EFFECTS: Explorable and Explainable Feature Extraction Framework for Multivariate Time-Series Classification

Ido Ikar
Bar-Ilan University
Ramat Gan, Israel
ido.ikar@live.biu.ac.il

Amit Somech
Bar-Ilan University
Ramat Gan, Israel
somecha@cs.biu.ac.il

ABSTRACT

We demonstrate EFFECTS, an automated system for explorable and explainable feature extraction for multivariate time series classification. EFFECTS has a twofold contribution: (1) It significantly facilitates the exploration of MTSC data, and (2) it generates informative yet intuitive and explainable features to be used by the classification model. EFFECTS first mines the MTS data and extracts a set of interpretable features using an optimized *transform-slice-aggregate* process. To evaluate the quality of EFFECTS features, we gauge how well each feature distinguishes between every two classes, and how well they characterize each single class. Users can then explore the MTS data via the EFFECTS Explorer, which facilitates the visual inspection of important features, dimensions, and time slices. Last, the user can use the top features for each class when building a classification pipeline.

We demonstrate EFFECTS on several real-world MTSC datasets, inviting the audience to investigate the data via EFFECTS Explorer and obtain initial insights on the time series data. Then, we will show how EFFECTS features are used in an ML model, and obtain accuracy that is on par with state-of-the-art MTSC models that do not optimize on explainability.

CCS CONCEPTS

• **Mathematics of computing** → **Time series analysis**; **Exploratory data analysis**; • **Information systems** → *Temporal data*.

KEYWORDS

Explainability; Exploration of time series data

ACM Reference Format:
Ido Ikar and Amit Somech. 2023. EFFECTS: Explorable and Explainable Feature Extraction Framework for Multivariate Time-Series Classification. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3614740

1 INTRODUCTION

The development of data science pipelines for multivariate time series classification (MTSC) has become increasingly important in recent years due to the proliferation of computing devices, platforms, and IoT sensors [9]. However, exploring the data and extracting important features is especially difficult in the context of MTSC. Unlike standard, tabular data, the raw signals in MTSC data are often unintuitive and difficult to interpret. Therefore, users must possess domain knowledge and employ appropriate data transformation and aggregations to discover features that point out the differences between the time series classes.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614740

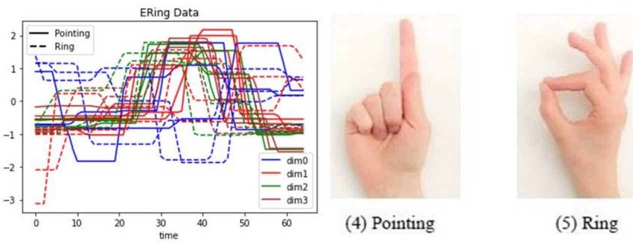

Figure 1: ERing dataset sample (taken from [10])

While feature extraction from time series data can be done automatically by statistical [1] or shape-based tools [11], the resulting features from such frameworks are often complex and difficult to interpret. For illustration, consider the following example.

Example 1.1. Data Scientist Clarice is building a classification model for the ERing dataset [10]. The data was generated by a ring device with 4 electric field sensors that captured data from users who were asked to perform hand gestures, such as Fist, Point, Ring, and Grasp (See Figure 1 for a sample).

As the raw sensor data is not very meaningful, she applies the Tsfel [1] tool, which extracts hundreds of statistical characteristics from the time series. She then feeds the extracted tabular data to a decision tree classifier. When looking at the important features, she sees that the top-1 feature is `fft_coeff_attr_'angle'_coeff_1`, but it is difficult to interpret and understand. □

To this end, we demonstrate EFFECTS, an Explorable and explainable Feature Extraction framework for multivariate Time Series Classification. EFFECTS efficiently extracts meaningful, interpretable features from the time series data and, more importantly, it then explains how they contribute to the identification of each class or the separation between different classes.

