, 2019). This attention led to the discovery of crucial limitations and weaknesses when dealing with artificial intelligence. The following main concerns have become increasingly important: resource management, efficiency, data security, but also interpretability and explainability. According to (Perc et al., 2019) these limitations originate from the social and the juristic domain.


Particularly the interpretability of the classifiers' decisions plays a crucial role in industry and safety-critical application areas. The legal situation reinforces the significance of interpretability. In the medical sector, financial domain, and other safety-critical areas (Bibal et al., 2020) explainable computations are required.

Over several years, a wide range of methods to explain neural networks was summarized by (Došilović et al., 2018). These methods involve both intrinsic and post-hoc approaches across a broad scope of modalities involving language processing, image classification, and time series analysis. The majority of these approaches have origin from image analysis since the visual criteria (Zhang and Zhu, 2018) and concepts are more intuitive for humans.

Due to the lack of evaluations of the existing approaches in the context of time series, the paper concentrates on their applicability and effectiveness in time series analysis. A comprehensive analysis of existing attribution methods as one class of commonly used interpretability methods is presented. The paper further covers the strengths and weaknesses of these methods. Specifically, a runtime analysis is done, which is relevant for real-time use cases. Besides the computational aspects, the Infidelity, Sensitivity, influence on accuracy, and correlations between the attributions were evaluated. For this purpose, AlexNet was used as architecture and experiments on well-

<sup>a</sup>  <https://orcid.org/0000-0001-8817-2744>

<sup>b</sup>  <https://orcid.org/0000-0002-6100-8255>

<sup>c</sup>  <https://orcid.org/0000-0002-4239-6520>

known and freely available time series datasets were executed.

The contribution includes a comprehensive analysis of several state-of-the-art attribution methods concerning runtime, accuracy, robustness, Infidelity, Sensitivity, model parameter dependence, label dependence, and dataset dependence. The findings illustrate the superior performance of gradient-based methods concerning runtime and Infidelity. In contrast, perturbation-based approaches give better results concerning the Sensitivity, occlusion game, and continuity of the attribution maps. The paper emphasizes that none of the two categories is superior in all evaluated characteristics and that the selection of the best-suited attribution methods depends on the desired properties of the use case.

## 2 RELATED WORK

Often Attribution methods are used to interpret classifiers. A comprehensive overview of the different categories involving attribution methods is given by Das et al. (Das and Rad, 2020). Attribution methods are well-known as they are compatible with various networks and therefore do not require any restrictions in the design of the network. Attribution methods belong to the class of posterior techniques that require less cognitive effort to interpret due to their simple visualization of the relevance of the input. Furthermore, no detailed knowledge about the analyzed classifier is needed. Especially for image classification, there is a wide range of attribution methods and different benchmark works. According to the authors of (Abdul et al., 2020), an explanation always results in a trade-off between accuracy, simplicity, and cognitive effort that is one reason for the popularity of the attribution methods.

Aspects like the Sensitivity, the change of the attribution map by permutation of the input signal, and other metrics are applied to understand the exact advantages and disadvantages of the methods. More details about the importance and impact of Sensitivity are summarized by Ancona et al. (Ancona et al., 2017). Besides Sensitivity, Infidelity, known as the change in classification when permutating the input, plays a role. According to Yeh et al. (Yeh et al., 2019), Infidelity serves a pivotal role in explaining the quality of an attribution method. Further aspects are the runtime and the difference between black box and white box requirements.

Also, aspects like the dependency on gradient calculation play a big role. Some methods work without backpropagation and use permutations and the

forward pass to calculate the relevance of the input points. A detailed differentiation of these categories can be provided by Anaconda et al. (Ancona et al., 2019) and Ivanovs et al. (Ivanovs et al., 2021).

The experiments are aligned with existing image processing surveys and used similar metrics. A comprehensive analysis for the image modalities was written by Adebayo et al. (Adebayo et al., 2018). Although this paper used similar experiment settings, the results may differ due to the diverse modalities.

However, the precise evaluation of these methods in the time series domain is crucial. Karliuk mentioned that (Karliuk, 2018) it was legally stipulated that neuronal networks, for example, may not be used in all areas of life as their interpretability and ethical problems still exist. Peres et al. (Peres et al., 2020) discussed which aspects are relevant for the application of neural networks in the economy. In addition to data protection restrictions and efficiency, the interpretability of neural networks plays a pivotal role, especially today.

## 3 EVALUATED METHODS

This section provides an overview of the different methods, their applicability, and categorization. First of all, the used methods are a subset that can be used in the field of time series analysis and do not require the selection of internal layers for calculation.

### 3.1 Gradient-based

Gradient-based methods include Integrated Gradients, Saliency maps, InputXGradient, GradientShap (Lundberg and Lee, 2017) and Guided-Backprop. In the case of Integrated Gradients (Sundararajan et al., 2017) backpropagation is applied to calculate an importance value for each input value relative to a baseline. An elementary part of this method is to know the baseline. The selection of this baseline is crucial for the computation of the gradients to make sense. In contrast, the Saliency (Simonyan et al., 2013) does not need a baseline and only computes the gradients. A method that is very similar to this is called Input X Gradient (Shrikumar et al., 2016). Here the calculated gradients are multiplied by the input to create a relation between them and the input values. Guided-Backpropagation (Springenberg et al., 2014) also uses a backward run to compute the importance of the values. However, a modification to the network is required. The resulting limitation is the access to the activation function to modify it. Previously mentioned methods require a backward calcu-

lation leading to noisy explanations due to the gradients. In addition, they need to access internal parameters. The core concept of GradientShap relies on the estimation of the SHAP values of the input. SHAP values are estimated using targeted permutations of the input sequence. These values are an approximation since the exact calculation of the SHAP values is very time and resource-intensive. GradientShap is in this respect very similar to Integrated Gradients.

### 3.2 Perturbation-based

These methods are different to the gradient based methods, as they do not need access to the gradients. Perturbation-based methods slightly change the input and compare the output to the baseline to create an importance ranking. Example approaches for this category are Occlusion (Zeiler and Fergus, 2014) and Feature Permutation (Fisher et al., 2019) and Feature Ablation. All these methods differ in the way they modify the individual points. Another method that makes use of the perturbation principle is Dynamask (Crabbé and van der Schaar, 2021). A mask gets learned utilizing permutations to calculate the relevant input values. Apart from Dynamask, the above methods have the advantage that no backpropagation and thus no full access to the network and the parameters is required. Dynamask particularly allows easy visualization and restriction to a percentage of the features. The disadvantages of these methods are the correct choice of permutation depending on the dataset. In addition, the increased runtimes due to the multiple forward passes are negative too.

