IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 January 2024, accepted 12 February 2024, date of publication 19 February 2024, date of current version 26 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3366994

RESEARCH ARTICLE

Translating Image XAI to Multivariate Time Series

LORENZO TRONCHIN[1], (Student Member, IEEE), ERMANNO CORDELLI[1], LORENZO RICCIARDI CELSI[2], (Senior Member, IEEE), DANIELE MACCAGNOLA[3], MASSIMO NATALE[3], PAOLO SODA[1,4], (Member, IEEE), AND ROSA SICILIA[1], (Member, IEEE)

[1]Unit of Computer Systems and Bioinformatics, Department of Engineering, University Campus Bio-Medico of Rome, 00128 Rome, Italy
[2]ELIS Innovation Hub, 00159 Rome, Italy
[3]Advanced Analytics, Assicurazioni Generali Italia, 20145 Milan, Italy
[4]Department of Diagnostics and Intervention, Radiation Physics, Biomedical Engineering, Umeå University, 901 87 Umeå, Sweden

Corresponding author: Lorenzo Tronchin (l.tronchin@unicampus.it)

This work was supported in part by the Piano Nazionale di Ripresa e Resilienza (PNRR) Ministero dell'Universitá e della Ricerca (MUR) Project, Rome Technopole (National Research Programme (NRP) Mission 4 Component 2 Investment 1.5, funded by the European Union–NextGenerationEU), under Grant PE0000013-FAIR, Grant ECS 0000024, and Grant CUP C83C22000510001; and in part by the ELIS Innovation Hub within a Joint Research Project with Assicurazioni Generali Italia.

ABSTRACT As Artificial Intelligence (AI) is becoming part of our daily lives, the need to understand and trust its decisions is becoming a pressing issue. EXplainable AI (XAI) aims at answering this demand, providing tools to get insights into the models' behaviour and reasoning. Following this trend, our research paper explores the explainability of a deployed multimodal architecture applied to a real-world dataset of multivariate time series. The study aims to enhance the trustworthiness of an AI agent responsible for crash detection in an insurance company's automatic assistance service. By introducing an XAI layer, we provide insights into the AI agent's decision-making process, enabling the optimization of emergency medical services allocation. The dataset consists of real-world telematics data collected from vehicles equipped with black box technology. The challenge lies in explaining the complex interactions within the multivariate time series data to accurately understand the forces applied to vehicles during accidents. To this end, we adapt to this context two state-of-the-art XAI model-specific approaches, originally designed for images. We offer a qualitative and a quantitative evaluation, also comparing with a well-known agnostic method, and further validating our findings on an external dataset. The results show that Integrated Gradients, among the methodologies examined, is the most effective approach. Its ability to handle the complexity of the data provides the most comprehensive and insightful explanations for the considered use case. The findings emphasize the potential of XAI to enhance the trustworthiness of AI systems and optimize emergency response in the insurance industry. Code is available at https://github.com/ltronchin/translating-xai-mts.git.

INDEX TERMS Explainability, evaluation, multivariate time series, car crash detection.

I. INTRODUCTION

Artificial Intelligence (AI) has become an integral part of our daily lives, serving as a precious tool to assist the human component in numerous tasks, mainly when data is generated at an ever-increasing pace. A case in point is the processing of Multivariate Time Series (MTS) data [1], [2], available in huge amounts due to the ubiquity of sensors and the advances in the Internet of Things (IoT) technologies [3]. Univariate Time Series (UTS) can be regarded as sequences of data points that are ordered according to time and divided into uniform time intervals, while an MTS is a collection of different UTS (also named as attributes) [4]. AI and deep learning techniques have achieved state-of-the-art performance in their classification [5], [6]. However, there are still issues with trusting these decisions due to the intrinsic lack of transparency of the most used AI techniques, artefacts hidden in the training data, and possible biases inherited from human prejudices.

Besides the recent trend in the literature to realise intrinsically interpretable and transparent AI models [7], there exist several scenarios where the models already

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

deployed are "black boxes," so that their reasoning is not humanly interpretable by design [6]. In particular, EXplainable Artificial Intelligence (XAI) has been attracting the interest of the scientific community [8], [9], [10] since the complex nature of AI models prevents the end user from understanding and validating the very decision process performed by the AI model itself. In this respect, XAI is aimed at providing insights into the behaviour of such systems, thus allowing, through algorithm fairness and through the identification of any potential bias occurring within the training data, complex AI models to become more and more transparent and understandable to humans [10]. Hence, research efforts are directed not only towards the challenge of developing AI models that are interpretable and explainable by design but also towards answering the even more pressing demand to explain deployed models that are black boxes for humans.