In a nutshell, EFFECTS first generates a set of candidate features from the time series data using a three-step *Transform-Slice-Aggregate* process, where a range of transformation functions is applied on different intervals of the raw signals, then a scalar value is computed via numerous aggregation functions. As the number of such features is exponentially large, EFFECTS uses an efficient, linear-time *slice discovery algorithm* that effectively detects promising time slices, in which in-class signals are *similar* and cross-class

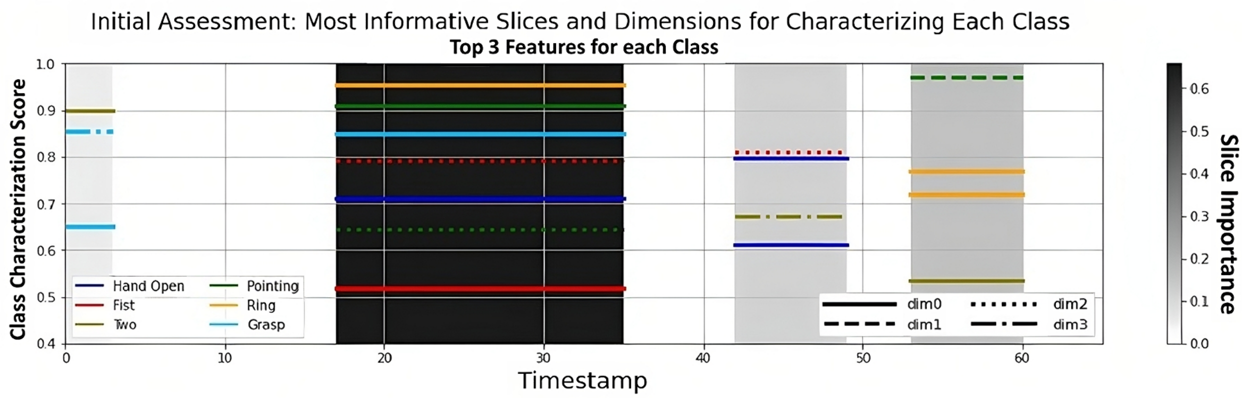

Figure 2: EFFECTS Class Characterization Overview. This view illustrates the top-k features and corresponding time slices that distinguish each class. Each horizontal line represents a single EFFECTS feature, where its length and location depict the feature's time interval, its color depicts the class, and the line type represents the dimension that the feature is calculated from. Important time slices appear in a darker color.

signals are *different*. Next, EFFECTS calculates multiple scores that measure their ability to distinguish between classes and to identify each individual class. The scores are then aggregated in several ways to asses the overall feature importance, as well as of time intervals and dimensions.

The feature scores are then utilized by the EFFECTS Explorer which assists users in understanding the MTS dataset and effectively building an ML classification pipeline. Beginning with an overview of locations with important features that characterize each class (see, e.g., Figure 2), then further examining more specific aspects such as: *What are the top features that separate the classes 'Pointing' and 'Ring'? What time slices, on which dimension, characterize the 'Ring' gesture?* (See, e.g., Figure 4). Last, the user can choose, based on the feature scores and the insights gained thus far, which of the obtained features should be included in a classification ML model.

Interactive Demonstration. We demonstrate EFFECTS over a variety of MTSC datasets, showing how quickly users can obtain an initial assessment of the data, and understand, e.g., what classes are difficult or easy to isolate, and how certain classes can be distinguished from one another. Finally, we will show that the informative features discovered by EFFECTS are also highly useful for ML-based classification, by comparing the accuracy of a model based on EFFECTS features, compared to state-of-the-art MTSC models such as LSTM-FCN [5] and WEASEL+MUSE [7] (both optimize only on the accuracy of prediction, rather than on the explainability or explorablity of the features).

Related Work. Numerous tools exist for extracting informative features from time series data [1, 2, 12]. First, tools such as Tsfel [1] automatically extract a multitude of features comprising of statistical characteristics of the time series. Unfortunately, as mentioned in Example 1.1, such features may be highly difficult to interpret and gain insights from. Another common approach is *Shapelets-based* algorithms, which detect short, distinctive patterns within the time series [6, 12]. However, shape-based features may be difficult to explicitly describe, as they represent abstract visual patterns.

A parallel line of research is explaining neural-network models for MTSC [2, 3]. Such solutions offer ad-hoc convolutional neural networks (CNN), that also output an estimation of important time intervals and dimensions, alongside the prediction. While using CNN for MTSC tasks is convenient and hardly requires processing the raw data, the explanations provided are rather implicit and are not directly connected to the raw data.