### 3.3 Miscellaneous

Shapley Value Sampling (SVS) (Mitchell et al., 2021) is based solely on a random permutation of the input values. The influence on the output is determined utilizing multiple forward calculations. Using SVS requires further points in addition to the data point under consideration to be changed. Finally, Lime (Ribeiro et al., 2016) tries to explain the model using a local model trained on perturbed input samples related to the original input to train an interpretable model and create importance values based on this model.

## 4 DATASETS

For the experiments a subset of the datasets from UEA & UCR (Bagnall et al., 2021) repositories was used. The selected datasets cover different aspects such as a

Table 1: UEA & UCR Datasets related to critical infrastructures.

Domain & Dataset	Train	Test	Steps	Channels	Classes
<b>Communications</b>					
UWaveGestureLibraryAll	896	3,582	945	1	8
<b>Critical manufacturing</b>					
FordA	3,601	1,320	500	1	2
Anomaly	35,000	15,000	50	3	2
<b>Public health</b>					
ECG5000	500	4,500	140	1	5
FaceDetection	5,890	3,524	62	144	2
<b>Telecommunications</b>					
CharacterTrajectories	1,422	1,436	182	3	20

variance in the number of channels, sequence length, classes, and task. The tasks include point anomaly and sequence anomaly classification in which an occurrence of a single anomalous point is enough to change the label. Furthermore, the datasets cover traditional sequence classification not related to atypical behavior. These datasets are taken from different critical domains that require explainability and in addition privacy. In addition, to the UEA & UCR datasets, The point anomaly dataset proposed by Siddiqui et al. (Siddiqui et al., 2019) was included as it is unique compared to the others, and a perturbation on single points can change the complete prediction. Table 1 lists the different datasets used in this paper.

## 5 EXPERIMENTS & RESULTS

In this section, different aspects of the above methods are evaluated. The methods were not optimized to ensure fairness among the approaches. Fine-tuning an attribution method requires assumptions about the dataset. However, in a real case, this prior knowledge is not necessarily given. The work covers the following aspects: Impact on the accuracy, Infidelity, Sensitivity, runtime, the correlation between the methods, and impact of label and model parameter randomization. In existing work such as (Adebayo et al., 2018; Huber et al., 2021; Nielsen et al., 2021) these measurements are judged as significant.

In general, all experiments are executed for the previously mentioned datasets. However, identical results were excluded due to the limited space and the low amount of insights they provide to the reader. The preprocessing of the data covers a standardization to achieve a mean of zero and a standard deviation. Therefore, the baseline signal is a sequence of zeros. AlexNet was modified to work with 1D data and trained the network using an SGD optimizer and a learning rate of 0.01 to evaluate the different attribution techniques. In Table 2 the network structure of the AlexNet is shown. The layer names used in the rest of the paper refer to those mentioned in the architecture figure. All networks were trained for a maxi-

Table 2: **Architecture.** AlexNet architecture includes layer names used in this paper. Dropout layers are excluded from the table. The padding of every layer was set to 'same'. The variables 'c', 'w', and 'l' depend on the input channels, width, and the number of classes of the used dataset.

Name	Type	In	Out	Size	Stride
conv_1	Conv, ReLu, Batch	<i>c</i>	96	11	4
pool_1	MaxPool	96	96	3	2
conv_2	Conv, ReLu, Batch	96	256	5	1
pool_2	MaxPool	256	256	3	2
conv_3	Conv, Relu, Batch	256	384	3	1
conv_4	Conv, Relu, Batch	384	384	1	1
conv_5	Conv, Relu, Batch	384	256	1	1
pool_2	MaxPool	256	256	3	2
dense_1	Dense, ReLu	$w * 256$	4,096		
dense_2	Dense, ReLu	4,096	4,096		
dense_3	Dense	4,096	<i>l</i>		

Table 3: **Accuracies.** Evaluation of the test data using the original split provided by the datasets. Subset covers the performance of the model on the 100 samples subset that is used in the rest of the paper due to the computational limitations. The values show the weighted-f1 scores and provide evidence that the difficulty of the sets is similar.

Dataset	Test Set	Attribution. Subset
Anomaly	0.9801	0.9464
CharacterTrajectories	0.9930	1.0000
ECG5000	0.9352	0.8907
FaceDetection	0.5956	0.7097
FordA	0.9204	0.9400
UWaveGestureLibraryAll	0.9318	0.9802

num of 100 epochs. In addition, the learning rate was reduced by half after a plateau and performed early stopping based on the validation set. In the particular case of label permutation, the labels of the training data were randomized. All experiments used fixed random seeds to preserve reproducibility.

Due to the immense computational effort, a set of 100 test samples was selected to evaluate the attribution methods. In addition, these samples preserve the class distribution of the test set. In Table 3 the weighted f1 scores are shown. The differences in the weighted-f1 scores between the original data and the subsets are less than 5%. Only the FaceDetection dataset shows a difference of 19%. This difference does not hinder the analysis as those two sets are never compared.

## 5.1 Impact on the Accuracy

To evaluate the performance of the attribution methods, the drop in accuracy under the addition and occlusion of the data points was inspected. To occlude the data, the points were set to zero as this is the mean of the data corresponding to the baseline. Respectively, the start point is zero when adding points step-wise. This experiment was performed in both directions adding important points and insignificant data. In Figure 1 the results show that

most of the methods were able to correctly identify the data points that have the most influence on the accuracy. Intuitively, data points that have a higher impact on accuracy should be ranked higher. The top row shows the accuracy increase adding the most significant points step-wise. The bottom row shows the behavior of adding the insignificant data points first. Ultimately, reading each plot starting from 100 to 0 percent results in excluding the least important ones for the top row and most important ones for the bottom row. The experiments highlight that for most datasets, namely Anomaly, CharacterTrajectories, ECG5000, and UWaveGestureLibraryAll, a small number of data points is enough to recover the accuracy. Surprisingly, adding unimportant data points resulted in higher accuracy values. Examples of this behavior are the Lime, Saliency, and Dynamask approach. This behavior appears in the ECG5000, FordA, and UWaveGestureLibraryAll datasets. Saliency has shown to suffer from the noisy backpropagation. The drawbacks of Lime and Dynamask are their hyperparameters. These are the number of neighborhood samples for Lime and the area size and continuity loss for Dynamask.