The explainability challenge is even more pronounced when dealing with time series analysis, which is complex by nature, especially for MTS when multiple variables are involved [11]. This complexity can make it difficult to identify patterns and relationships in the data, and consequently to provide insights that are humanly understandable. Although there is a pressing need to offer explanations for time series data, current research predominantly focuses on UTS rather than the more complex MTS and neglects real-world scenarios where both datasets and deployed models are available for analysis.

In this work, we face the challenge of explaining a real-world black box already deployed to perform the task of car crash detection on telematics data from vehicles. The classifier is a Convolutional Neural Network (CNN) trained to recognise if a crash event occurred or not using an MTS given by a car acceleration signal that, in turn, collects three UTS, which are its spatial components (x, y and z). The development of an XAI approach for MTS-based AI models is rather challenging due to the peculiarity that characterises this type of data: the points of each UTS are connected with the other sequences via the time dimension, thus implying complex non-linear temporal dependencies between the attributes. With the aim of considering the temporal and spatial relationships between each dimension of an MTS at the same time, we regard each of the samples as a 2D image such that each pixel embeds visual abstract features, mapping both temporal and spatial relationships between UTS. In this way, we manage to gain the advantage that follows from analysing the MTS as a whole, exploiting the rich literature available in terms of explainability relative to images by choosing three approaches that belong to different XAI families, namely Grad-CAM, Integrated Gradients (IG), and LIME.

A further issue we tackle in this work is evaluating the provided explanation for the MTS, a topic that is only in its infancy in the current literature [12], [13]. Indeed, different XAI methods produce explanations that might not be the same or equally interpretable [12]. So a pressing need is to quantify the quality of the explanations to evaluate and compare them. Furthermore, it is worth noting that previous studies on explainable AI have primarily focused on image data or single-variable time series, whereas this research explains a multimodal architecture deployed for the insurance industry by adapting XAI algorithms originally designed for images.

This work also significantly extends our previous conference paper [11] for three reasons:

- First, we present an extensive and larger set of experiments providing valuable performance comparison among three XAI methods, considering three possible alternative perturbations: this allows us to experimentally validate, in a true industrial use case, the hypothesis speculated in [11] but not yet quantitatively assessed, that is, that XAI can be used to explain MTSs.
- Second, we tackle the IG baseline problem, an open issue when using this approach in MTS. Furthermore, also in this case, we offer an exhaustive experimental analysis.
- Third, we include a generalisation analysis that applies our approach on a different pair of black box and dataset to assess the ability to explain MTSs in a different context.

In summary, we hereby introduce three main contributions: (i) we explain a real-world architecture deployed to perform crash event detection from vehicles' telematics data; (ii) we translate two state-of-the-art XAI methods from the image domain to the MTS domain, presenting how we customise and employ them to deal with a black box architecture working on multivariate data; (iii) we evaluate the explanations provided by the two methods and compare them with those provided by another state-of-the-art approach to test their effectiveness.

The manuscript is structured as follows: the next section analyses the XAI literature both for UTS and MTS, and section II-A describes our use case, including both the dataset and the AI black box to explain. Section III introduces the explainability methods, their methodological adaptation to the specific MTS field and the evaluation framework. Section IV discusses the experimental settings and the results attained, whereas section V presents the external validation. Section VI provides concluding remarks.

II. RELATED WORK

XAI aims at building AI methods that are explainable by design or rather at developing independent methods to explain existing architectures, a scenario particularly relevant when dealing with deployed black box models. The literature distinguishes the XAI approaches according to different aspects [8], e.g., the problem they try to tackle or the type of model they are applicable to. Far from proposing a novel taxonomy, we hereby present the work on XAI for UTS and MTS, considering as the main categorising aspect the distinction between agnostic explanators, i.e., comprehensible predictors that are not tied to a particular type of black box [8], and model specific explanators, i.e., methods that are tied to a specific architecture type to derive the explanation. Most of the XAI approaches have been proposed principally for problems that deal either with images or with text and tabular data. So far, only limited effort has been directed towards XAI on time series and in the following, we will focus on approaches to explain MTS-based models, with particular attention to model-specific methods designed for CNNs, as this is the type of deployed black box studied in this work [3], [13], [14], [15], [16], [17], [18], [19], [20].

