Ranking by Aggregating Referees: Evaluating the Informativeness of Explanation Methods for Time Series Classification

Surabhi Agarwal, Trang Thu Nguyen, Thach Le Nguyen, and Georgiana Ifrim

School of Computer Science, University College Dublin, Ireland
{surabhi.agarwal,thu.nguyen}@ucdconnect.ie
{thach.lenguyen,georgiana.ifrim}@ucd.ie

Abstract. In this work, we focus on quantitatively evaluating and ranking explanation methods for time series classification based on their informativeness. Time series classification has many applications and evaluating which parts of the time series are most informative for a classifier decision is important. For example, to decide between Arabica and Robusta coffee leaves, we can use an explanation method to highlight the time series parts which differentiate these leaves. Although many explanation methods have been proposed for images and time series data, it is still unclear how to objectively evaluate them. Here, we evaluate two model-specific explanation approaches - ResNet-CAM and MrSEQL-SM, and two model-agnostic approaches, LIME combined with classifiers MrSEQL and ROCKET. We generate saliency-based explanations for each classifier on three time series classification datasets from the UCR benchmark. Importance weights for all points in the timeseries are extracted based on each explanation method, in order to perturb specific parts of the time series and assess the impact on the classification accuracy of referee classifiers. We propose a new ranking-based methodology to compare multiple explanation methods on the basis of their informativeness, by using explanation-based perturbation and aggregating the explanation rank over the referee classifiers. This enables us to compare explanation methods within a single dataset and also across multiple datasets. We provide an in-depth analysis of the results attained, also including runtime analysis for each method. Our results indicate that model-specific approaches MrSEQL-SM and ResNet-CAM are much faster than model-agnostic approaches MrSEQL-LIME and ROCKET-LIME and that MrSEQL-SM yields the highest informativeness rank among the explanation methods compared.

Keywords: Time Series Classification · Explanation Methods

1 Introduction

In recent years Machine Learning (ML) systems have become highly impactful in our everyday life. These methods are growing in terms of their complexity, performance as well as their impact. With the rise in the complexity of ML models, it is also becoming more important to understand their decision-making process which is connected to their *interpretability* [19]. Interpretability is the degree to which a human can understand the cause of a decision [10]. The higher the interpretability of a machine learning model, the easier it is for someone to understand why certain decisions or predictions are made. Understanding the reasons behind these predictions is also important in assessing trust if actions are to be made based on the predictions of the model. Such an understanding gives insights into the model, which can be further used to transform an unstable or inaccurate model or prediction into a stable and trustworthy model [19]. If one can ensure that the ML model can explain decisions and have high interpretability, then the models can be evaluated using some traits such as fairness, privacy, reliability, causality, and trust [7]. The existing approaches can be categorized as techniques that are intrinsic or post-hoc and whether they are global or local [8,16]. A time series is an ordered sequence of numeric values and time series classification (TSC) helps us with predicting a class label for time series. Explainable AI and evaluating the interpretability of TSC methods, help the user understand exactly which part of the time series data resulted in the prediction. This explanation can be visualized as a saliency map by highlighting the parts of the time series which are informative for the classification decision. There are several empirical surveys in recent TSC literature [2,3] and methods which help in designing intrinsic as well as post-hoc explainable models [1,18,19]. However, there is still a strong need to *objectively evaluate and compare* such methods and attain useful explanations. In this work, we evaluate recent explanation methods and propose strategies to provide a *quantitative evaluation using informativeness*. Figure 1 shows the saliency maps produced by four explanation methods: MrSEQL-SM, ResNet-CAM, MrSEQL-LIME and ROCKET-LIME. We can see that the four explanation methods do not agree on which are the important parts of the time series. We aim to evaluate explanation methods based on their informativeness through an explanation-driven perturbation. We focus on methods that produce explanations in the form of saliency maps. In our experiments, we consider two model-specific explanation methods - ResNet-CAM [26] and MrSEQL-SM [11], and two model-independent methods - LIME [19] combined with MrSEQL and ROCKET [4]. The main contributions of this work include:

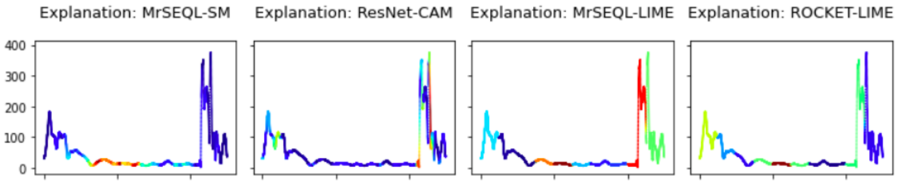

Fig. 1: Saliency map explanations for a motion time series from the dataset CMJ. The most informative parts are highlighted in deep red and the non-informative parts in deep blue.

- A review of the state-of-the-art approaches for explanation of TSC including model-specific explanation methods such as ResNet-CAM and MrSEQL-SM and model-agnostic explanation methods such as LIME and Shapley.
- A new ranking-based methodology to compare multiple explanation methods on the basis of their informativeness, by using explanation-based perturbation and aggregating the explanation rank over a set of referee classifiers.
- Generation of explanations using LIME for the recent efficient time series classifier ROCKET.
- An empirical analysis of the runtime and an in-depth quantitative evaluation and discussion of the results of four TSC explanation methods ranked over three UCR datasets.