## 5.2 Prediction Agreement

In addition to the accuracy drops, the agreement with the original data was computed. Therefore, In Table 4 the percentage of data required to produce a similar prediction as with the original sample are shown. To do so, data points are included step-wise based on their importance. Initially, all data samples start with zeros. In every step, the next most important data point was added. The results show that the required data for an agreement of 90% of the predictions is in most cases reached with far less than 50% of the data. The results show that the perturbation-based approaches overall performed better. In addition, the results show that the required amount of data highly differs based on the dataset. Intuitively, Dynamask did not perform well on this task as it provides only a binary decision on whether a feature is significant or not. Besides Dynamask, the Saliency and KernelShap have shown a worse performance too. On the other side, the FeatureAblation, FeaturePermutation, GuidedBackProp, and ShapleyValueSampling approaches have shown superior performance to the other methods using the data suggested to be important by those methods resulted in a much earlier agreement of the prediction. Interestingly, the point anomaly dataset has shown that highlighting only one percent of the data is enough to reach a 90% agreement. In addition, getting to a similar prediction for the UWaveGesture

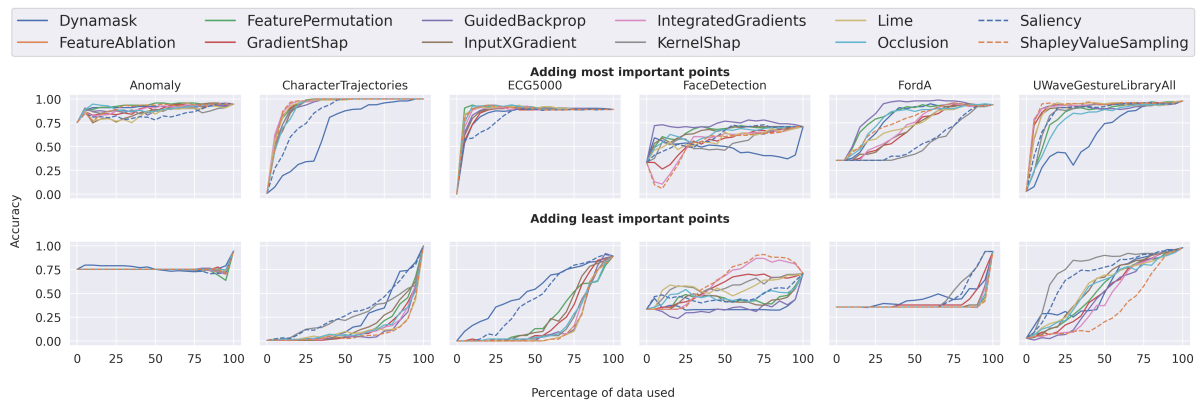


Figure 1: **Impact on accuracy.** Shows the impact when adding points to the baseline signal using the attribution scores as sequence order. **Top:** Shows the increase adding the most important points. **Bottom:** Shows the increase adding the least important points. Precisely, each plot read from 100 percent used data to 0 shows the impact removing the least important points for the top row, respectively the most important for the bottom row. The values show the weighted-f1 scores. Except Dynamask, Saliency, and KernelShap the performances of the approaches are similar.

Table 4: **Prediction agreement.** Evaluation of how many data points are required to reach a specific agreement between the original and modified input. All numbers are in percentage, and lower numbers are better as less data was needed to restore the ground-truth predictions. The numbers in each cell show the percentage of data points added to the baseline to achieve the required agreement concerning the prediction. Perturbation-based approaches have shown a significantly better performance.

Method	Anomaly			CharacterTraj.			ECG5000			FaceDetection			FordA			UWaveGesture		
	90	95	100	90	95	100	90	95	100	90	95	100	90	95	100	90	95	100
<b>Gradient-based</b>																		
GradientShap (Lundberg and Lee, 2017)	1	44	97	15	18	32	15	20	75	60	71	98	69	77	96	12	38	100
GuidedBackprop (Springenberg et al., 2014)	1	76	98	17	27	45	13	14	83	2	2	5	33	61	98	11	16	100
InputXGradient (Shrikumar et al., 2016)	1	51	92	16	21	29	18	24	42	26	36	55	69	81	98	12	38	100
IntegratedGradients (Sundararajan et al., 2017)	1	3	99	12	15	31	11	18	38	63	81	97	70	79	98	12	39	100
Saliency (Simonyan et al., 2013)	1	76	97	34	41	48	32	37	75	48	51	54	88	93	100	20	53	100
<b>Perturbation-based</b>																		
Dynamask (Crabbé and van der Schaar, 2021)	1	5	100	55	72	92	18	31	100	100	100	100	50	71	100	61	74	98
FeatureAblation (Zeiler and Fergus, 2014)	1	2	48	15	20	28	6	9	60	25	30	35	44	52	82	26	55	99
FeaturePermutation (Fisher et al., 2019)	1	2	48	15	20	28	6	9	60	25	30	35	44	52	82	26	55	99
Occlusion (Zeiler and Fergus, 2014)	1	3	83	19	20	29	9	15	46	16	47	87	43	55	96	33	68	100
<b>Others</b>																		
KernelShap (Lundberg and Lee, 2017)	1	58	100	15	22	43	8	15	84	70	84	99	90	94	98	16	34	100
Lime (Ribeiro et al., 2016)	1	90	100	15	17	49	8	17	75	49	52	81	79	86	99	13	17	100
ShapleyValueSampling (Mitchell et al., 2021)	1	30	51	12	13	30	10	18	71	68	90	93	65	79	97	9	15	100

dataset required every method to include almost every point.