Among the agnostic approaches, we account for the work by Gee et al. [14], which presents a method to learn the prototypes, i.e., representative data examples encountered during model training that describe influential data regions. It exploits a lower-dimension latent representation that is learnt by relying on an encoder network. The latent representation is passed to a feed-forward prototype network to provide relevant insights about the most important features that are employed to perform the classification task. The authors evaluate the proposed framework with respect to three classification tasks using UTS: detecting bradycardia out of electrocardiograms, detecting apnea out of respiration and detecting spoken digits out of audio waveforms. Thus, they show that the prototypes learn features on two-dimensional time-series data and eventually produce explainable insights during the above-mentioned classification tasks. Ates et al. [16] introduce an agnostic explainability technique providing counterfactual explanations for individual predictions. The counterfactual explanations are artificial samples that have to be as similar as possible to the instance that needs to be explained while obtaining a different classification label from the model. This framework is evaluated on classifiers working on 4 different MTS datasets, namely three datasets relative to high-performance computing system telemetry and one motion dataset. To the best of the authors' knowledge, the proposed approach outperforms state-of-the-art explainability methods, exhibiting satisfactory evaluation metrics, especially in terms of faithfulness and robustness. Other methods that regard agnostic time series explainability are based on shapelets, i.e., time series sub-sequences that are maximally representative of a class distribution [21]. Karlsson et al. [15] exploit shapelets to formulate the problem of counterfactual explanations in terms of interpretable time series tweaking: such a problem requires the identification of the minimum number of changes that have to be applied to a time series in order to switch the decision of the classifier. In this work, explainability is evaluated starting from time series shapelets derived by a random shapelet forest classifier [22]. The authors test the proposed algorithm on two UTS use cases, finding that the proposed method is both computationally efficient and valuable. Further to these agnostic explanation approaches designed for time series, Saluja et al. [20] use different model agnostic explainability methods, i.e. Local Interpretable Model agnostic Explanations (LIME) [23], and Shapley Additive explanations [24] to assess explainability in an MTS forecasting task of a company sales activities, analysing the explanations resulting from a human evaluation study.

With respect to the model-specific solutions, their main characteristic is the strong tie with the black box model nature. A popular recent research direction for time series classification exploits CNNs on multivariate data [6], which achieved state-of-the-art performance in terms of classification accuracy using MTS data [25], [26], [27]. Both [3] and [17] employ as specific explanator for their CNNs the Gradient-weighted Class Activation Mapping (Grad-CAM), applied in different configurations depending on the designed CNN. Assaf et al. [3] use a two-stage CNN performing a sequence of 2D and 1D convolutions to harvest both spatial and temporal features in an MTS prediction task. Grad-CAM is applied at two different levels of the CNN: (i) in the first stage, with respect to the output of the last 2D convolutional layer for explaining the feature importance, i.e., spatial inference; (ii) in the second stage, with respect to the feature maps of the last 1D convolutional layer to obtain the timestamp importance, i.e. temporal inference. The authors test the proposed approach on two use cases, the former relative to photovoltaic energy forecasting and the latter relative to server outage prediction, finding that the proposed framework allows to visualise the attention of the network over the time dimension and features. Fauvel et al. [17] proposes a different strategy to apply Grad-CAM: they introduce a CNN with parallel branches, one branch using 2D convolutional filters and the other using 1D filters, extracting directly from the input MTS samples both spatial and temporal features. Grad-CAM is then applied to the last convolutional layers in both branches. The explainability is tested on a synthetic dataset of MTS, comparing this approach against the one in [3]. They observe that even if both [3] and [17] correctly identify the discriminative time window, reference [17] provides more precise attribution maps and thus a more informative explanation. Inspired by the approach of Network Dissection, namely a method showing the spatial locations that each unit in the CNN is looking at [28], Siddiqui et al. [18], and Cho et al. [19] rely on the neuron and filter activation of a CNN with the aim of identifying the contribution of the raw input data when performing MTS classification. The former [18] creates a dummy dataset for time series anomaly detection with three features, that is, pressure, temperature and torque. The latter [19] interprets deep temporal representations using two open-source MTS datasets. A final interesting contribution is by Schegel et al. [13], which evaluates 5 explainability methods, considering both agnostic and model-specific approaches previously used for image and text-domain. They apply the selected algorithms on UTS data from 9 public datasets. To evaluate the relevance of the obtained explanations, they perform a perturbation-based analysis, modifying the values of the relevant time stamps and computing the change in model performance. The authors use both a CNN and a Recurrent Neural Network (RNN) as baseline models for performing binary classification on each of the datasets. Their findings underline that the heatmap obtained from each of the XAI methods is hard to interpret, raising the need for further exploration of CNN and RNN explainability.

The literature analysis reported so far highlights some limitations regarding the study of XAI for MTS. First, the majority of the efforts are directed towards UTS, which does not consider the more demanding scenario of MTS. Second, only a few multivariate public datasets are available [6], making it even more challenging to provide a fair comparison between available XAI methods. Third, to the best of our knowledge, the literature has not considered real-world scenarios where not only the dataset is available but also the model to be explained is already deployed. Given these limitations, we decided to consider the extensive literature about XAI for image data as a starting point to enrich the contributions in the field of XAI for MTS. In detail, we study how to adapt, employ and evaluate their applicability to MTS in a real-world scenario.

