Multivariate Asynchronous Shapelets for Imbalanced Car Crash Predictions

Mario Bianchi[1], Francesco Spinnato[1,2] ✉, Riccardo Guidotti[1,2], Daniele Maccagnola[3], and Antonio Bencini Farina[3]

[1] University of Pisa, Pisa, Italy, {name.surname}@unipi.it
[2] ISTI-CNR, Pisa, Italy, {name.surname}@isti.cnr.it
[3] Generali Italia, {name.surname}@generali.com

Abstract. Real-time vehicle safety and performance monitoring through crash data recorders is transforming mobility-related businesses. In this work, we collaborate with Generali Italia to improve their in-development automatic decision-making system, designed to assist operators in handling customer car crashes. Currently, Generali uses a deep learning model that can accurately alert operators of potential crashes, but its black-box nature can hinder the operator's trustworthiness in the model. Given these limitations, we propose MARS, an interpretable shapelet-based classifier using novel multivariate asynchronous shapelets. We show that MARS can handle Generali's highly irregular and imbalanced time series dataset, outperforming state-of-the-art classifiers and anomaly detection algorithms, including Generali's black-box system. Further, we validate MARS on multivariate datasets from the UEA repository, demonstrating its competitiveness with existing techniques and providing examples of the explanations MARS can produce.

Keywords: Time Series · Crash Prediction · Explainable AI

1 Introduction

The availability of real-time sequential data, paired with accurate Artificial Intelligence (AI) decision-making systems, is transforming the business landscape for many mobility-related companies [30]. Crash Data Recorders (CDRs) are increasingly being used in cars to monitor safety measures, establish human tolerance limits, and quantify vehicle status [36]. These recorders are usually installed on the airbag control module, collecting data before and after a crash [36]. Recently, with the use of powerful Machine Learning (ML) models, these devices have become a valuable source of data for both academic research and businesses, such as insurance companies, to monitor and improve customer service quality [34].

In this work, we collaborate with Generali Italia, one of the largest global insurance and asset management providers[4]. Generali is developing an automatic decision-making system to provide first aid to its customers. As part of their

[4] https://www.generali.it

insurance products, Generali offers to install a CDR in their customers' vehicles. This system monitors the vehicle during its use and, among other services, tracks speed and acceleration on the three car axes. This data is used to train a deep learning model that enables the AI system to alert a Generali operator of possible car crashes. By examining the CDR data and model predictions, the operator can make an informed decision and only contact the customer if assistance is really necessary. The main challenge Generali faces is that the operator cannot interpret the predictions of the deep learning classifier. This lack of transparency could hinder the operator's understanding of the model's outcome, potentially leading to a lack of trust, especially if it produces incorrect classifications.

In such a critical scenario, eXplainable AI (XAI) [5] is essential for interpreting these black-box predictions to ensure reliability in decision-making. XAI for multivariate time series classification is an emerging yet underdeveloped field [27]. This challenge is exacerbated with big, highly imbalanced, and irregular datasets like those used by Generali, where off-the-shelf XAI approaches often fail due to the limits of their implementation [19,25]. In [26], we addressed a similar problem by adapting existing XAI approaches in a workflow to replace Generali's black-box with an interpretable pipeline. Although the results were promising, they were still inferior in performance compared to Generali's black-box classifier. Additionally, the explanations were based on *univariate* shapelets [33], i.e., subsequences that are most representative of a given class, extracted from *single* signals of the time series. This was a significant limitation for Generali because the explanations did not utilize the multivariate nature of the data, making them potentially less useful for understanding the nature of the crash.

Given these limitations, in this work, we propose MARS (Multivariate Asynchronous Shapelets), an interpretable-by-design approach based on *multivariate shapelets*. MARS works on irregular time series data and accurately distinguishes between crash and no-crash instances. We demonstrate that our proposal outperforms state-of-the-art classifiers and anomaly detection algorithms, as well as the black-box currently used by Generali. Furthermore, we evaluate the performance of MARS on several multivariate datasets from the UEA repository, showing that our approach is on par with current state-of-the-art competitors. Finally, we provide a qualitative example of the explanation provided by MARS.

The rest of the paper is organized as follows. In Section 2, we present the related literature on crash prediction and underline the differences with the current proposal. In Section 3, we introduce all the concepts necessary for understanding our proposal, which is detailed in Section 4. In Section 5, we present our empirical evaluation of the proposed approach, and finally, in Section 6, we draw our conclusions, discussing limitations and future challenges.

2 Related Works

The main problem faced in this work consists of identifying car crashes from imbalanced multivariate time series data. The literature on crash prediction studies car accidents from various perspectives [18], with the most common

distinction being between *real-time* and *long-term* crash prediction. Long-term crash prediction involves using advanced data analytics and machine learning techniques to forecast the likelihood and severity of vehicle accidents over an extended period. This process includes collecting and analyzing vast amounts of data from various sources, such as driving patterns [32], road conditions, mobility traces, and road networks [22]. By identifying patterns and trends within this data, predictive models can be developed to anticipate future crashes. A significant portion of the literature is dedicated to real-time crash prediction, which examines collision areas and conditions [23,31] using data from internal and external sensors that capture mobility features [20], visual information [24], or physiological parameters [1]. The task tackled in our work is somewhat unique, given that the crash prediction is not employed to predict close or distant future classes but to explain crashes that have already happened, enabling appropriate and effective responses. Specifically, in our setting, crash prediction consists of using a series of inputs to train a classifier to detect whether a crash has occurred such that Generali operators can contact customers if assistance is needed.