In contrast, EFFECTS generates intuitive and explicit features, and is also model agnostic – can be used in various pipelines rather than just CNN. In addition, EFFECTS provides a dedicated exploration framework for the MTSC task, allowing users to gain useful insights before building their classification ML pipeline.

2 EFFECTS ARCHITECTURE

An illustration of our system architecture is provided in Figure 3. We begin this section by briefly describing our model and common time-series operations, then present the main components of EFFECTS: Feature Extractor, which uses the Slice Discovery Algorithm for efficiently detecting promising time intervals; EFFECTS Feature Scoring, used for measuring the quality of output features; EFFECTS Explorer, which aggregates and processes the features information used for an interactive exploration and investigation of the time series data; Last, we describe how the features can be filtered and used by data scientists when building ML classification pipelines.

2.1 Model, Definitions, Time Series Operations

Following [7], a single, multivariate time series (MTS) T is a matrix of size (m, n), such that $T_{d,t}$ is the value of the signal projected on dimension $1 \leq d \leq m$, at time point $1 \leq t \leq n$. An MTSC dataset $D = \langle N \times T, CL \rangle$ contains N multivariate time series $T^1, \ldots, T^N$ alongside a target *classification* vector CL that maps each individual time series T to one of C class labels. EFFECTS supports various, known time-series operations from the three following categories: transformations, time-interval slicing, and aggregations:

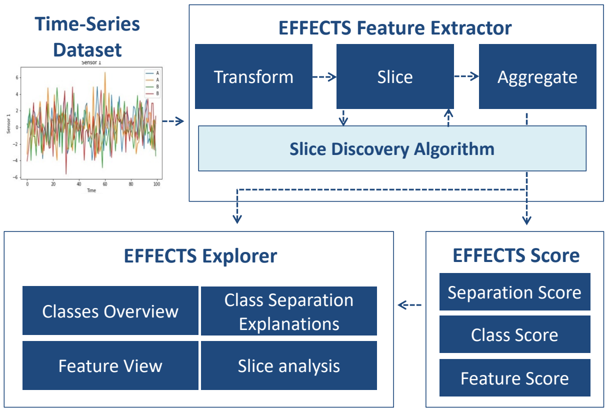

Figure 3: EFFECTS Workflow & Architecture

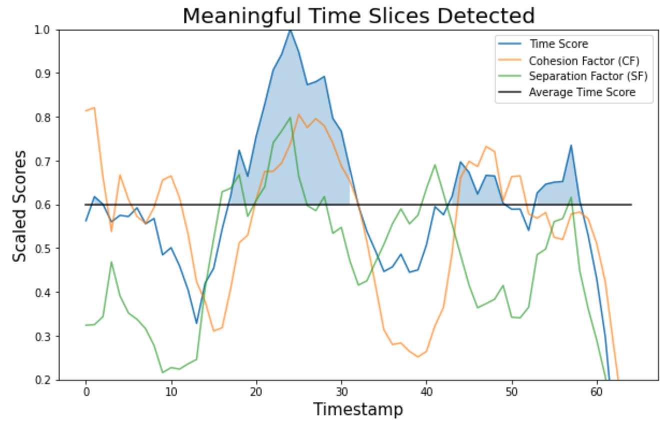

Figure 4: Meaningful Time Slices Discovered

(1) Transformations. These operations are mainly used to de-noise the raw signal and refine it in order to extract meaningful information and data patterns. We support various known transformation functions such as *diff* (i.e., the derivative of the series), *cumulative sum*, and *Discrete Fourier Transform.*

(2) Time Slicing. The majority of signal information useful for the classification task often resides only in a limited number of time intervals, rather than on the entire range. Therefore, features that are calculated on the entire series may be less meaningful and reduce the MTSC model performance [8]. we therefore divide the full range $(1, n)$ to *time slices* (a, b), s.t. $1 \leq a < b \leq n$, and extract features only on *meaningful*, informative such slices. As the number of possible such slices is exponential, we use an optimized algorithm for *slice discovery*, as described below.