2 Related Work

We first discuss the recent literature on TSC algorithms followed by explanation methods for TSC as well as some of the approaches used to evaluate these explanations.

2.1 Time Series Classification

Time series are commonly used for representing data such as stock prices, weather readings, and biological observations. Time Series Classification (TSC) is a technique used to predict class labels for a given time series [12] and has many applications. In the survey [3] TSC methods have been categorized into five categories including distance-based, interval-based, dictionary-based, ensemble-based, and Deep Learning (DL) based classifiers. The traditional **distance-based classification** technique uses distance measures to determine the class membership. The 1-Nearest-Neighbour algorithm is used as a baseline classifier to classify univariate time series using Euclidean distance and Dynamic Time Warping (DTW) as well as multivariate time series using Frobenius distance [20]. **Interval-based classifiers** select one or more intervals of the series to generate results. An example of interval-based classifiers includes Time Series Forest Classifier (TSF) which adapts the random forest classifier to series data [5]. **Dictionary-based classifiers** form counts of string patterns and then build classifiers based on the resulting features [3]. With the introduction of Bag of SFA symbols (BOSS) [21], Word Extraction for Time Series Classification (WEASEL) [23], SAX-VSM [11] and MrSEQL [11], dictionary-based classifiers have seen major advancements.

Other important classes of TSC algorithms are **DL-based classifiers** and **Ensemble-based classifiers**. DL-based approaches include the use of Multi-Layer Perceptron (MLP), Fully Convolutional Neural Network (FCN), Residual Network (ResNet), Encoder, Multi-scale Convolutional Network (MCNN), Time Le-Net (t-LeNet) and a few others [9, 11]. Ensembled-based approaches include Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [2] which has high accuracy but a heavy computational cost. HIVE-COTE predictions are a weighted average of predictions produced by classifiers such as Shapelet Transform Classifier, BOSS, Time Series Forest, and RISE.

2.2 Explanation Methods for Time Series Classification

The goal of an explanation is to relate the feature values of an instance to its model prediction in a way that is understandable to humans [16]. One such tool to represent these explanations is a saliency map.

Saliency Maps. A saliency map is a heatmap that highlights parts of an input that most influenced the output classification [17]. Saliency maps can be used in TSC to highlight the parts of the time series that are important. They are often generated by matching a time series with a vector of weights (w) using a colour map. This vector of weights contains a corresponding weight value for each data point in the time series. The process of generating saliency maps in TSC and producing the vector of weights for the mapping is called the TSC explanation method, and the saliency map produced is known as the TSC explanation [17]. Figure 2 shows a visual representation of how a shape can be converted into a time series using an example of a *Verbena urticifolia* leaf as shown in [25]. The authors of [11] use this representation to classify the Coffee dataset and to produce explanations for the classifier decision as shown in Figure 3. The highlighted regions of the image correspond to the caffeine and chlorogenic acid components of the coffee blends Arabica and Robusta. An explanation approach has three important aspects as highlighted in [16]:

- Intrinsic or post-hoc: Intrinsic models are those which are considered interpretable due to their simplicity, such as linear models or decision trees. Post-hoc models are black-boxes and special methods need to be developed to obtain explanations.
- Model-specific or model-agnostic: Model-specific approaches are specific to a single model or a group of models. These rely on the working capabilities of the particular model to provide explanations. On the other hand, model-agnostic approaches can be utilized for any ML model regardless of the complexity of the model.
- Local or global scope: The scope of the model can be either local or global depending on whether the method explains an individual prediction or the entire model.

Recent work [17] has shown some contribution towards a quantitative approach for evaluating explanation methods for TSC, such as CAM, MrSEQL-SM and LIME. That methodology proposed an explanation-based perturbation to compute informativeness, but did not provide a way to directly compare and rank explanation methods within and across datasets. In this work, we focus on two model-specific approaches - ResNet-CAM and MrSEQL-SM and two model-agnostic approaches - MrSEQL-LIME and ROCKET-LIME - in order to quantitatively evaluate and rank these methods based on their informativeness.

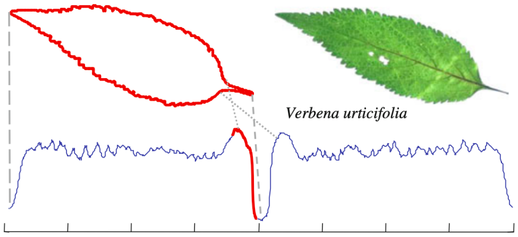

Fig. 2: An example of how a shape can be converted into a *time series* representation (reprinted from [25]).

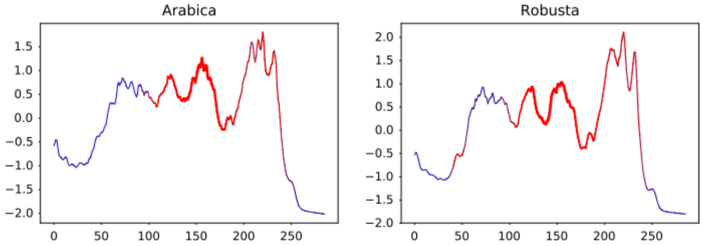

Fig. 3: Saliency mapping generated using the MrSEQL classifier proposed in [11] on the Coffee dataset (reprinted from [11]).