In car crash prediction tasks, the classes are typically imbalanced. Therefore, the problem can be viewed either as a classification or as an anomaly detection task, where the minority class is considered an outlier. Considering this task as a standard classification problem, any of the common approaches in the literature can be applied, such as tree-based [35] or recurrent models [12]. The state-of-the-art time series classifier is ROCKET [8], which uses random convolutional kernels and different kinds of pooling to quickly and accurately predict the class of time series. The main drawback of ROCKET is that it is not interpretable. Contrary to the aforementioned approaches, in this work, we develop an interpretable classifier, which also performs well on time series anomaly detection. Car crash prediction can also be seen as an anomaly detection task. A recent survey [4] highlights how only a handful of approaches tackle the problem of *whole time series* anomaly detection, while most approaches usually focus on finding anomalous points or subsequences inside the time series. One of the best-performing models in this field is ROCKAD [28], a semi-supervised method using ROCKET as a feature extractor and k-nearest neighbor anomaly detectors to produce an anomaly score. Although methods that leverage outlier detection are a valid alternative in the case of highly imbalanced classifications, it is important to note that outlier detection algorithms are not easily explainable [29].

Arguably, some of the most interpretable algorithms for time series classification are shapelet-based, meaning they use the distances between time series and subsequences as input features for classification algorithms. These approaches focus on finding subsequences, known as shapelets, that can discriminate between classes regardless of their position [33], and they differ in how they extract these shapelets and build the subsequent classifier. These algorithms are considered interpretable because shapelets can be visually analyzed by the user and associated with different classes based on their shape [27]. Most shapelet-based approaches focus on univariate time series and univariate shapelets, while there are significantly fewer works on multivariate time series data. Specifically, since

the theoretical introduction of the concept of shapelets that span more than one signal, i.e., *multivariate shapelets* [10], the major theoretical analysis was proposed in [6], whereas the only practical implementation is [21], which is an architecture for shapelet learning by embedding them as trainable weights in a multi-layer neural network. The main limitation of [21] is that shapelets are embedded in a deep neural network model and need to be aligned temporally. This limits the flexibility of such an approach, given that temporal data is usually not perfectly aligned. In contrast, in our proposal, we allow for *asynchronous shapelets*, i.e., shapelets that are not perfectly aligned temporally, since their extraction. Finally, a few works exist that aim to extract 2D domain-specific shapelets [3,16], but they can only be used with trajectory data, i.e., with latitude and longitude signals.

In our previous work [26], we addressed a related problem by employing existing XAI techniques to construct a novel XAI pipeline to explain a black-box model's predictions on a different car crash dataset. The present study diverges significantly by introducing an entirely new interpretable-by-design algorithm rather than relying on existing state-of-the-art post-hoc XAI methods. Additionally, we rigorously test our novel proposal on a new, larger, and more imbalanced dataset, and further demonstrate its robust performance across multiple other time series classification datasets.

3 Background

This section introduces all the necessary concepts to understand our proposal. We begin by defining time series data, particularly multivariate time series.

Definition 1 (Multivariate Time Series). *A* multivariate time series, $X = \{\mathbf{x}_1, \dots \mathbf{x}_d\} \in \mathbb{R}^{d \times m}$, *is a collection of* $d > 1$ *signals, each containing* m *real-valued observations,* $\mathbf{x} = [x_1, \dots, x_m] \in \mathbb{R}^m$.

Time series are usually collected in so-called time series datasets (or panels) for supervised tasks, such as time series classification.

Definition 2 (TSC Dataset). *A* time series classification dataset $\mathcal{D} = (\mathcal{X}, \boldsymbol{y})$ *is a set of* n *time series,* $\mathcal{X} = \{X_1, \dots X_n\} \in \mathbb{R}^{n \times d \times m}$, *with a vector of assigned classes,* $\boldsymbol{y} = [y_1, \dots, y_n] \in \{0, \dots, c-1\}^n$, *where* c *is the number of classes.*

Time Series Classification (TSC) is defined as follows.

Definition 3 (TSC). *Given a TSC dataset* $\mathcal{D}$, Time Series Classification *is the task of training a function* f *from the space of possible inputs* $\mathcal{X}$ *to a probability distribution over the class variable values in* $\boldsymbol{y}$.

If $c = 2$, we have a binary classification problem, such as that proposed by Generali, while if $c > 2$, we have a multiclass problem. In this work, we tackle imbalanced TSC. Let n_i denote the number of instances belonging to class i, where $i \in \{0, \dots, c-1\}$. In the case of class imbalance, at least one class j

exists, such that $n_j \ll n_i$ for some $i \neq j$. This imbalance can adversely affect the performance of traditional classification algorithms, which often assume a roughly equal distribution of instances across classes.

We focus on interpretable TSC algorithms, where we can also access an explanation for the model's behavior besides the model's output, which can help assess the reasoning behind its prediction. In particular, we focus on shapelet-based models [33]. To define a shapelet, we need to start from a subsequence.

Definition 4 (Subsequence). *Given a signal* $\boldsymbol{x} = [x_1, \ldots, x_m]$*, a univariate* subsequence *of length* l*, is an ordered sequence of values from* $\mathbf{x}$ *such that* $1 \leq j \leq m - l + 1$*, i.e.,* $\boldsymbol{s} = [x_j, \ldots, x_{j+l-1}]$.

Shapelets are time series subsequences that are highly representative and discriminative for a particular class in a time series dataset. Shapelet-based classifiers are some of the most common models for interpretable TSC [27]. Three steps are usually performed to train a shapelet-based model: shapelet extraction, matching, and transformation. *Shapelet extraction* can be performed in many supervised and unsupervised ways, but usually, a good shapelet discriminates well between classes, i.e., it refers mostly to instances belonging to a specific class compared to other classes. *Shapelet matching* refers to the act of comparing the shapelet to a given time series instance, usually through a distance measure such as the minimum sliding-window Euclidean distance between the shapelet and the time series. Finally, the so-called *Shapelet Transform* [17] transforms the dataset into a tabular form, given a set of real or synthetic shapelets.