### 5.3 Infidelity & Sensitivity

The Infidelity measurements provide information about the change concerning the predictor function when perturbations to the input are applied. The metric derives from the completeness property of well-known attribution methods and is used to evaluate the quality of an attribution method. In the results in Table 5 the Infidelity represents a mean error using 100 perturbed samples for each approach. A lower Infidelity value corresponds to a better attribution method, and the optimal Infidelity value should be zero. The results show that the tested methods do differ by a large margin of less than 7.2% on average, and in addition, the Infidelity values strongly depend on the dataset. Neither the gradient-based approaches

nor the perturbation-based or other approaches are superior. The mean increase of the worst-performing and the best method was 7.2%. The experiments identified the highest increases for the CharacterTrajectories dataset (15.8%) and the lowest for the FordA (3.4%).

Further, the Sensitivity of the methods for a single sample was compared. Computationally, the Sensitivity is much more expensive but provides a good idea about the change in the attribution when the input is perturbed. Using the Sensitivity the robustness against of the methods concerning noise was evaluated. Ultimately, an attribution method tends to show low Sensitivity, although this depends on the model itself. In Table 6 the results of the Sensitivity for all methods are presented. The results show that Dynamask has a Sensitivity of zero. Dynamask by design forces the importance values to be either one or zero. Although this is a benefit concerning the Sen-

Table 5: **Infidelity comparison.** Computed values show the average Infidelity over the 100 sample subsets. Results show differences between the different methods when applied to time series data. No category has shown a superior performance, although the gradient-based approaches were slightly better.

Method	Anomaly	CharacterTrajectories	ECG5000	FaceDetection	FordA	UWaveGestureLibraryAll
<b>Gradient-based</b>						
GradientShap	2.3803	<b>1.1408</b>	<b>0.7897</b>	0.0014	1.3734	11.4717
GuidedBackprop	2.4057	1.1665	0.8060	0.0014	1.3782	11.6886
InputXGradient	<b>2.3056</b>	<b>1.1475</b>	0.8135	0.0014	1.3854	11.5830
IntegratedGradients	2.3594	1.2064	0.8260	<b>0.0013</b>	<b>1.3537</b>	<b>11.3763</b>
Saliency	2.3788	<b>1.0921</b>	0.8174	0.0014	<b>1.3636</b>	11.7546
<b>Perturbation-based</b>						
Dynamask	2.4382	1.2650	0.8271	<b>0.0013</b>	1.3806	11.6034
FeatureAblation	2.3859	1.1513	0.8459	0.0014	1.3869	11.5511
FeaturePermutation	2.4015	1.1654	<b>0.7949</b>	0.0014	1.3991	11.5112
Occlusion	<b>2.3430</b>	1.2078	0.8107	0.0014	1.3752	<b>11.3569</b>
<b>Others</b>						
KernelShap	2.4115	1.1802	0.8288	0.0014	1.3785	11.6568
Lime	2.4259	1.1584	<b>0.8040</b>	0.0014	<b>1.3732</b>	11.6323
ShapleyValueSampling	<b>2.3352</b>	1.1671	0.8153	0.0014	1.3745	<b>11.4625</b>

Table 6: **Sensitivity comparison.** Computed values show the Sensitivity of a sample. Results show larger values for Lime and Shap-based approaches. Overall the performance of the perturbation-based approaches was superior to most of the other approaches.

Method	Anomaly	CharacterTrajectories	ECG5000	FaceDetection	FordA	UWaveGestureLibraryAll
<b>Gradient-based</b>						
GradientShap	0.9364	0.6610	0.9149	0.9764	1.0369	1.0347
GuidedBackprop	0.1324	0.1531	0.0562	0.1339	<b>0.0398</b>	0.2057
InputXGradient	0.1890	0.1017	0.0709	0.0952	0.0924	0.1927
IntegratedGradients	0.1166	0.1144	0.0458	<b>0.0419</b>	0.0906	0.2086
Saliency	0.1902	0.1126	0.1841	0.0995	0.0762	0.2220
<b>Perturbation-based</b>						
Dynamask	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>
FeatureAblation	<b>0.0414</b>	<b>0.0360</b>	<b>0.0350</b>	0.0581	0.0463	<b>0.0444</b>
FeaturePermutation	<b>0.0414</b>	<b>0.0360</b>	<b>0.0350</b>	0.0581	0.0463	<b>0.0444</b>
Occlusion	0.0645	<b>0.0167</b>	<b>0.0305</b>	<b>0.0506</b>	<b>0.0254</b>	<b>0.0352</b>
<b>Others</b>						
KernelShap	1.0908	0.9405	0.2162	0.9248	0.8876	1.0283
Lime	0.8221	0.4986	0.1408	1.5613	0.6974	0.6378
ShapleyValueSampling	0.9132	0.3917	0.1852	0.5938	0.5536	0.3458

sitivity it results in a drawback when ranking the features as shown in the accuracy drop experiment. In addition, perturbation-based approaches have shown 30.9% better results on average concerning their Sensitivity across all datasets. The FordA dataset has shown the most significant difference between the attribution methods (42.1%), while the CharacterTrajectories dataset has shown the lowest (26.1%). Besides, the impressive performance of Dynamask, the Occlusion, FeatureAblation, and FeaturePermutation have shown results underlining their robustness against permutations.

## 5.4 Runtime

The runtime and resource consumption are important aspects. Even though, the availability of resources increases, they are not unlimited. Depending on the throughput of the approach real-time interpretability can be possible. For mobile devices, the computation capacity is limited, and low resource dependencies are beneficial. A Quad-Core Intel Xeon processor, Nvidia

GeForce GTX 1080 Ti, and 64 GB memory were used to compare the methods concerning their computational effort. The attribution and execution time for a single sample of each dataset was computed. In Figure 2 shows that especially the simple gradient-based methods like the Saliency, IntegratedGradients, and InputXGradient show a low computation time. On the other side methods like KernelShap and ShapleyValueSampling have shown increased time consumption. There is always the trade-off between how many samples are processed and the computational costs using SVS and KernelShap. During The analysis, the default values suggested in the corresponding papers of the methods were used. In the case of the FaceDetection dataset, the computational overhead of the FeatureAblation, FeaturePermutation, and Occlusion increased a lot as they strongly depend on the number of features. The FaceDetection dataset needs 41 times longer than the anomaly dataset. Overall the computation time of the FaceDetection dataset is four times longer than the aggregated computation of all others. The characteristics of the FaceDetection dataset favor