2.3 Model-Specific Approaches

ResNet-CAM. Class Activation Map (CAM) is a model-specific explanation method that helps in explaining the output predictions of a neural network. In previous work [26], CAM is implemented for image classification to visualize the predicted class scores and highlight the discriminative image features used by the CNN to classify the image. The implementation of CAM relies on performing Global Average Pooling (GAP) just before the final output layer. Using the above technique and the network architecture, the weights from the GAP layer can be used to highlight the important parts of the time series which led to the prediction. The obtained weights can then be used to visualize the explanation using the saliency mapping of the weight vector to the original time series.

MrSEQL-SM. Multi-resolution Symbolic Sequence Learner (MrSEQL) [6, 11] classifier is an efficient TSC algorithm that trains a linear classification model. The algorithm transforms numeric time-series data into multiple symbolic representations of different domains such as SAX [13] in the time domain and SFA [22] in the frequency domain. The classifier selects the most important subsequences from the symbolic data which are then used as input features for training the SEQL classifier [11]. SEQL trains using logistic regression and outputs a linear model which is a set of weighted symbolic subsequences. For the SAX features which are in the time domain, saliency maps are then produced when these features and weights are mapped back to the original time series. This explanation produced in the form of a saliency map for MrSEQL with SAX features is called MrSEQL-SM [17].

2.4 Model-agnostic Approaches

LIME. Local Interpretable Model-agnostic Explanations (LIME) [19] is a model-agnostic technique that explains the predictions of any classifier by approximating it locally with an interpretable model. In [19] the authors propose an implementation of LIME focused on training interpretable or local surrogate models to explain individual predictions. LIME examines how variations to the data fed into a black-box model, impact the model predictions. To achieve this, LIME perturbs the data and obtains black-box predictions for the new data points. Then, LIME trains an interpretable model on this perturbed dataset. The new samples are weighted according to their proximity to the instance of interest for which the explanation needs to be generated. This way LIME obtains the explanations for the instances locally and does not give a global approximation. LIME was previously implemented with text, image and tabular data [19]. For tabular data, variations of the data were produced by perturbing each feature individually. In the case of images, the variations are created by segmenting the image into *superpixels* which can be turned on or off with a user-defined colour. LIME can also be adapted for time series data as shown in [15, 17]. Some of the key advantages of LIME are that it makes human-friendly and easily interpretable explanations and has local fidelity in terms of giving insight into explaining the black-box predictions locally [16]. LIME also has drawbacks, e.g., it samples data points using a Gaussian distribution which ignores feature correlation. There is also instability in the explanations produced, i.e., the explanations vary depending on some hyperparameters. An alternative to LIME is the Shapley value-based SHAP [14]. Even though SHAP gives benefits of local and global interpretability, it requires a lot of computation time since it is computing all possible feature permutations globally. Hence, LIME would have an advantage of speed when compared to SHAP. There is also no open implementation of SHAP for time series, hence we use LIME in this work.

ROCKET. RandOM Convolutional KErnal Transform (ROCKET) [4] is a classification method that transforms time series using random convolutional kernels (shape features) and trains a linear classifier using those transformed features. ROCKET can attain state-of-the-art accuracy using a fraction of the time as compared to other algorithms, including CNN. Since ROCKET uses a combination of shape features and numeric features - the proportion of positive values (ppv), it becomes difficult to obtain a saliency map directly from the linear model and we thus use LIME to obtain a post-hoc explanation for ROCKET, called ROCKET-LIME.

2.5 Evaluation Measures for Explanation Methods

According to [7], there are three main levels for the evaluation of interpretability - application grounded, human grounded, and function grounded. These vary in terms of complexity and the need according to different tasks. TSC explanation is aimed at focusing on the **discriminative** parts of the time series i.e., the parts important for classification. In TSC explanation, we want to evaluate explanations for individual predictions on the function level. There are several measures that can be used to judge how good an explanation method or explanation is [16]. Explanation methods have measures such as - *expressive power* in terms of the structure of the explanation generated by the model, *translucency* describing how much of the explanation method relies on looking into model parameters, *portability* describing the range of ML models that can implement this explanation method and the *algorithmic complexity* of the algorithm. Individual methods also possess an array of measures such as *accuracy* (how well the explanation reacts to unseen data), *fidelity* (how effectively the method estimates the prediction of black-box models), *consistency* (does the explanation vary between similar models or does it stay the same), *stability* (is a similar explanation generated on each iteration), *comprehensibility* (how well do humans understand the explanations), *certainty* (i.e confidence of the model prediction), *degree of importance* (w.r.t the importance of features or parts of the explanation), *novelty* (is the explanation coming from a new distribution of the training data), and *coverage* in terms of the area covered.

Recent work [17] has used *informativeness* as an evaluation measure and the authors entail that if the explanation is truly informative, it should point out those parts of the time series that are most relevant for the classification decision. The authors highlight the discriminative parts of the time series by identifying a threshold k to find the parts where the weight vector belongs to the (100 - k) percentile discriminative weights. The authors have also made use of perturbation to provide evaluation for both single explanation methods as well as multiple explanation methods. In this work, we propose a novel methodology to calculate and compare informativeness. This extends the work of [17] and is a ranking-based methodology that uses perturbation to compute the ranks of multiple explanation methods over different referee classifiers and datasets. We choose informativeness over other evaluation measures because it helps in quantifying the evaluation for a single explanation and also gives an objective measure to perform a comparison of multiple explanation methods.