Definition 5 (Shapelet Transform). *Given a time series dataset* $\mathcal{X}$ *and a set* S *containing* h *shapelets, the* Shapelet Transform *converts* $\mathcal{X} \in \mathbb{R}^{n \times d \times m}$ *into a real-valued matrix* $T \in \mathbb{R}^{n \times h}$*, obtained by taking the distance between each time series in* $\mathcal{X}$*, and each shapelet in* S.

This is the standard formalization for *univariate shapelets*, i.e., shapelet extracted from single time series signals. In [6], three different kinds of *multivariate shapelets* are introduced. First are independent shapelets (ST_I), which are subsequences extracted independently from each signal and assessed individually against each corresponding dimension of the multivariate time series. ST_I method extracts identical shapelets that would occur in a standard univariate Shapelet Transform. Second, multidimensional dependent shapelets (MST_D) maintain phase alignment across all channels in the extraction and matching phases. The process involves sliding, extracting the multivariate shapelet along the time series, and normalizing both the shapelet and subsequences dimension-wise to find the best match. The minimum distance indicates the closest match. Finally, multidimensional independent shapelets (MST_I) are extracted from each dimension's subsequence, as in MST_D, but matching is performed by finding the minimum distance to its respective dimension independently. Distances from the different dimensions are then aggregated via a sum. In the following, we go beyond these definitions, proposing multivariate *asynchronous* shapelets.

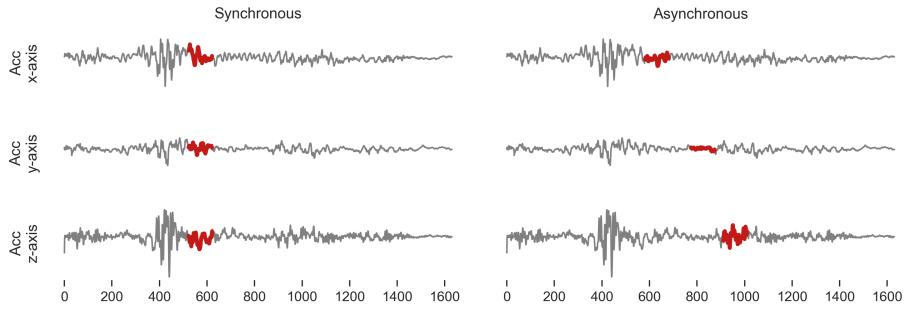

Fig. 1. Example of synchronous (left) and asynchronous (right) shapelets extracted from the x, y, z acceleration signals of a car from Generali's dataset.

4 Multivariate Asynchronous Shapelets

In this section, we present MARS, our proposal to extract Multivariate Asynchronous Shapelets to build an interpretable-by-design time series classification model based on *multivariate shapelets*. One of the main limitations of all the existing multivariate shapelet approaches presented in Section 3, is that they are extracted from temporally aligned indices. This is a problem, mostly when the time series is highly irregular, e.g., when the signals have different sampling rates and vary greatly in length, such as in Generali's setting. For this reason, we define here multivariate asynchronous shapelets.

Definition 6 (Multivariate Asynchronous Shapelet). *Given a time series, X, a* multivariate asynchronous shapelet *of length l is a collection of subsequences from each signal in X, each starting at possibly different timesteps j_k, i.e., $S = \{\mathbf{s}_1, \dots, \mathbf{s}_d\} = \{\mathbf{x}_{1,j_1:j_1+l-1}, \dots, \mathbf{x}_{d,j_d:j_d+l-1}\} \in \mathbb{R}^{d \times l}$, where $\mathbf{x}_{k,j_k:j_k+l-1} = [x_{k,j_k}, \dots, x_{k,j_k+l-1}] \in \mathbb{R}^l$ for $k = 1, \dots, d$.*

In simple terms, a multivariate shapelet is a short, fixed-length subsequence extracted from each dimension of a multivariate time series. Each dimension, $\mathbf{s}$, of the shapelet, S, corresponds to a segment of the original time series. Traditionally, these segments are synchronized across all dimensions to maintain temporal alignment, as shown in Figure 1 (left). In this work, we propose using different starting indices, j_k, for each dimension of X, as shown in Figure 1 (right). This allows for shapelets that are not aligned temporally since the extraction phase, hence the name *asynchronous shapelets*.

4.1 Shapelet Extraction

To enforce a given synchronicity of the shapelet dimensions, we introduce a constraint that limits how distant the starting points for each dimension of the shapelet can be. We call this parameter the *asynchronicity limit*, α, which is a constraint on the maximum allowable difference between the starting points of

Algorithm 1: `ExtractAsynchronousShapelet`(X, l, α)

Input : $X \in \mathbb{R}^{d \times m}$ - time series, l - shapelet length, α - asynchronicity limit
Output : S - shapelet

1 $S = \mathbf{0}^{d \times l}$; // initialize empty shapelet
2 $j \sim \mathcal{U}(1, m - l - \alpha + 1)$; // sample global starting index
3 **for** $k = 1$ **to** d **do** // for each dimension
4 $\quad j_k \sim \mathcal{U}(j, j + \alpha)$; // sample dimension-specific starting index
5 $\quad S[k] = X[k, j_k : j_k + l - 1]$; // extract shapelet from dimension k
6 **return** S

the shapelet dimensions. Formally, let j_k be the starting position of the k-th dimension of the shapelet. The asynchronicity limit, α, ensures that:

$$\max_{k,k'} |j_k - j_{k'}| \leq \alpha,$$

where k and k' are indices of the dimensions of the multivariate time series.