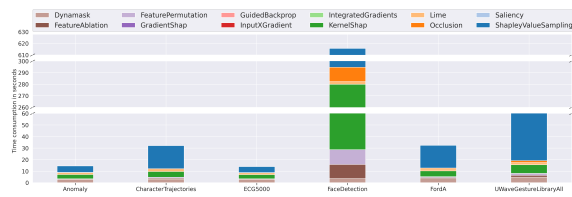


Figure 2: **Time comparison.** Shows the time spend to compute the attribution of a single sample. Note that some bars are not visible due to their fast computation time compared to the other methods and the time of Dynamask is lowered by parameter optimization due to the otherwise unsuitable time consumption. Hardware: Quad-Core Intel Xeon processor, Nvidia GeForce GTX 1080 Ti, and 64 GB memory.

methods that are independent of the number of features. The high number of channels and time-steps when every data feature gets evaluated separately increases up to an unacceptable point. In addition, it has to be mentioned that only 100 epochs instead of the default 1,000 for each optimization of Dynamask were used to lower the computation times. The results show that this does not change the overall results of Dynamask but lowers the computational time by a factor of ten. Using the default 1000 epochs would not be suitable in any case as the computation time would increase by a factor of ten.

### 5.5 Attribution Correlation

Another aspect is the correlation of the different attribution maps. Therefore, different correlation measurements were used, namely the Pearson correlation (Benesty et al., 2009), Spearman correlation (Myers and Sirois, 2004) and Jaccard Similarity (Niwattanakul et al., 2013). The Pearson correlation measures the correlation between two series concerning their values. Spearman correlation is a ranked measurement that compares the ranks for each of the features. Finally, the Jaccard Similarity is used as a set-based measurement. During this experiment, the similarity of the attributions computed over the 100 test sample subsets was evaluated. Ultimately, only the important points matter concerning a correct attribution. That means intuitively, the similarity of the methods concerning irrelevant points. To consider that, percentile subsets of the important features were selected for the Jaccard Similarity to understand the agreement of the methods concerning those features. Summarizing the different similarity and correlation metrics, the absolute correlation using the Pearson correlation, the ranking using the Spearman correlation, and the important set of features using the Jaccard similarity were used.

Figure 3 shows the results. The correlation matrices for the CharacterTrajectories and FordA dataset

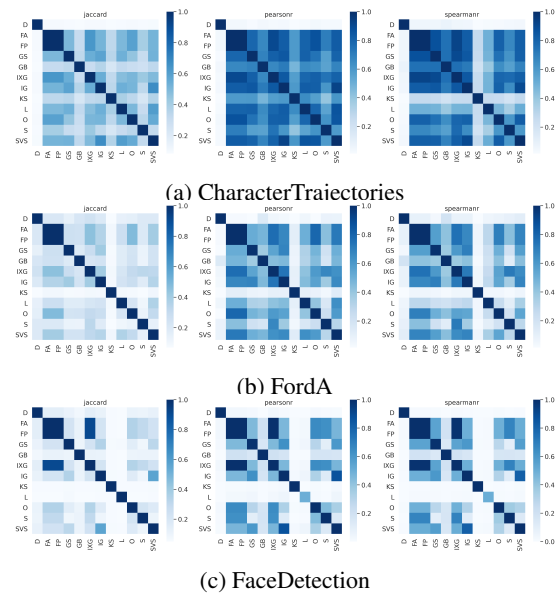


Figure 3: **Attribution correlation** Shows the average correlation/similarity of over 100 attributions. The ten percent most important features were selected for the Jaccard similarity. The method names are shortened using only the capital characters. KernelShap shows a significantly lower correlation to other methods compared to all others. Feature Ablation and FeaturePermutation have shown a high correlation.

as the other datasets have similar show results. Overall every matrix shows the same behavior. Feature-Ablation (FA) and FeaturePermutation (FP) are very similar. In addition, the Dynamask (D) approach and KernelShap (KS) are different from any of the others. This difference is the case for Dynamask, as the technique only makes a binary decision if a feature is significant or not. Intuitively, this should result in a high similarity for the Jaccard measurement. However, this is not the case as the attribution of Dynamask has an internal smoothing based on the loss used to optimize the mask. This smoothing will include less important features in the important feature set to preserve a continuous mask. Furthermore, Lime (L) and KernelShap (KS) seemed less similar to the other approaches.

### 5.6 Dependency on Model Parameter

Attribution methods should depend on the model parameter and the labels of the data. Therefore, the impact of label permutation and parameter randomization of the model was evaluated. The paper only shows the results using the CharacterTrajectories dataset as the results on the other datasets are similar.

The idea of the label permutation is that attribution methods should depend heavily on the labels. Good results in this experiment show a high intrinsic

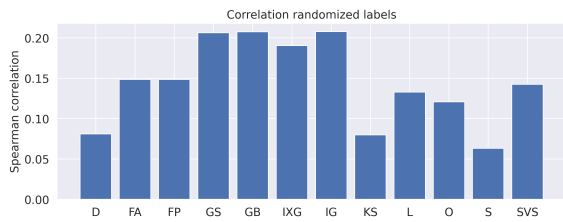


Figure 4: **Attribution comparison.** Shows the Spearman correlation (rank correlation) of the attribution methods evaluated on the same model architecture using randomized training labels using the CharacterTrajectories dataset. The method names are shortened using only the capital characters. Dynamask, KernelShap, and Saliency show a significantly lower dataset dependence.

sic data characteristic dependence which is not a desired feature of an attribution method. The models were trained similar to the baseline model on the same training data but permuted the labels. This permutation results in a model that does not generalize well but learns to replicate the training set. In addition, this approach did not require the validation dataset. The accuracies of those models are very high for the training set. Nevertheless, they fail on the test set. Precisely speaking, these models do not have a label dependence. All models reached a near-perfect performance on the training set. Figure 4 highlights that the correlation drops down to values between 0.05 and 0.2. Based on the overall low correlation, the attribution methods highly depend on the labels rather than dataset characteristics. GradientShap, GuidedBackprop, InputXGradient, and IntegratedGradients have shown three times larger correlations in contrast to Dynamak, KernelShap, and Saliency. However, their correlation is still low enough to justify the label dependency.