3 Proposed Methods

Here we discuss the technique used to perform the perturbation of the test set in order to evaluate the informativeness of a TSC explanation method. The perturbation process is then used for comparing different explanation methods based on their informativeness and for our ranking approach.

3.1 Explanation-based Perturbation of Time Series

The main aim of a TSC explanation method is to emphasize those important regions of the time series that were most impactful for the classification decision. Hence, if an explanation is **informative**, it should point out those discriminative parts. In order to evaluate this, the discriminative regions of the time series test sets are perturbed to examine if a decrease in the classification accuracy is observed. The more informative the explanation is, the higher the expectation of a decrease in accuracy after perturbation based on this explanation method [17]. Here, we work with explanation methods that produce a saliency map for the time series. This information is stored as an array of positive weights w_t, one weight for each step in the time series having t steps. The discriminative weights are ranked through setting a threshold k ($0 \leq k \leq 100$) that is set at the (100 - k) percentile of the positive weight vector (w) that explains the time series [17]. Through this threshold, we can emphasize on the weights having the highest magnitude in the time series. For example, for $k = 10$, the focus with be on the top 10% of the highest weights coming from the explanation method. The time series is perturbed by adding Gaussian noise to its original signal. For a given time series represented by a vector x, the resulting perturbed vector is represented by $x_{perturbed}$ where the entire time series is perturbed and the distribution for the Gaussian noise is $N(\mu, \sigma^2)$, where μ is the mean of the distribution and σ is the magnitude of the noise.

$$x_{perturbed} = x + N(\mu, \sigma^2) \tag{1}$$

In this work, only a region is perturbed by adding noise based on the corresponding weights in the explanation vector. The rest of the time series remains unchanged. For the perturbation parameters we use $\mu = 0$ and $\sigma = 0.2 * range$. This effectively adds or subtracts about 20% of the magnitude range of values in that time series.

3.2 Calculating Informativeness as an Evaluation Metric

In order to quantitatively evaluate the informativeness of an explanation method, an experiment is proposed. Firstly, a time series classifier is trained using the original, non-perturbed training datasets as shown in Figure 4. This classifier acts as an evaluation classifier or *referee classifier*. Thereafter, perturbed test datasets are created by adding noise to the discriminative parts of the time series. Multiple versions of the perturbed test datasets are obtained for multiple explanation methods, at the same threshold k ($0 \leq k \leq 100$). Each of these perturbed test datasets corresponds to an explanation or weight profile obtained from an explanation method.

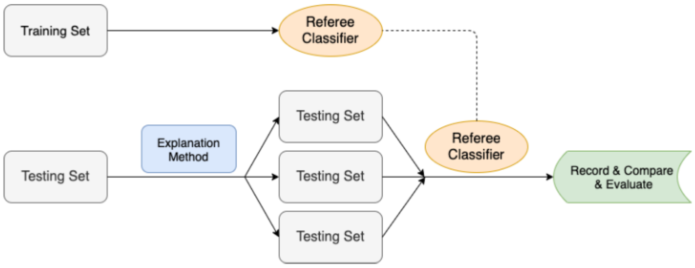

Fig. 4: Method of generating explanation-driven perturbed test sets and evaluating the explanation method through a referee classifier (reprinted from [17]).

If an explanation method is truly informative, the perturbation should impact the referee classifier more strongly than the other explanation methods. The informativeness of an explanation method is calculated by estimating the area under the explanation curve (AUC) described by accuracy at different perturbation levels k with the help of the trapezoidal rule. This metric is coined as an *explanation loss* or *eLoss* in the work [17] since a reduction of accuracy is observed after adding noise to the time series based on the given explanation method.

$$eLoss = \frac{1}{2}k\sum_{i=1}^{t}(acc_{i-1} + acc_i) \tag{2}$$

Here, k represents the values of each step normalized in the range 0-1 where k = 0 corresponds to the original test dataset and the step k = 100 corresponds to perturbing the entire time series, t represents the number of steps in the time series (t=$\frac{100}{k}$) and acc_i represents the accuracy at step i. Here, we call the *eLoss* the **explanation AUC**, this is a numeric measure that varies between 0 and 1. The explanation methods are then compared using an *independent referee* classifier. In this work, we use three state of the art classifiers, MrSEQL, ROCKET and WEASEL and propose a new methodology to rank and compare explanations methods by aggregating over referees. The explanation methods are *ranked* based on their **explanation AUC** for each referee classifier. The lower the AUC, the higher the rank. Once the rank is calculated for an explanation method for one particular referee classifier, the overall rank is calculated by taking the average of all the obtained ranks across referees. The explanation method that ranks the highest is considered to be the most informative explanation method over the set of referees for that dataset. We provide more details on this strategy in the next section.