The asynchronous shapelet extraction process involves several steps, as summarized in Algorithm 1. After initializing the empty shapelet (line 1), a generic starting position j is sampled uniformly from the range $[1, m - l - \alpha + 1]$, i.e., $j \sim \mathcal{U}(1, m - l - \alpha + 1)$ (line 2). This ensures that the sampled starting position falls within the allowable range, taking into account the shapelet and signal lengths as well as the asynchronicity limit. Next, for each dimension k of the multivariate time series, a dimension-specific starting index j_k is sampled uniformly, $j_k \sim \mathcal{U}(j, j + \alpha)$ (lines 3-4). This step ensures that the starting positions of the shapelet dimensions are close to each other, differing by at most α, thereby adhering to the asynchronicity constraint. The index j_k is then used to extract a univariate shapelet of length l from signal k of time series X, which is stored in the multivariate shapelet, S, (line 5). The proposed methodology for shapelet extraction allows each dimension to have a slightly different starting point while maintaining a controlled degree of synchronicity across all dimensions. Setting $\alpha = 0$ results in synchronous shapelets (or MST_I from Section 3), where all dimensions start simultaneously.

Given the unfeasible complexity of a brute-force shapelet extraction approach [33], we randomly extract h shapelets from the TSC dataset. Specifically, this extraction is performed in a supervised manner, selecting $\lfloor \frac{h}{c} \rfloor$ shapelets from time series belonging to each of the c classes, ensuring that each class is represented by some shapelets. This approach maintains good performance even in heavily imbalanced datasets, as demonstrated in Section 5.

4.2 Shapelet Matching

To perform shapelet matching, we compare each extracted asynchronous shapelet to time series instances. Formally, for each dimension $k \in \{1, \ldots, d\}$, we compute

the minimum Euclidean distance between the subsequence $\mathbf{x}_{k,j:j+l-1}$ in X and the corresponding shapelet dimension $\mathbf{s}_k$ over all possible starting positions j:

$$D_k(X, S) = \min_{j \in \{1,\dots,m-l+1\}} \sqrt{\sum_{i=0}^{l-1} (x_{k,j+i} - s_{k,i})^2}.$$

Then, the overall distance $D(X, S)$ is the sum of the distances for each dimension, i.e., $D(X, S) = \sum_{k=1}^{d} D_k(X, S)$. $D(X, S)$ is used as the feature that represents the relationship between shapelets and time series in the Shapelet Transform.

4.3 Shapelet Transform

Once we have extracted our multivariate shapelets and defined a matching criterion, similar to a standard Shapelet Transform, we can use them to convert our time series dataset into a tabular form:

Definition 7 (Multivariate Asynchronous Shapelet Transform). *Given a time series dataset $\mathcal{X}$ and a set $\mathcal{S}$ containing h multivariate asynchronous shapelets, the* Multivariate Asynchronous Shapelet Transform *converts $\mathcal{X} \in \mathbb{R}^{n \times d \times m}$ into a real-valued matrix $T \in \mathbb{R}^{n \times h}$. Each value in T is obtained by taking the sum of the minimum Euclidean distances between each signal and the corresponding shapelet dimension, via a sliding window.*

Since each shapelet is compared to every time series signal using a sliding window, the time complexity of MARS is $O(h \cdot n \cdot c \cdot m^2)$. The resulting tabular dataset T can be used by any classifier, offering interpretable input features.

5 Experiments

In this section, we focus on the Generali case study by comparing MARS against state-of-the-art classifiers and anomaly detection methods. We also test alternatives of MARS, analyzing the effect of different configurations and hyperparameter choices, and provide examples of the shapelet-based explanations that MARS can generate. Furthermore, we propose synthetically unbalanced benchmarks on standard multivariate datasets from the UEA repository[5].

5.1 Case Study - Predicting Car Crashes

The dataset and model made available by Generali are built on telemetry data and are summarized in the following. However, some details can not be disclosed due to company policies. The dataset is composed of a 4-dimensional multivariate time series, where the first three signals are high-frequency recordings of the accelerations on the three car axes and are composed of around 1600 observations

[5] Code is available at https://github.com/bianchimario/MARS.

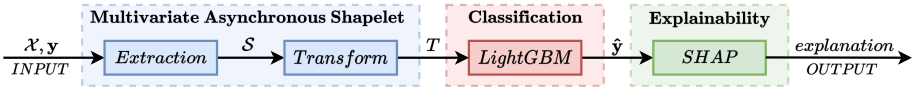

Fig. 2. A simplified schema of the MARS pipeline. Time series and their respective labels $(\mathcal{X}, \mathbf{y})$ are provided as input to MARS (blue), which extracts the shapelets, $\mathcal{S}$, and performs the shapelet transform. The resulting dataset, T, is then used to train a classifier, in our case, LightGBM [14]. The classifier's predictions, $\hat{\mathbf{y}}$, are interpreted using SHAP [19], which offers an explanation in terms of shapelet relevance for classification.

each, while the last one is a low-frequency recording of the speed and is composed of around 40 observations. The task is a binary classification problem, where a model has to distinguish between *No-Crash* (0), and *Crash* (1) instances. Crashes are much rarer; therefore, the problem is heavily imbalanced. In particular, the dataset is composed by a training set containing of $\approx 40,000$ instances with an imbalance of 98.5%/1.5%, a validation set with $\approx 16,000$ (98.5%/1.5%), and a test set with $\approx 100,000$ records, which is even more imbalanced (99.9%/0.1%).

Generali's black-box model is a Convolutional Deep Neural Network, built on several layers that make the model accurate, but non-interpretable. Since this model gives its predictions as continuous variables in the range 0–1, two different prediction thresholds have been set to manage the model sensitivity: a lower threshold, *thr_low*, and a higher threshold, *thr_high*. If the prediction is higher than the threshold, then the predicted label is 1; otherwise, it is 0.