In addition to the label permutation, layers of a correctly trained network were systematically randomized to understand the dependency concerning the model parameters. To understand the impact of the layers, each layer was randomized independently. Further, the model was randomized starting from the bottom to the top and vice-versa. The results in Figure 5 show all three approaches. Interestingly, the correlation of GuidedBackprop stays high when randomizing the top layers but significantly drops when randomizing the bottom layers. Randomizing the upper layers, the correlation of Guidedbackprop is close to the original attribution map, whereas the correlation of the other methods drops by 0.5 or more. That suggests that this method is more based on the values of the first few layers. In addition, the results show that for all attribution techniques, a single randomized layer is enough to get an attribution that is no longer related to the original attribution map. This

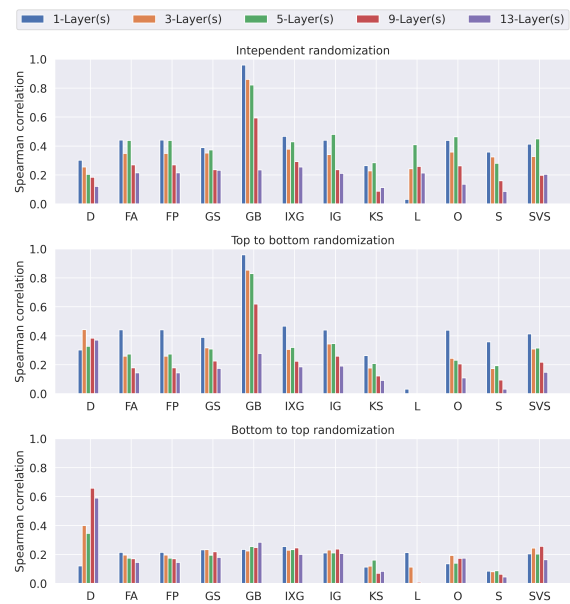


Figure 5: **Correlation to original attribution.** Shows the Spearman correlation of the attribution methods evaluated on the trained model and randomized layer weights using the CharacterTrajectories dataset. Weights are either randomized for each layer independently, from top to the bottom layer or vice versa. Only layers with trainable parameters (conv, batchnorm, dense) are included when counting the number of randomized layers. The method names are shortened using only the capital characters. GuidedBackprop shows significant correlations when only the upper layers are randomized. The correlation of all other methods drops significantly.

high dependency on the model parameter is the desired property. The top to bottom randomization further shows that except for the Dynamask approach, the correlation continuously gets smaller when randomizing more layers. Finally, the bottom to top randomization highlights that the randomization of the first layer of the network is enough to produce attribution maps that are not related to the original.

## 5.7 Visual Attribution Comparison

Figure 6 shows all computed attribution maps for a reference sample. Due to interpretability reasons, an anomalous instance of the anomaly dataset was selected. The example in the top left corner contains a single anomaly in one channel that is important for the classification. The rest of the figure shows the different attribution maps and the impact of randomization on the methods. The figure shows the robustness to randomized parameters. In the second column, the Integrated gradients approach was able to find the peak. This column corresponds to a model trained on randomized labels. Therefore, the model



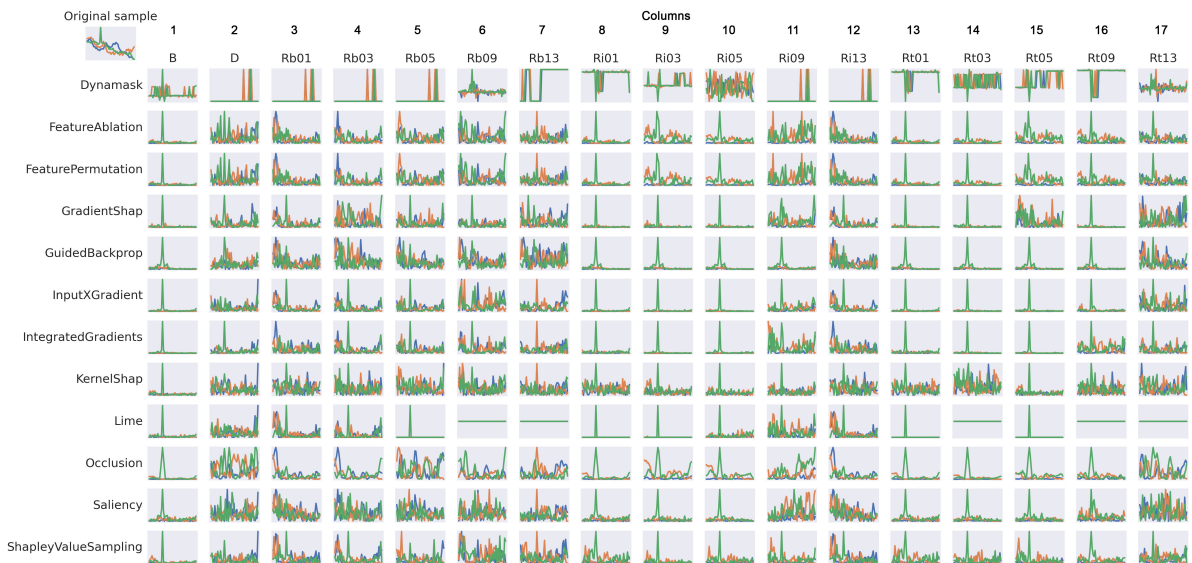


Figure 6: **Visual comparison.** Shows all attributions for a selected anomaly sample. The important part is the peak of the sample. 'Ri', 'Rb', 'Rt', 'D', and 'B' correspond to the independent, bottom to top, top to bottom randomization, label randomization, and original attribution map. Only conv, batchnorm, and dense layers are counted. Changing the data labels during training significantly worsens the performance of all approaches except IntegratedGradients for the anomaly dataset. Overall randomizing lower layers resulted in much more noise compared to randomization in the upper layers.

used in column two is not generalized and learned only to map the training data. Columns three to seven show a model randomization starting from the bottom layers. The results show that some methods still perform well when only one or three layers starting from the bottom are randomized other attribution methods directly collapsed. Columns eight to twelve show the independent layer randomization. Except for Dynamask, the attribution techniques were able to deal with up to handle the layer randomization in the upper layer of the network quite well, whereas all attribution methods collapsed when the lower layers were randomized. Columns thirteen to seventeen show the randomization starting from the top of the network. Most attribution methods were able to recover from the randomization for a high number of randomized layers. Overall the randomization of the lower layers changed the attribution much more concerning the noise. Interestingly, changes in the upper layers did not affect the attribution methods that much.