4 Experiments

Next, we discuss the steps required to generate the informativeness of each explanation method. We use the popular library sktime [24] and extend the open source code of [17]. For each of the explanation methods, i.e., MrSEQL-SM, ResNet-CAM, MrSEQL-LIME and ROCKET-LIME, the following steps are followed in order to evaluate them with respect to informativeness:

1. For each dataset, a referee classifier is trained and the weights are extracted.
2. Each test time series is perturbed with Gaussian noise at different noise levels k (i.e., 0, 10, 20, ..., 100).
3. The explanation AUC is calculated for each of the explanation methods with each referee classifier.
4. The weights are mapped back to the original time series to generate the saliency map for each method for each of the three datasets. The time taken to run and generate results for each explanation method is also recorded by using the `timeit` library.
5. The methods are then evaluated and ranked based on their informativeness using our proposed ranking-based methodology.

4.1 Perturbing and Measuring Metrics

An explanation method should point to discriminative parts of the time series if it is truly informative. If these discriminative parts are perturbed then a decrease in classification accuracy should be observed. Once the test datasets are perturbed, the new accuracy scores are generated and the explanation AUC is computed for each of the explanation methods with each referee classifier to computationally evaluate the usefulness of these explanation methods. Table 1 shows the accuracy at different noise levels k when using MrSEQL as a referee classifier, on the ROCKET-LIME explanation method, over the CMJ dataset. We note that the accuracy decreases as the noise levels increase from 10 to 100. As can also be seen in Figure 5, this behaviour varies depending on the robustness to noise of the referee classifier. Table 2 shows the explanation AUC and the referee rank when using ROCKET as a referee classifier on the four explanation methods over the CMJ dataset.

Table 1: Accuracy for explanation ROCKET-LIME using MrSEQL as a referee classifier after adding Gaussian noise at levels k from 10-100 on the CMJ dataset.

Noise Level	10	20	30	40	50	60	70	80	90	100
Accuracy	0.9609	0.9553	0.9553	0.9497	0.9497	0.9497	0.9441	0.9385	0.9385	0.9385

4.2 Experimental Results and Evaluation

The four explanation methods are evaluated on the basis of their informativeness based on their ranking across the referee classifiers, over the datasets CMJ, Coffee and GunPoint. Due to the computational cost of LIME, MrSEQL-LIME is evaluated with only CMJ and GunPoint datasets whereas ROCKET-LIME is evaluated with the CMJ dataset only.

Table 2: Explanation AUC and rank for the explanation methods using ROCKET as a referee classifier over the CMJ dataset.

Dataset	Weights	Explanation AUC for referee ROCKET	Rank
CMJ	MrSEQL-SM	0.8874	2
CMJ	ResNet-CAM	0.9126	4
CMJ	MrSEQL-LIME	0.9115	3
CMJ	ROCKET-LIME	0.8866	1

Accuracy. Figure 5 shows the accuracy curve for the CMJ dataset after Gaussian noise is added to the time series. This is shown for all the four explanation methods and the three referee classifiers. It can be seen that as the noise levels increase from zero to a hundred, a dip in referee accuracy is seen for all the explanation methods. This supports the fact that performing perturbation decreases the accuracy of the referees.

In order to compare the explanation methods against each other based on the accuracy curve, the accuracy curves are aggregated to see which method is the most informative. The lower curve indicates that performing perturbation decreases the accuracy of the explanation method more. This indicates that the explanation method is more informative. Figure 6 shows the comparison of the accuracy curves for all four explanation methods on the CMJ dataset. It can be seen that although there is an overlap between the curves, MrSEQL-SM shown by the red curve is slightly more informative as compared to the other methods.

Explanation AUC. Table 3 represents the explanation AUC obtained for each of the datasets and the explanation methods across the referee classifiers. We observe that the explanation AUC varies across the three classifiers. The lower explanation AUC value indicates a higher referee rank contributing towards higher informativeness.

Table 3: Explanation AUC obtained for the four explanation methods and referee classifiers. In bold is the lowest AUC over explanations, for a given referee, which results in rank 1 for that explanation method and referee.

Dataset	Explanation Method	MrSEQL	ROCKET	WEASEL
CMJ	MrSEQL-SM	**0.9441**	0.8874	**0.6575**
	ResNet-CAM	0.9453	0.9126	0.6793
	MrSEQL-LIME	**0.9441**	0.9115	0.6933
	ROCKET-LIME	0.9492	**0.8866**	0.7039
Coffee	MrSEQL-SM	**0.9625**	**1.000**	0.9804
	ResNet-CAM	0.9696	**1.000**	**0.9696**
GunPoint	MrSEQL-SM	**0.9477**	**0.7137**	0.5440
	ResNet-CAM	0.9610	0.7350	**0.5280**
	MrSEQL-LIME	0.9677	0.7637	0.5727