Metrics. Since the dataset is severely imbalanced, the classification task must be evaluated with appropriate metrics, as the metrics typically used for classification problems, such as accuracy, are ineffective for extremely imbalanced scenarios like the one presented by Generali. Thus, following [11], we measure imbalanced classification performance using the Geometric Mean ($\sqrt{sensitivity \times specificity}$), the F1-score, and the Gini coefficient ($2 \times AUC - 1$).

MARS Configuration. A simplified schema of the MARS pipeline for our case study can be viewed in Figure 2. In order to find the best MARS pipeline configuration for Generali's task, we study here how different hyperparameter choices can affect MARS performance on the validation set. First, we study shapelet length l and asynchronicity α. Generali's experts identified only two ranges of shapelet lengths that would be most relevant in a crash prediction scenario, i.e., short shapelets with $8 \leq l \leq 20$ and long shapelets, with $21 \leq l \leq 40$. The length of each shapelet is sampled uniformly at random in those ranges. As for asynchronicity, we test three scenarios, i.e., synchronous shapelets ($\alpha = 0$), shapelets with low asynchronicity ($\alpha = 20$), and totally asynchronous shapelets ($\alpha = m - l = \max$). Further, we also test extracting univariate shapelets from single signals, particularly the x-axis (front-to-back) acceleration, and the car's speed. After the shapelet transform, as shown in Figure 2, we use a LightGBM classifier [14] with default parameters on the resulting dataset, due

Table 1. Hyperparameter analysis for different MARS configurations of shapelet length and asynchronicity. Top three models by metric are presented in bold, the best result by metric is underscored (higher is better).

shapelet type	l	α	*multi*	*f1*	*gm*	*gini*
short, sync	8-20	0	✓	0.11	0.74	0.54
short, low async	8-20	20	✓	0.07	0.63	0.39
short, high async	8-20	max	✓	**0.14**	**<u>0.80</u>**	**<u>0.64</u>**
long, sync	20-40	0	✓	0.09	0.66	0.43
long, low async	20-40	20	✓	0.08	0.70	0.48
long, high async	20-40	max	✓	**<u>0.17</u>**	**0.79**	**0.62**
x-axis only	8-20	✗	✗	0.11	0.72	0.52
speed-axis only	8-20	✗	✗	0.04	0.47	0.22

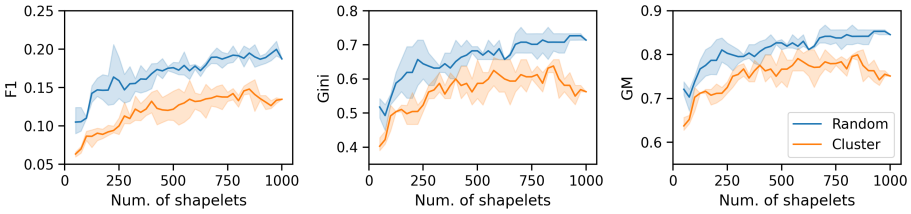

Fig. 3. Average performance (*f1*, *gini*, *gm*) and 0.95 confidence interval for random (blue) and cluster-based (orange) shapelet extraction, when varying the number of shapelets. Higher is better.

to its well-known fast and accurate performance, even on large datasets. Results are reported in Table 1. In general, highly asynchronous shapelets perform better than synchronous ones. Moreover, shapelet length does not seem to impact performance significantly, so from now on, we will extract shapelets in the full 8-40 range, i.e., both short and long subsequences.

Then, we attempt to determine how many shapelets are necessary to achieve good performance. For this test, we use two different kinds of extractions. The first is the original MARS approach presented in Section 4, i.e., a random approach. The second is a variant of MARS in which shapelets are extracted from 25 medoids for each class, i.e., a cluster-based approach. In particular, we rely on the CLARA [13] clustering method, applied after Piecewise Aggregate Approximation (PAA) [15] to reduce the number of points to a more manageable size for the algorithm. Results of three randomly seeded runs are reported in Figure 3. For all metrics, it is clear that the random approach is superior. Furthermore, performance seems to plateau around the 750 shapelets mark, with the variability dropping close to 1000. From these tests, the best MARS configuration for crash prediction is 1000 shapelets of length between 8 and 40, with maximum asynchronicity.

Table 2. Comparison among State-of-the-Art methods. Best result by metric in bold.

	MARS	GEN_L	GEN_H	TSF	ROCKET	ROCKAD	XGB	LGBM
f1	0.19	0.17	**0.31**	0.18	0.28	0.01	0.14	0.12
gini	**0.71**	0.66	0.47	0.64	0.53	0.28	0.58	0.66
gm	**0.84**	0.81	0.69	0.80	0.73	0.56	0.76	0.81
tp	**38**	35	25	34	28	18	31	35
fp	315	330	**84**	283	121	6122	349	514
fn	**15**	18	28	19	25	35	22	18
$time_{tr}$	17 h	-*	-*	3 h	1 h	5 h	5 min	**1 min**
$\overline{time}_{in}$	0.82 s	0.09 s	0.09 s	0.14 s	0.07 s	0.01 s	35 µs	**30 µs**

* Data unavailable due to training on Generali's system.

State-Of-The-Art Comparison. We benchmark MARS against several competitors[6]. As baselines, we use LightGBM (LGBM) [14] and XGBoost (XGB) [7] directly applied to the flattened time series where all signals were concatenated. These approaches ignore the sequential structure of the data. For state-of-the-art time series classifiers, we test ROCKET [8] and Time Series Forest (TSF) [9]. As an anomaly detection algorithm, we benchmark ROCKAD [28]. Finally, we compare with the two thresholded versions of the black-box model provided by Generali, i.e., GEN_L for the low threshold, and GEN_H for the high threshold. All methods are run in parallel and evaluated using their default library parameters[7].