## 5.8 Continuity

One aspect that is missing most times is attribution continuity. In the image domain, the use of superpixels solves this problem. However, in the time series domain, it is not that easy. Most of the attribution methods do not consider groups of values. In Table 7 shows the evaluation of the continuity. The continuity calculates the absolute difference between the attribu-

tion value of a point  $t$  and  $t + 1$  for each time-step and each channel. Using the mean across a sample provides a value that indicates how continuous the explanation is. Lower values correspond to an explanation that does not contain many switches from important to not important features. This measurement was computed over the 100 attributed samples and took the mean for each dataset. The results indicate that the perturbation-based approaches favor continuous explanations. Gradient-based methods overall have shown the worst performance. One reason for this is the noisy gradients used to compute the attribution maps.

## 6 DISCUSSION

A summarization and discussion in a detailed manner is offered to provide on choosing an attribution method. The different aspects and application scenarios are described below. First, it has to be mentioned that every attribution method has shown satisfying results. However, the choice of an attribution method should depend on the required characteristics. The overall results are presented in Table 8. The results highlight that choosing an attribution method can be very important, as mentioned by Vermeire et al. (Vermeire et al., 2021).

Starting with the accuracy drop, the evaluation shows to which extend the methods rank the most

Table 7: **Continuity comparison.** Computed values show the mean continuity of the attribution maps. Lower values correspond to continuous maps. Continuity was calculated by shifting the attribution map, subtracting it from the original one, taking the absolute values, and computing the mean. Lower values are better. Perturbation-based methods have been shown to outperform gradient-based with respect to the continuity on almost all datasets. Specifically, Dynamask and Occlusion have been shown to perform well across all datasets.

Method	Anomaly	CharacterTrajectories	ECG5000	FaceDetection	FordA	UWaveGestureLibraryAll
<b>Gradient-based</b>						
GradientShap	0.0947	0.0368	0.0616	0.0613	0.0813	0.0543
GuidedBackprop	0.1201	0.0537	0.0913	0.0957	<b>0.0801</b>	<b>0.0526</b>
InputXGradient	0.0801	0.0390	0.0508	0.0620	0.0855	0.0537
IntegratedGradients	0.0864	0.0369	0.0609	0.0632	0.0858	0.0508
Saliency	0.1176	0.0748	0.1439	0.1170	0.1229	0.0842
<b>Perturbation-based</b>						
Dynamask	<b>0.0282</b>	<b>0.0014</b>	<b>0.0252</b>	<b>0.0107</b>	<b>0.0159</b>	<b>0.0015</b>
FeatureAblation	0.0784	0.0395	0.0584	0.0624	0.0815	0.0601
FeaturePermutation	0.0784	0.0395	0.0584	0.0624	0.0815	0.0601
Occlusion	<b>0.0623</b>	<b>0.0183</b>	<b>0.0419</b>	<b>0.0367</b>	<b>0.0535</b>	<b>0.0284</b>
<b>Others</b>						
KernelShap	0.1423	0.1086	0.0641	0.1671	0.1973	0.1795
Lime	0.1122	0.0496	<b>0.0498</b>	<b>0.0010</b>	0.0883	0.0928
ShapleyValueSampling	<b>0.0773</b>	<b>0.0365</b>	0.0505	0.0583	0.0885	0.0713

Table 8: **Overall Evaluation.** Overall results with respect to the different aspects evaluated in this paper. A = Accuracy Impact / Agreement, I = Infidelity, S = Sensitivity, R = Runtime, Ld = Label dependency, Md = Model Parameter Dependency, C = Continuity.

Method	A	I	S	R	Ld	Md	C
<b>Gradient-based</b>							
GradientShap		⊕		⊕			
GuidedBackprop	⊕			⊕		⊖	
InputXGradient		⊕		⊕			
IntegratedGradients		⊕		⊕			
Saliency	⊖			⊕	⊕		
<b>Perturbation-based</b>							
Dynamask	⊖		⊕	⊖	⊕		⊕
FeatureAblation	⊕		⊕	⊖			
FeaturePermutation	⊕		⊕	⊖			
Occlusion		⊕	⊕	⊖			⊕
<b>Others</b>							
KernelShap	⊖			⊖	⊕		
Lime		⊕		⊕		⊕	⊕
ShapleyValueSampling	⊕			⊖			⊕

and least significant features based on the impact on the accuracy. Most of the methods were able to show high-quality results across all datasets. However, there were some outstanding performances. Specifically, the perturbation-based were able to perform slightly better than the other methods on some datasets. Saliency and Dynamask have shown some weaknesses for some datasets, such as the CharacterTrajectories and FordA. Both methods require further adjustments and knowledge about the data to achieve good results. One example is the ratio of significant points for the Dynamask approach to select the correct number of features. If additional information is available, such as the ratio of selected features, methods like Dynamask can express their full potential. The attribution agreement shows similar results.

Concerning Infidelity and Sensitivity, every method performed well, and no approach suffered more. The results show that gradient-based methods obtained the best Infidelity results. It was the opposite for the Sensitivity. Especially, GradientShap, InputXGradient, and Saliency approaches are robust against significant perturbations in the input space (Infidelity). On the other side, the Dynamask, FeaturePermutation, and Occlusion approaches have shown good robustness concerning changes in the attribution when small perturbations to the input are applied (Sensitivity). Dynamask has a loss that forces a binary decision whether a feature is selected or not ensures this behavior. Using attribution methods with low Sensitivity values in cases where adversarial attacks can occur is suggested.

The runtime aspect gets critical when the use case requires near real-time explanations. In addition, the results have shown that the dataset characteristics are relevant. The findings show that approaches based on the sequence length and number of channels suffer from very high runtimes for single samples. These runtimes make it impossible to use them in a real-time scenario. However, if the time consumption is not of interest, this aspect is not relevant. Furthermore, gradient-based methods are less dependent on the dataset characteristics and very suitable when time matters. Contrarily, besides Dynamask and Lime, the perturbation-based approaches suffer from the number of features. In the case of Lime, the number of samples required to populate the space to train the surrogate model increases with a higher number of features. Dynamask does not suffer from the feature number. However, the approach needs an additional training phase. This training requires multiple epochs

and in addition repetitions based on the different areas checked during the training. Ultimately, the back-propagation needs resources and time. Based on the computational times, the use of ShapelyValueSampling and KernelShap in real-time scenarios is nearly impossible. For completeness, it has to be mentioned that it is possible to tweak hyperparameters.