(3) Aggregations. Aggregation function of the form $f^A : \mathbb{R}^n \to \mathbb{R}$ are used to generate scalar values from transformed time-slices obtained in (1) and (2). we support various aggregation functions such as *max*, *min*, *sum*, *trend*, and *peak.*

2.2 Feature Extraction, Slice Discovery

As mentioned above, EFFECTS features are combinations of transformation, slicing and aggregation operations. Namely, each EFFECTS feature F is defined by $F := f^A(s_{a,b,d})$, where $s_{a,b,d}$ is the transformed slice between time a and b, from dimension d. Naturally, this can result in an exponential number of features, mainly due to the large number of possible time slices.

We therefore devise a Slice Discovery Algorithm that effectively detects promising, informative time slices of a transformed MTS. Ideally, such candidate time intervals should contain data patterns that (1) identify a class, by demonstrating *similar* trends in same-class time series, and (2) separate between different classes, by demonstrating *contrasting* trends for time series of different classes.

For each individual timestamp $1 \leq t \leq n$, we calculate the following scores:

(1) *cohesion factor* - $CF(t)$ which represents the degree of similarity between time series of the same class. $CF(t)$ is calculated by the negative sum of entropy, for each class c_i, namely, $CF(t) = -\sum_{d=0}^{d<m} \sum_{c \in C} H\left(T^{\star}_{d,t}[c]\right)$ where $T^{\star}_{d,t}[c]$ is the subset of $T^{\star}_{d,t}$ that belongs to class c.

(2) *Separation Factor* – $SF(t)$, which measures the disparity of time series that belong to different classes. The $SF(t)$ score is derived from the sum of KL-divergence for every two classes $c_i, c_j \in C$, i.e., $SF(t) = \sum_{c_i,c_j \in C} D_{KL}\left(T^{\star}_{d,t}[c_i] \mid T^{\star}_{d,t}[c_j]\right)$.

Next, the CF and SF scores are scaled between 0 and 1 using *min-max* scaling, and we calculate the final time-score given by $\widetilde{CF}(t) + \widetilde{SF}(t)$ (where $\widetilde{CF}$ and $\widetilde{SF}$ are the scaled individual scores).

Finally, we derive *meaningful* time slices by returning all intervals $1 \leq a < b \leq n$ with a consecutive, *high* time-score (we use the mean scaled score as a threshold in our implementation). Figure 4 illustrates the meaningful time series (light-blue areas) extracted by EFFECTS for the ERing dataset, the CF and SF scores, and the cumulative time score.

2.3 Calculating EFFECTS Feature Scores

While computed from promising time slices, the final features generated by EFFECTS Extractor may vary in their ability to separate or identify the different classes. We therefore analyze the utility of an EFFECTS feature, by performing a class-pairwise evaluation of its ability to distinguish between each two classes. We then leverage this analysis to gain an (1) overall importance assessment of a feature, dimension, and time slice, and (2) an identification difficulty estimation of a class.

Given a feature F and two classes $c_i, c_j \in C$, we define the *pairwise separation score* $Score_{i,j}(F)$, by calculating the overlap between the feature *value distributions* for the two classes. Intuitively, the *lower* the overlap between the two classes' feature distribution – the *higher* the feature score. Let X be an $N \times |\mathcal{F}|$ matrix that contains the extracted feature values for all N time series samples in our dataset. We further denote by $X^F[c]$ the vector containing the values of feature F, for all rows in X that are labeled with class c. We then determine the *histogram overlap* between $X^F[c_i]$ and $X^F[c_j]$ by first dividing the joint space to $|M|$ bins of equal width, then checking the class heterogeneity (i.e., entropy) $H(m)$ of each bin m. Finally, the class-pairwise feature score is defined as follows:

$$Score_{i,j}(F) = \frac{\sum_{m \in M} |m| \cdot H(m)}{|X^f[c_i]| + |X^f[c_j]|}$$