Results are reported in Table 2. MARS has the best overall performance in terms of *gini* and Geometric Mean (*gm*), while it achieves third place in *f1*, following GEN_H and ROCKET. Furthermore, MARS also has the highest number of True Positives (*tp*) and the lowest number of False Negatives (*fn*). A high number of *tp* indicates effective recognition of true *Crashes*, while a low number of *fn* is crucial, as misclassifying a *Crash* as a *No-Crash* could be potentially dangerous for the operator, who might need to assist a customer in real need. The best model for avoiding False Positives (*fp*) remains GEN_H, which is expected given that a high threshold was set for this purpose. Minimizing *fp* is also important to avoid unnecessary calls by operators, which could be perceived by the customer as harassment. TSF consistently performs well, even if it is not in the top three. Interestingly, a tabular approach like LGBM achieves very good *gm* and *gini* scores, although these are still lower than our approach. Given that MARS also uses LGBM as a classifier, this further suggests that our multivariate shapelet representation contains better information than the raw time series data. The runtime comparison shows a downside of MARS, i.e., its slow training time ($time_{tr}$) of 17 hours. However, given that crash reports are not overly common, this is not a significant problem, and the most important aspect is an average fast inference on individual instances ($\overline{time}_{in}$). On average, a multithreaded run of MARS takes only 0.82 ± 0.02 seconds to classify a single instance.

[6] System: Macbook Air, 8-core M1 CPU, 16GB Memory.
[7] We use the following libraries: `sktime`, `lightgbm`, `xgboost`.

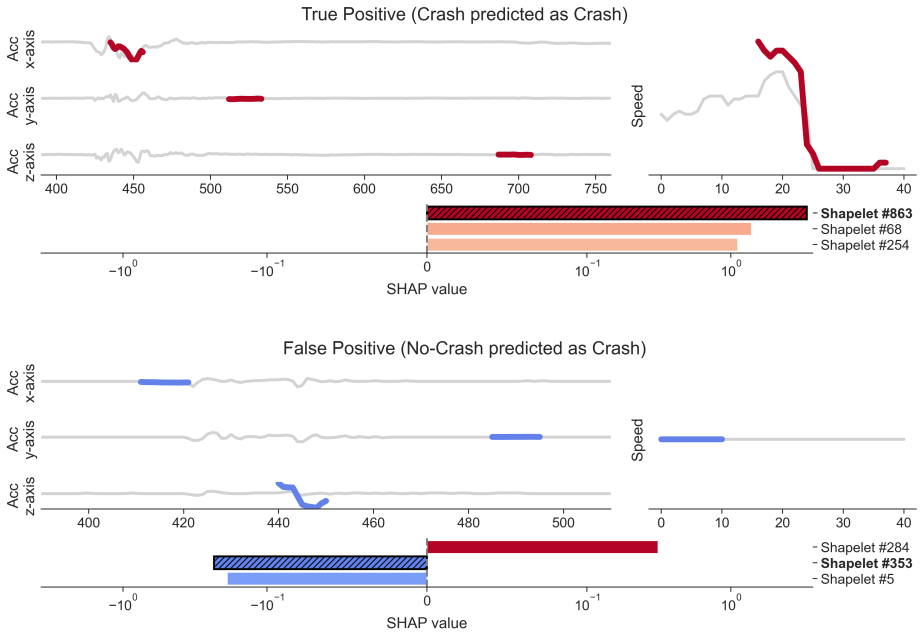

Fig. 4. Two explanations for the MARS prediction for a True Positive (top), and a False Positive (bottom). For each instance, only one of the most important multivariate shapelets is visualized on the top of the time series. The shapelet on the top in depicted in red, indicating that it contributes toward the class *Crash*; on the contrary, the blue shapelet on the bottom indicates a contribution toward *No-Crash*.

Explaining Car Crashes. Here, we show an example of an explanation that can be obtained using the MARS multivariate asynchronous shapelet representation[8]. An interesting feature of the classifier used by MARS, i.e., LGBM, is that it natively produces SHAP values [19], which can be used to visualize the most important shapelets for the classification. In practice, we combine the already human-interpretable shapelets with a score, indicating the contribution of that particular feature in the classifier's prediction.

In Figure 4, we show the explanation for two instances of the test set. At the top is a True Positive instance, a time series correctly classified as a *Crash*. At the bottom is a false positive, a *No-Crash* incorrectly classified by MARS as a *Crash*. The two bar plots at the bottom of each figure depict the importance (SHAP values) of the top-3 most important shapelets for the classification. Positive importance, depicted in red, indicates the shapelet contributes toward the class *Crash*, while negative importance indicates it contributes toward the class *No-Crash*. At the top of each figure are the channels of the classified time series: accelerations on the left and speed on the right. The hatched and bolded shapelet is plotted on the time series in its best alignment. For the True Positive instance,

[8] More explanations are available in the code repository.