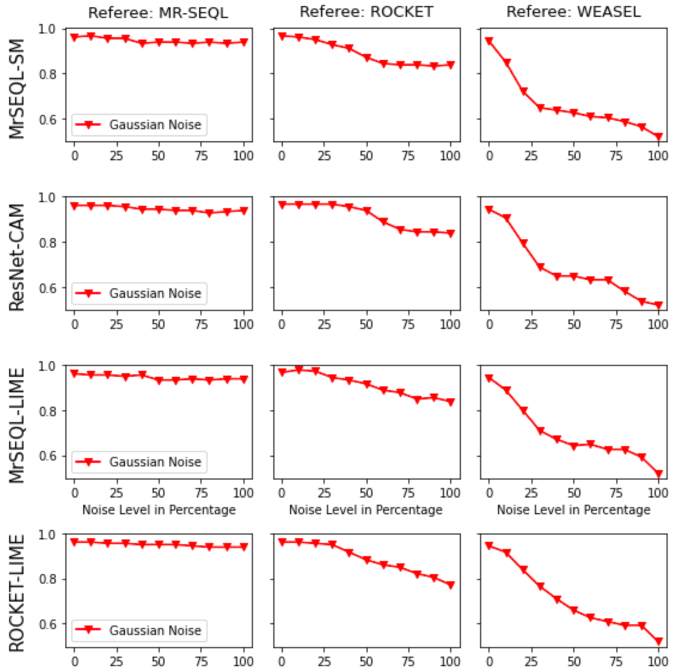

Fig. 5: The change in accuracy when perturbation is performed by adding Gaussian noise to the test time series for each explanation method from (top to down) with the three referee classifiers from (left to right) on the CMJ dataset.

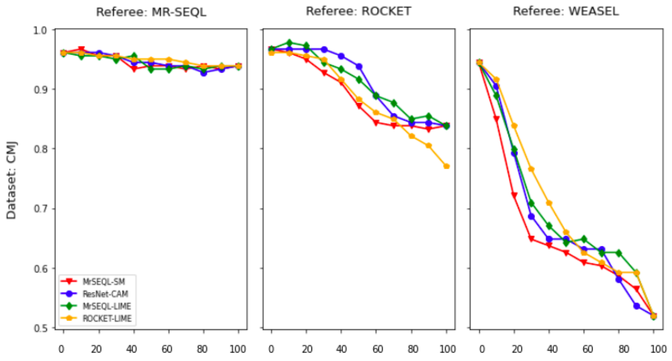

Fig. 6: Comparison of accuracy after perturbation with Gaussian noise for MrSEQL-SM, ResNet-CAM, MrSEQL-LIME and ROCKET-LIME using the CMJ dataset and the referee classifiers, MrSEQL, ROCKET and WEASEL. The lower curve indicates more impact of the explanation method on the referee classification accuracy.

Informativeness. The explanation methods are *ranked* and evaluated based on their explanation AUC for each classifier. Then the overall rank is calculated as the average-rank by aggregating over the referees as shown in Table 4. The explanation method that ranks the highest is taken as the most informative explanation method.

Table 4: Ranking of explanation methods based on their informativeness.

Dataset	Explanation Method	MrSEQL	ROCKET	WEASEL	Average Rank
CMJ	MrSEQL-SM	1.00	2.00	1.00	1.33
	MrSEQL-LIME	1.00	3.00	3.00	2.33
	ResNet-CAM	2.00	4.00	2.00	2.67
	ROCKET-LIME	3.00	1.00	4.00	2.67
Coffee	MrSEQL-SM	1.00	1.00	2.00	1.33
	ResNet-CAM	2.00	1.00	1.00	1.33
GunPoint	MrSEQL-SM	1.00	1.00	2.00	1.33
	ResNet-CAM	2.00	2.00	1.00	1.67
	MrSEQL-LIME	3.00	3.00	3.00	3.00

We make the following observations with regards to the average rank of explanation methods for each dataset:

- **CMJ:** MrSEQL-SM has the highest average rank and is thus the most informative followed by MrSEQL-LIME, ResNet-CAM and ROCKET-LIME. Both ResNet-CAM and ROCKET-LIME seem to be equally informative due to a similar average rank.
- **Coffee:** Both MrSEQL-SM and ResNet-CAM show the same average rank and hence are equally informative.
- **GunPoint:** MrSEQL-SM is the most informative followed by ResNet-CAM and then MrSEQL-LIME.

It is also important to note that the ranks vary across different referee classifiers and the referee classifier contributes towards the informativeness computation of the explanation methods. Even though MrSEQL-SM performs well with MrSEQL and WEASEL as referee classifiers for the CMJ dataset, it ranks second in the case of ROCKET as a referee classifier. This is also seen for ROCKET-LIME as it ranks first when trained with ROCKET itself as a referee classifier but not in other cases. Therefore, to obtain an aggregate behaviour of each explanation method over referees, an average rank is computed. MrSEQL-SM has the highest average rank followed by ResNet-CAM, MrSEQL-LIME and finally ROCKET-LIME. It is also important to note that these ranks depend on the problem statement and the dataset and can be different for different datasets. Further work is also needed to evaluate MrSEQL-LIME with Coffee dataset and ROCKET-LIME with GunPoint and Coffee datasets.

Runtime Analysis. The runtime of each explanation method to train a classifier, return the weights and plot the explanation in the form of a saliency map is calculated and displayed in Table 5. For each dataset, the run time is observed when performing the experiment for each of the explanation methods.

Table 5: Time (seconds) for model training, getting weights and getting the explanation for all the explanation methods for each dataset.