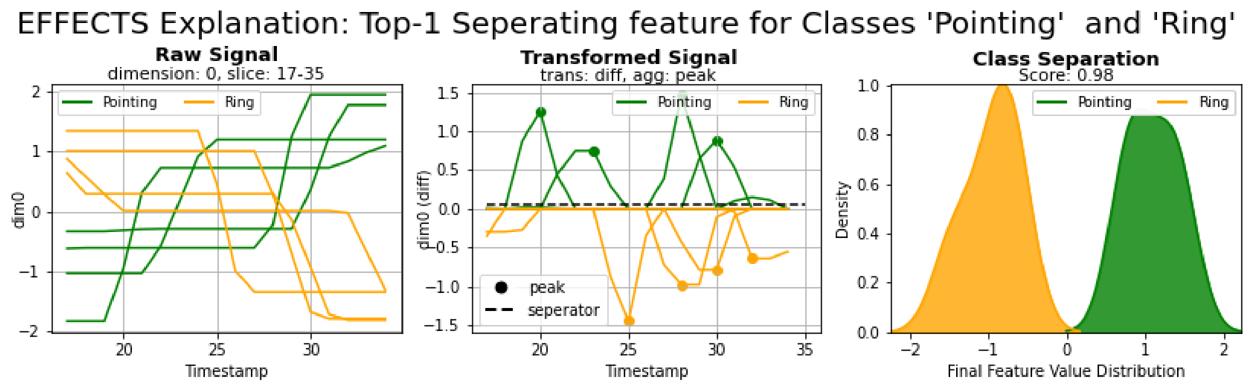

Figure 5: EFFECTS Classes Separation Explanation. This view shows the best feature for distinguishing between the classes 'pointing' and 'ring'. The figure from the left shows samples from the raw signals on Dimension 0, between timestamps 17 and 35. The middle figure shows the *transformed* signals (using *diff*), with their aggregated values (using *peak*) marked by round dots. One can already see that the aggregated values of 'Pointing' samples are high (above 0.5) and 'Ring' samples are low (below -0.5). Indeed, as shown in the right-hand-side figure, which depicts the feature distribution for the two classes, the separation obtained by this figure is very high (score of 0.98).

Using the pairwise feature score as a basis, we can now estimate the overall importance of a feature. This is simply done by aggregating over the pairwise scores, e.g., using $Fscore_{avg}(F) = \sum_{c_i,c_j \in C} Score_{i,j}(F)$, or $Fscore_{max}(F) = \max_{c_i,c_j \in C} Score_{i,j}(F)$.

Next, we calculate the *time-slice importance*, by summing the $Fscore$ of the features extracted from it, i.e., $SliceScore(a,b) := \sum_{F \in \mathcal{F}_{a,b}} Fscore(F)$ where $\mathcal{F}_{a,b}$ are all features extracted from the time slice a, b. In a similar manner, we compute the importance of dimensions and aggregation functions.

Last, we want to estimate the difficulty of identifying a given class c_k. We do so by first calculating the *Class Characterization Score* of a single feature F, which reflects its importance in characterizing c_k: $Cscore(F, c_k) = \sum_{c_i \in C} Score_{k,i}(F)$. The $Cscore$ of a class is then given by $Cscore(c_k) = \sum_{F \in \mathcal{F}} Cscore(F, c_k)$.

2.4 EFFECTS Explorer

The explicit features extracted by EFFECTS alongside the scores calculated for the features, time slices, dimensions, and classes are used by EFFECTS Explorer in order to provide meaningful insights for the users that may assist them in understanding the data and building an effective classification pipeline.

The first view provided by EFFECTS Explorer is the *Class Characterization Overview*, as depicted in Figure 2. This visualization utilizes the class characterization and slice importance scores. It shows the top-k features w.r.t. the characterization scores for each class, and depicts their origin dimension and time slice. This allows the user not only to grasp the more informative time slices and dimension, but also to understand how well their corresponding features identify each class. EFFECTS Explorer also assists users in examining what distinguish between a certain pair of classes, via the *Class Separation Explanation*, as depicted in Figure 5. Given two input classes, the system outputs the Top-k features that separate between those classes, in a threefold visual explanation that showcases how the features are explicitly created from the raw data. In addition, EFFECTS provides visual explanation used for an up-close examination of individual features, time slices, and dimensions, assisting the users in answering questions such as *How many classes can a certain feature identify What is the most useful dimension for identifying the 'Fist' gesture? Which pairs of classes can be distinguished using the first 10 timestamps?*

Using EFFECTS Features in ML Classification Pipelines. The features extracted by EFFECTS are intuitive to understand yet are also quite powerful when used in by classification models (See our Github repository [4] for preliminary results). To perform feature selection, users can filter the features using their overall feature scores (both $Fscore_{avg}$ and $Fscore_{max}$) as well as according to their class characterization score $Cscore$.