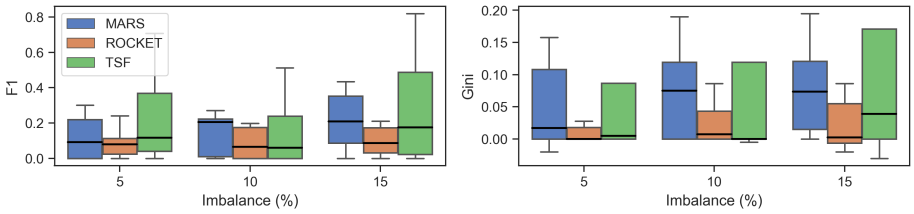

Fig. 5. Boxplots of the performance metrics for different dataset imbalances and different models (higher is better).

the most important shapelet (#863) shows a rapid decrease in speed followed by a stop, a common indicator of a crash in many instances of Generali's dataset. The shapelet on the y and z-axes is mostly flat for the acceleration signals, while there is a jolt highlighted on the x-axis, likely due to a sudden braking action, causing a forward shift detectable primarily along the x-axis. This forward shift indicates sudden deceleration, a common physical response in a crash scenario. The flatness of the y and z axes suggests minor lateral and vertical jolts are not relevant for classification.

For the False Positive instance, it is interesting to see that the second and third most important shapelets push toward the class *No-Crash*. Shapelet #353 shows a zero speed signal and mostly flat acceleration, likely caused by external forces such as being pushed or pulled by another vehicle, which our model misclassifies as a *Crash*. A skilled operator would detect the model mistake by observing the presence of shapelets contributing toward *No-Crash*. More generally, this kind of plot allows the operator to focus on important patterns in the data, providing insights into the shapelets' contribution towards classification.

5.2 UEA Datasets Benchmarking

We experimented with MARS on the 8 smallest multivariate datasets from the UEA repository [2][9], against ROCKET [8] and TSF [9]. Specifically, 4 datasets contain binary classification tasks, and 4 a multiclass problem. In order to understand the effectiveness of MARS on imbalanced time series classification tasks, we synthetically unbalanced the classes of these datasets using random downsampling, such that the minority class is 5, 10 or 15% of the total dataset. For multiclass datasets, we downsample half of the classes. Therefore, half of the labels become the majority and the other half minority classes. For the hyperparameter configuration, we maintained the same settings as in Generali's task, except that the upper bound on the shapelet length was adjusted to half of the signal size (instead of 40), since, for these new datasets, we do not have the domain-specific knowledge of Generali's experts.

[9] Cricket, Epilepsy, EthanolConcentration, FingerMovements, Heartbeat, RacketSports, SelfRegulationSCP1, SelfRegulationSCP2.

Table 3. Average classification performance and standard deviations of MARS and competitor models on UEA datasets for 5, 10, and 15% class imbalances. Higher is better, best models by metric and imbalance in bold.

	f1			*gini*			*gm*		
	5%	10%	15%	5%	10%	15%	5%	10%	15%
MARS	.12±.12	**.22±.28**	.26±.25	.05±.07	**.15±.24**	.12±.17	.05±.15	**.15±.31**	.11±.26
ROCKET	.08±.08	.16±.27	.18±.29	.02±.05	.09±.20	-.03±.50	.07±.14	.10±.28	.13±.30
TSF	**.22±.25**	.20±.30	**.28±.31**	**.10±.19**	.12±.22	**.16±.27**	**.09±.26**	.10±.27	**.14±.29**

Average performance and standard deviation are reported in Table 3, while boxplots are depicted in Figure 5. In general, ROCKET seems to perform worse than both MARS and TSF. Regarding the mean performance, TSF performs slightly better than MARS for the 5 and 15% imbalance, while MARS is better for the 10% imbalance. The boxplots show that the median performance of MARS is almost always better than competitors, even if the interquartile range is wider (and taller) for TSF. This indicates that TSF performs slightly better on average but has a higher variability. Given the many zeros that make it hardly readable, the boxplot for the geometric mean is not included. In summary, MARS can be a valid alternative also outside the Generali case study for imbalanced datasets.

6 Conclusion

In this paper, we proposed MARS, an interpretable-by-design time series classifier utilizing multivariate asynchronous shapelets. These shapelets capture complex patterns across multiple variables that are not perfectly aligned in time, enhancing MARS's ability to handle real-world data. Experimentally, we tested various hyperparameter configurations to optimize MARS, and we evaluated it on the case study provided by Generali and several unbalanced datasets from the UEA repository. Results showed that MARS outperforms competing approaches in Generali's use case and is competitive with state-of-the-art classifiers on standard benchmark datasets, while also providing interpretable predictions.

MARS has some limitations. First, the training runtime is slow. Although this is not a concern for Generali's specific use case, where inference is required only on a small set of instances, optimizing time complexity remains a crucial goal for broader application. The primary computational challenge lies in shapelet matching, which can be improved through various strategies, such as optimizing distance computations, employing approximations, or leveraging early classification techniques. Second, the asynchronicity limit is only applied in the extraction phase, not the matching phase. For future work, we plan to test different shapelet extraction methods and introduce an asynchronicity parameter also in the matching phase to improve control over shapelet alignment. Finally, we aim to enhance the explanations, for example, by using decision rules and trees, to make them global and more expressive, and therefore easier for users to understand.

Acknowledgments. This work has been partially supported by the EU H2020 programme under the funding schemes ERC-2018-ADG G.A. 834756 "XAI: Science and technology for the eXplanation of AI decision making", "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics", by the European Commission under the NextGeneration EU programme – National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) – Project: "SoBigData.it – Strengthening the Italian RI for Social Mining and Big Data Analytics" – Prot. IR0000013 – Avviso n. 3264 del 28/12/2021, and M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", and by Fondo Italiano per la Scienza FIS00001966 MIMOSA.