The label permutation and layer randomization provided insights concerning the role of the model parameters during the attribution computation. Intuitively, all methods have shown a high dependency on the labels of the data. Training a model with randomized targets has shown, the attributions depend on the labels as they should. Although all methods have shown this dependency, the Saliency, Dynamask, KernelShap, and Lime have shown more dependence on the targets. Concerning the model parameters, the results show that randomizing any layer results in changes of the attribution maps. Besides, the Guided-Backprop attribution maps significantly change after any modification. Specifically, Lime collapses completely. This collapse emphasizes that Lime directly depends on the model, and GuidedBackprop is relying more on data. An explanation for this behavior is that some methods detect dataset differences. Especially in the image domain, it was shown that some attribution methods can act like an edge detectors.

Finally, continuity plays a pivotal role in human understanding. In use cases that include human evaluation, it is beneficial to have continuous attribution maps. Imagine there is a significant frame with many important but some less important features. It might be superior to mark the whole window as important, although this covers some insignificant features. In the time series domain, the context matters, and continuous attribution maps are easier to understand. The results show that the Dynamask approach, Lime, Occlusion, and ShapleyValueSampling are superior concerning their continuity. Intuitively, the attribution maps produced by gradient-based techniques look noisy, whereas permutation-based look smoother. Dynamask includes a loss term that ensures a smoother attribution map. Lime and ShapleyValueSampling produce smoother maps. The results suggest using a perturbation-based approach if a human inspection is relevant.

Comparing the gradient-based, perturbation-based, and other approaches, every category has shown advantages over the other category in some aspects. Generally, gradient-based methods are fast, show high Infidelity, label dependency but are noisy, not continuous, and suffer concerning the Sensitivity. In contrast to gradient-based methods, perturbation-based approaches produce continuous maps, shine

concerning the Sensitivity, label dependency but suffer when it comes to the runtime.

## 7 CONCLUSION

A comprehensive evaluation of a large set of state-of-the-art attribution methods applicable to time series was performed. The results show that most attribution methods can identify significant features without prior knowledge about the data. In the evaluation, the perturbation-based approaches have shown slightly superior performance in the data occlusion game. In addition, the results are validated by measuring the agreement of the methods using different correlation and similarity measurements. Except for Dynamask and KernelShap, the correlation between the attribution methods showed high values. Further experiments were conducted to highlight the high dependence of the attribution methods on the model and the target labels. Only Guided-Backprop has shown lower reliance on the top layers of the network. Concerning Infidelity, the gradient-based attribution methods showed superior performance. The perturbation-based attribution methods are superb concerning Sensitivity and continuity. Continuity is an important aspect when it comes to human interpretability. The results hold across a set of different tasks, sequence lengths, feature channels, and the number of samples. Furthermore, the results show that the choice of an attribution method depends on the target scenario, and different aspects like runtime, accuracy, continuity, noise are indispensable.

## ACKNOWLEDGMENT

This work was supported by the BMBF projects SenSAI (BMBF Grant 01IW20007) and the ExplAINN (BMBF Grant 01IS19074). We thank all members of the Deep Learning Competence Center at the DFKI for their comments and support.

## REFERENCES

- Abdul, A., von der Weth, C., Kankanhalli, M., and Lim, B. Y. (2020). Cogam: Measuring and moderating cognitive load in machine learning model explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14.
- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*.

- Allam, Z. and Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. *Cities*, 89:80–91.
- Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. (2017). Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*.
- Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. (2019). Gradient-based attribution methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 169–191. Springer.
- Bagnall, A., Lines, J., Vickers, W., and Keogh, E. (2021). The uea & ucr time series classification repository.
- Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.
- Bibal, A., Lognoul, M., de Stree, A., and Fréney, B. (2020). Impact of legal requirements on explainability in machine learning. *arXiv preprint arXiv:2007.05479*.
- Crabbé, J. and van der Schaar, M. (2021). Explaining time series predictions with dynamic masks. In *Proceedings of the 38-th International Conference on Machine Learning (ICML 2021)*. PMLR.
- Das, A. and Rad, P. (2020). Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*.
- Došilović, F. K., Brčić, M., and Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE.
- Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81.
- Huber, T., Limmer, B., and André, E. (2021). Benchmarking perturbation-based saliency maps for explaining deep reinforcement learning agents. *arXiv preprint arXiv:2101.07312*.
- Ivanovs, M., Kadikis, R., and Ozols, K. (2021). Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*.
- Karliuk, M. (2018). Ethical and legal issues in artificial intelligence. *International and Social Impacts of Artificial Intelligence Technologies, Working Paper*, (44).
- Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.
- Mitchell, R., Cooper, J., Frank, E., and Holmes, G. (2021). Sampling permutations for shapley value estimation. *arXiv preprint arXiv:2104.12199*.
- Myers, L. and Sirois, M. J. (2004). Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences*, 12.
- Nielsen, I. E., Rasool, G., Dera, D., Bouaynaya, N., and Ramachandran, R. P. (2021). Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:2107.11400*.
- Niwattanakul, S., Singthongchai, J., Naenudorn, E., and Wanapu, S. (2013). Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.
- Perc, M., Ozer, M., and Hojnik, J. (2019). Social and juristic challenges of artificial intelligence. *Palgrave Communications*, 5(1):1–7.
- Peres, R. S., Jia, X., Lee, J., Sun, K., Colombo, A. W., and Barata, J. (2020). Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook. *IEEE Access*, 8:220121–220139.
- 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, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.
- Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. (2016). Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*.
- Siddiqui, S. A., Mercier, D., Munir, M., Dengel, A., and Ahmed, S. (2019). Tsviz: Demystification of deep learning models for time-series analysis. *IEEE Access*, 7:67027–67040.
- Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.
- Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.
- Vermeire, T., Laugel, T., Renard, X., Martens, D., and Detryniecki, M. (2021). How to choose an explainability method? towards a methodical implementation of xai in practice. *arXiv preprint arXiv:2107.04427*.
- Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D. I., and Ravikumar, P. K. (2019). On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32:10967–10978.
- Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.
- Zhang, Q. and Zhu, S.-C. (2018). Visual interpretability for deep learning: a survey. *arXiv preprint arXiv:1802.00614*.