3 IMPLEMENTATION & DEMONSTRATION

To assist real-life data scientists, we implemented EFFECTS as a Python library, compatible with the popular Scikit-learn framework for ML models. Our transformations and aggregations are built with Numpy and Scipy, and EFFECTS Explorer uses Matplotlib for creating visualizations. See [4] for our code base.

In our demonstration, we invite the audience to use EFFECTS for exploring and easily building ML pipelines for one or more of our MTSC datasets. After a brief explanation about the dataset and classification task, the participants will first use EFFECTS Explorer and observe the initial important insights it provides, such as which classes are easy/difficult to distinguish; what time slices and dimensions are of higher importance; and what particular EFFECTS features can be used for separating specific classes.

In the second part of the demonstration, participants will use the EFFECTS features in an ML classification model of their choice, and gauge the features' contribution to the accuracy of the model. For reference, we will present the accuracy results of state-of-the-art MTSC models (e.g., Tsfel [1], LSTM-FCN [5], and WEASEL+MUSE[7]).

Last, we provide a look under the hood of EFFECTS – showing the time-slice identified by our Slice Discovery Algorithm (as in Figure 4), running times analysis, as well as a comparison of our framework to explainable neural-network solutions such as [2, 3].

REFERENCES

[1] Marília Barandas, Duarte Folgado, Fernandes, et al. 2020. TSFEL: Time series feature extraction library. *SoftwareX* 11 (2020), 100456.

[2] Paul Boniol, Mohammed Meftah, Emmanuel Remy, and Themis Palpanas. 2022. DCAM: Dimension-Wise Class Activation Map for Explaining Multivariate Data Series Classification. In *SIGMOD*.

[3] Tsung-Yu Hsieh, Suhang Wang, Yiwei Sun, and Vasant Honavar. 2021. Explainable multivariate time series classification: a deep neural network which learns to attend to important variables as well as time intervals. In *Proceedings of the 14th ACM international conference on web search and data mining*. 607–615.

[4] Ido Ikar and Amit Somech. 2023. https://github.com/11k4r/EFFECTS.

[5] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. 2019. Multivariate LSTM-FCNs for time series classification. *Neural Networks* 116 (2019), 237–245.

[6] Chen Li, Yuan Wan, Wenjing Zhang, and Huanhuan Li. 2022. A two-phase filtering of discriminative shapelets learning for time series classification. *Applied Intelligence* (2022), 1–19.

[7] Patrick Schäfer and Ulf Leser. 2017. Multivariate time series classification with WEASEL+ MUSE. *arXiv:1711.11343* (2017).

[8] Romain Tavenard, Simon Malinowski, Laetitia Chapel, et al. 2017. Efficient Temporal Kernels between Feature Sets for Time Series Classification. In *ECML-PKDD*.

[9] Nikhil M Thoppil, V Vasu, and CSP Rao. 2021. Deep learning algorithms for machinery health prognostics using time-series data: a review. *JVET* 9, 6 (2021), 1123–1145.

[10] Mathias Wilhelm, Daniel Krakowczyk, Frank Trollmann, and Sahin Albayrak. 2015. ERing: Multiple Finger Gesture Recognition with One Ring Using an Electric Field. In *iWOAR '15*. Article 7.

[11] Xiunan Zou, Xiangwei Zheng, Cun Ji, and Yuang Zhang. 2021. An improved fast shapelet selection algorithm and its application to pervasive EEG. *Personal and Ubiquitous Computing* (2021), 1–13.

[12] Jingwei Zuo, Karine Zeitouni, and Yehia Taher. 2019. Exploring Interpretable Features for Large Time Series with SE4TeC.. In *EDBT*. 606–609.