Dataset	Method	TrainingTime	GettingWeights	GettingExplanation	Total(sec)
CMJ	MrSEQL-SM	362.74	134.57	4.17	501.48
	ResNet-CAM	6.10	5.47	7.62	19.19
	MrSEQL-LIME	178.28	4438.53	4.75	4621.56
	ROCKET-LIME	33.26	3962.98	6.28	4002.53
Coffee	MrSEQL-SM	5.37	1.57	2.12	9.07
	ResNet-CAM	1.86	1.34	0.73	3.93
GunPoint	MrSEQL-SM	4.61	2.73	2.35	9.70
	ResNet-CAM	1.75	3.81	0.84	6.41
	MrSEQL-LIME	4.13	465.38	3.21	472.73

The following points summarize the findings for each dataset:

- **CMJ:** ResNet-CAM is the fastest to reproduce the results since a pre-trained model is used for training, otherwise ROCKET would be the fastest. After ResNet-CAM, we have MrSEQL-SM followed by ROCKET-LIME and then finally MrSEQL-LIME.
- **Coffee:** ResNet-CAM is somewhat faster than MrSEQL-SM in getting the weights and the explanation. Even if the classifier is fast to train, adding an explanation with LIME make the explanation step slow.
- **GunPoint:** MrSEQL-LIME is computationally expensive as opposed to ResNet-CAM and MrSEQL-SM.

We note that model-specific approaches such as MrSEQL-SM and ResNet-CAM are much faster than model agnostic approaches involving LIME, i.e., MrSEQL-LIME and ROCKET-LIME. Hence, even though ROCKET is an extremely fast classification method, its computational cost increases when it is combined with LIME to obtain an explanation.

Visualizing Saliency Mappings. The weights extracted for each of the explanation methods are mapped back to the time series in order to visualize them with the help of a saliency map. The most discriminative regions of the time series are highlighted in red by the explanation methods whereas the least discriminative regions are highlighted in blue by the explanation method on a scale of 0-100. Saliency maps help us validate the informativeness of the explanation methods. Each explanation method is compared with one another based on the generated saliency. It is clear from the figures shown in the sections below that all the methods give different explanations highlighting the importance of an objective evaluation approach. The CMJ dataset contains three classes - NORMAL, BEND and STUMBLE. Figure 7 shows the saliency maps generated by MrSEQL-SM, ResNet-CAM, MrSEQL-LIME and ROCKET-LIME for the CMJ dataset. It can be seen that each explanation method highlights a different region to be most informative. MrSEQL-SM appears to be the most informative since it clearly highlights the low-middle parts of the class NORMAL, the hump-middle part of the class BEND and the high peak part of the class STUMBLE (please refer to [11] for details on discriminative regions in this dataset). MrSEQL-LIME and ROCKET-LIME also highlight similar regions however, the explanations produced by MrSEQL-LIME is more similar to MrSEQL-SM than ROCKET-LIME. On the other hand, ResNet-CAM does not clearly highlight known discriminative parts in the time series of this dataset.

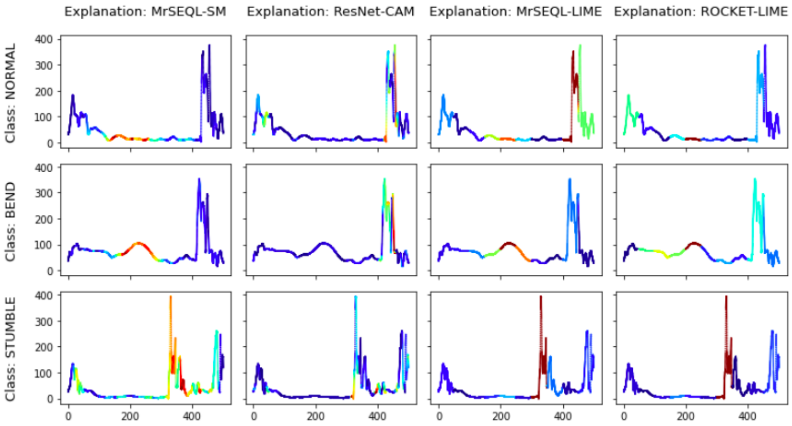

Fig. 7: Saliency maps produced by MrSEQL-SM, ResNet-CAM, MrSEQL-LIME and ROCKET-LIME explanation methods for an example time series from the three classes of the CMJ dataset.

Discussion. From the previous experiments we observe that explanation methods can indeed be quantitatively compared using the notion of informativeness based on ranking. The key takeaways from this work are summarized below.

- **Informativeness as an Evaluation Metric:** Through the ranking methodology, we observe that even though there is an overlap between the explanations produced by the explanation methods, MrSEQL-SM seems to be the most informative having the highest average rank across the three referee classifiers and the chosen datasets.
- **Computation Time:** LIME generates multiple perturbations of the new example and classifies it again in order to generate an explanation, which

results in high computational time. Hence, ROCKET-LIME and MrSEQL-LIME had a higher computation time. This is why it was challenging to evaluate these methods with all the datasets. Whereas in the case of MrSEQL-SM and ResNet-CAM we do not face this challenge since these simply use the trained model internals to generate explanations for a new example. Therefore, model-specific approaches like MrSEQL-SM and ResNet-CAM are faster as opposed to model-agnostic approaches like MrSEQL-LIME and ROCKET-LIME.