References

1. Ba, Y., Zhang, W., Wang, Q., Zhou, R., Ren, C.: Crash prediction with behavioral and physiological features for advanced vehicle collision avoidance system. Transportation Research Part C: Emerging Technologies **74**, 22–33 (2017)
2. Bagnall, A., Dau, H.A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., Keogh, E.: The uea multivariate time series classification archive, 2018. arXiv:1811.00075 (2018)
3. Bai, J., Goldsmith, J., Caffo, B., Glass, T.A., Crainiceanu, C.M.: Movelets: A dictionary of movement. Electronic journal of statistics **6**, 559 (2012)
4. Blázquez-García, A., Conde, A., Mori, U., Lozano, J.A.: A review on outlier/anomaly detection in time series data (Feb 2020), arXiv:2002.04236
5. Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., Rinzivillo, S.: Benchmarking and survey of explanation methods for black box models. Data Mining and Knowledge Discovery **37**(5), 1719–1778 (2023)
6. Bostrom, A., Bagnall, A.: A Shapelet Transform for Multivariate Time Series Classification (Dec 2017), arXiv:1712.06428 [cs]
7. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al.: Xgboost: extreme gradient boosting. R package version 0.4-2 **1**(4), 1–4 (2015)
8. Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Mining and Knowledge Discovery **34**(5), 1454–1495 (2020)
9. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Information Sciences **239**, 142–153 (2013)
10. Ghalwash, M., Obradovic, Z.: Early classification of multivariate temporal observations by extraction of interpretable shapelets. BMC Bioinform. **13**, 195 (2012)
11. Japkowicz, N.: Assessment metrics for imbalanced learning. Imbalanced learning: Foundations, algorithms, and applications pp. 187–206 (2013)
12. Jiang, F., Yuen, K.K.R., Lee, E.W.M.: A long short-term memory-based framework for crash detection on freeways with traffic data of different temporal resolutions. Accident Analysis & Prevention **141**, 105520 (2020)
13. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. John Wiley & Sons (2009)
14. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. NIPS **30** (2017)
15. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: ACM SIGKDD. pp. 285–289 (2000)

16. Landi, C., Spinnato, F., Guidotti, R., Monreale, A., Nanni, M.: Geolet: An interpretable model for trajectory classification. In: International Symposium on Intelligent Data Analysis. pp. 236–248. Springer (2023)
17. Lines, J., Davis, L.M., Hills, J., Bagnall, A.: A shapelet transform for time series classification. In: ACM SIGKDD. pp. 289–297 (2012)
18. Lord, D., Mannering, F.: The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives. Transportation research part A: policy and practice **44**(5), 291–305 (2010)
19. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
20. Mannering, F.L., Bhat, C.R.: Analytic methods in accident research: Methodological frontier and future directions. Analytic methods in accident research **1**, 1–22 (2014)
21. Medico, R., Ruyssinck, J., Deschrijver, D., Dhaene, T.: Learning multivariate shapelets with multi-layer neural networks for interpretable time-series classification. Advances in Data Analysis and Classification **15**(4), 911–936 (2021)
22. Nanni, M., Guidotti, R., Bonavita, A., Alamdari, O.I.: City indicators for geographical transfer learning: an application to crash prediction. GeoInformatica **26**(4), 581–612 (2022)
23. Salim, F.D., Loke, S.W., Rakotonirainy, A., Srinivasan, B., Krishnaswamy, S.: Collision pattern modeling and real-time collision detection at road intersections. In: ITSC 2007. pp. 161–166. IEEE (2007)
24. Saravanarajan, V.S., Chen, R.C., Dewi, C., Chen, L.S., Ganesan, L.: Car crash detection using ensemble deep learning. Multim. Tools Appl. (Jun 2023)
25. Spinnato, F., Guidotti, R., Monreale, A., Nanni, M., Pedreschi, D., Giannotti, F.: Understanding any time series classifier with a subsequence-based explainer. ACM Transactions on Knowledge Discovery from Data **18**(2), 1–34 (2023)
26. Spinnato, F., Guidotti, R., Nanni, M., Maccagnola, D., Paciello, G., Farina, A.B.: Explaining crash predictions on multivariate time series data. In: International Conference on Discovery Science. pp. 556–566. Springer (2022)
27. Theissler, A., Spinnato, F., Schlegel, U., Guidotti, R.: Explainable ai for time series classification: a review, taxonomy and research directions. IEEE Access **10**, 100700–100724 (2022)
28. Theissler, A., Wengert, M., Gerschner, F.: Rockad: Transferring rocket to whole time series anomaly detection. In: IDA 2023. pp. 419–432. Springer (2023)
29. Tritscher, J., Krause, A., Hotho, A.: Feature relevance xai in anomaly detection: Reviewing approaches and challenges. Frontiers Artif. Intell. **6**, 1099521 (2023)
30. Wang, A., Zhang, A., Chan, E.H., Shi, W., Zhou, X., Liu, Z.: A review of human mobility research based on big data and its implication for smart city development. ISPRS International Journal of Geo-Information **10**(1), 13 (2020)
31. Wang, J., Xu, W., Gong, Y.: Real-time driving danger-level prediction. Engineering Applications of Artificial Intelligence **23**(8), 1247–1254 (2010)
32. Wang, Y., Xu, W., Zhang, Y., Qin, Y., Zhang, W., Wu, X.: Machine learning methods for driving risk prediction. In: EM-GIS. pp. 1–6 (2017)
33. Ye, L., Keogh, E.: Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Min. Knowl. Discov **22**, 149–182 (2011)
34. Zantalis, F., Koulouras, G., Karabetsos, S., Kandris, D.: A review of machine learning and iot in smart transportation. Future Internet **11**(4), 94 (2019)
35. Zhu, M., Yang, H.F., Liu, C., Pu, Z., Wang, Y.: Real-time crash identification using connected electric vehicle operation data. Accid. Anal. Prev **173**, 106708 (2022)
36. Ziebinski, A., Cupek, R., Grzechca, D., Chruszczyk, L.: Review of advanced driver assistance systems (adas). In: AIP Conf. Proc. vol. 1906. AIP Publishing (2017)