- **Impact of Referee Classifier:** The referee classifier can impact the classification accuracy and the explanation AUC of the explanation methods. We can also observe from Figure 6 that ROCKET and WEASEL appear to be more sensitive to the noise added during perturbation and show a significant reduction of accuracy as the amount of Gaussian noise added increases. This is not the case for MrSEQL as a referee classifier since the reduction is not that significant. Note that here MrSEQL only uses SAX features (in the time domain), while WEASEL uses SFA features (in the frequency domain), and ROCKET uses a mix of features in the time domain (i.e., convolution kernels) and features similar to the frequency domain features (i.e., dilation).
- **Saliency Mappings:**. Saliency maps can yield an accurate visual representation of what parts of the time series are considered important by the explanation method. This not only cross evaluates the ranking methodology but also represents the vector of weights in a visual manner.

5 Conclusion

This work aimed to quantitatively evaluate the informativeness of different model-specific as well as model-agnostic explanation methods for TSC. Through experimental results, we showed that TSC explanation methods can be evaluated and ranked based on their informativeness and that saliency-based visualizations support the results attained. Our simple ranking-over-referees technique can be implemented for practical applications in order to evaluate current TSC explanation methods or understand the classification decision-making process of TSC algorithms. In this work, four explanation methods are explored on three datasets, however, this technique can be adopted and expanded to evaluate other explanation methods and datasets based on the needs of a given problem statement. For future work we will extend the study of perturbation approaches, extend the set of referees and apply this methodology to more datasets that have available explanation ground truth. Given the fast growth of XAI and the amount of new methods proposed for explaining classifiers, we consider that having an effective methodology to objectively evaluate and compare these methods is very important to make sure that real progress is made and that the new explanation methods are actually useful.

Acknowledgments

This publication has emanated from research supported in part by a grant from Science Foundation Ireland through the SFI Centre for Research Training in Machine Learning (18/CRT/6183), the Insight Centre for Data Analytics (12/RC/2289_P2) and the VistaMilk SFI Research Centre (SFI/16/RC/3835). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. The authors would like to thank the reviewers for their constructive feedback.

References

1. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. Journal of the Royal Statistical Society. Series B: Statistical Methodology **82**(4), 1059–1086 (2020). https://doi.org/10.1111/rssb.12377
2. Bagnall, A., Flynn, M., Large, J., Lines, J., Middlehurst, M.: A tale of two toolkits, report the third: on the usage and performance of HIVE-COTE v1.0 (2020), `http://arxiv.org/abs/2004.06069`
3. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. DAMI. **31**(3), 606–660 (2017). https://doi.org/10.1007/s10618-016-0483-9
4. Dempster, A., Petitjean, F., Webb, G.I.: ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. DAMI. `https://link.springer.com/article/10.1007/s10618-020-00701-z`
5. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Information Sciences **239**, 142–153 (2013)
6. Dhariyal, B., Nguyen, T.L., Gsponer, S., Ifrim, G.: An Examination of the State-of-the-Art for Multivariate Time Series Classification An Examination of the State-of-the-Art for Multivariate Time Series Classification. ICDMW. (2020)
7. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017)
8. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning (2019)
9. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. DAMI. **33**(4), 917–963 (2019)
10. Kim, B., Khanna, R., Koyejo, O.O.: Examples are not enough, learn to criticize! criticism for interpretability. Neurips. vol. 29, pp. 2280–2288. Curran Associates, Inc. (2016)
11. Le Nguyen, T., Gsponer, S., Ilie, I., O'Reilly, M., Ifrim, G.: Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. DAMI. **33**(4), 1183–1222 (2019)
12. Lei, Y., Wu, Z.: Time series classification based on statistical features. EURASIP Journal on Wireless Communications and Networking **2020**(1), 1–13 (2020)
13. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. DAMI. **15**(2), 107–144 (2007)
14. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions (2017)
15. Metzenthen, E.: Lime for time code repository. `https://github.com/emanuel-metzenthin/Lime-For-Time/blob/master/demo/LIME-Pipeline.ipynb`

16. Molnar, C.: Interpretable machine learning. `https://christophm.github.io/interpretable-ml-book/`
17. Nguyen, T.T., Le Nguyen, T., Ifrim, G.: A model-agnostic approach to quantifying the informativeness of explanation methods for time series classification. AALTD. (2020)
18. Ozyegen, O., Ilic, I., Cevik, M.: Evaluation of Local Explanation Methods for Multivariate Time Series Forecasting pp. 1–13 (2020), `http://arxiv.org/abs/2009.09092`
19. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. KDD. 1135–1144 (2016)
20. Santos, T., Kern, R.: A literature survey of early time series classification and deep learning. CEUR Workshop Proceedings **1793** (2017)
21. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. DAMI. **29**(6), 1505–1530 (2015). https://doi.org/10.1007/s10618-014-0377-7, `http://dx.doi.org/10.1007/s10618-014-0377-7`
22. Schäfer, P., Högqvist, M.: Sfa: a symbolic fourier approximation and index for similarity search in high dimensional datasets. EDBT. pp. 516–527 (2012)
23. Schäfer, P., Leser, U.: Fast and accurate time series classification with weasel. CIKM. pp. 637–646 (2017)
24. Turing, A.: Sktime specifications. `https://www.turing.ac.uk/research/research-projects/sktime-toolbox-data-science-time-series`
25. Ye, L., Keogh, E.: Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. DAMI. **22**(1-2), 149–182 (2011)
26. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization (2015)