A. USE CASE

Insurance companies have recently started using artificial intelligence to gain valuable insights about drivers and vehicle security, analysing data retrieved from telematics smart boxes mounted on board [29]. Our work considers a specific use case in this domain where an AI model, also referred to as an "AI agent," is employed to optimise the assistance service of the insurance company Generali Italia S.p.A. Whenever the vehicle is involved in an accident, the AI agent processes the deceleration burst collected by the black box on board and automatically triggers a call to an insurance operator who, in turn, dials the driver to check his/her conditions and the general situation. If necessary, this call is forwarded to the Emergency Medical Services, to offer any necessary pre-hospital treatment, and/or to the tow truck in charge of removing the vehicle itself. In this scenario, the insurance operator receiving a call should be able to understand and trust the decision of the AI model to properly take the subsequent actions, without uselessly bothering the driver, starting any intervention of third parties or, rather, potentially missing a dangerous event. This is not only an advantage for the insurer but also for the driver to whom the insurance operator can explain and detail the reasons behind the call and potentially provide interpretations about further relevant elements, such as his/her driving style.

Before delving into the details of the XAI methods employed to tackle this issue, the next subsections illustrate the dataset and AI agent provided by Generali S.p.A.

B. DATASET

This paper uses as a relevant dataset a collection of telematics data that result from being retrieved from the vehicle's black box, which, in turn, is equipped with a global positioning system and an accelerometer in charge of returning information about any possible crashes experienced by the car. The dataset is composed of 81173 samples. Each sample contains a multivariate acceleration signal recorded along the x, y, and z directions and a univariate speed magnitude signal. The former consists of a 2490-timestamp sample per axis. It is extracted from raw accelerometric data, considering a total temporal window of 15 seconds divided in the following way: (i) 9 seconds sampled at 10Hz; (ii) 6 seconds sampled at 400Hz. The latter, instead, consists of a 41-timestamp sample acquired over a 41 seconds time window at 1Hz. Each sample is annotated as a *crash* or *non-crash* event, and each crash/non-crash event comes from a different car. These two labels are assumed hereinafter to correspond to the positive and negative classes, respectively. With the aim of building the AI agent, the dataset was randomly split into training (50%), validation (25%) and test (25%) sets, as shown in TABLE 1. Furthermore, TABLE 1 shows how training, validation, and test sets are distributed among crash and non-crash samples, revealing that the dataset is skewed with more examples belonging to the non-crash class. The strategy to tackle this issue is presented at the end of the next subsection.

C. THE AI AGENT

Note that the problem can be formulated as a multimodal task since the two types of time series represent two distinct quantities sampled at different rates. Hence, the AI agent described in the next section is a multimodal CNN that processes both the acceleration MTS and the speed magnitude UTS. However, as demonstrated in section IV-A, the acceleration MTS retains the significant information exploited by the AI agent to detect a crash. This is why we will neglect the speed magnitude UTS from the XAI analysis, focusing only on the multivariate input.

The insurance company chose to employ a CNN to detect any car crash events from the telematics data described in section II-B, and FIGURE 1 presents the related multimodal architecture using both acceleration and speed signals. It is worth noting that the task at hand can also be tackled with anomaly detection techniques. However, the exploration of different classification approaches as well as their comparison, is out of the scope of this work, as it will be also discussed at the end of Section IV-C. In FIGURE 1 on the left side of each block, convolutional layers are denoted with Conv1D followed by the size of the kernel $K \times C$, with K and C representing the kernel size and the number of channels respectively, and by the number of filters F. On the right side of each block, the figure reports the size and the number of output images for each layer.

The chosen convolutional-based architecture exploits an MTS-to-image encoding approach for acceleration, with the aim to catch both the relationships between the attributes and similar patterns present at different time stamps.

TABLE 1. A-priori class distribution between crash and non-crash classes per set (training, validation, and test).

Class	Training	Validation	Test
Crash	4.1%	4.3%	6.3%
Non-crash	95.9%	95.7%	93.7%

Due to the fact that the acceleration and speed recorded in the dataset exhibit different sampling rates and different time windows, the designers relied upon a joint fusion strategy, which combines the feature representations learned by the intermediate layers of two neural networks (NNs) as input to a final model. Such an approach allows us to employ the loss, as a result of being back-propagated to the two NNs, for creating a better representation of the two modalities. In more detail, let us now discuss this multimodal architecture: the first NN works on the image-like MTS relative to acceleration (left branch in FIGURE 1), it accepts 2490 × 3 input tensors and foresees two consecutive blocks, each containing a stack of one 1D convolutional layer with ReLU activation and max-pooling, in order to reduce the size of the MTS itself. The filter being used in each convolutional layer has $5 \times C$ size, with C accounting for the number of channels from the previous layer, i.e., the width of the MTS as a result of each convolution operation. The second NN, instead, processes the incoming signal related to speed (right branch in FIGURE 1); it accepts 41 × 1 input tensors and it consists of a single 100-neuron dense layer with ReLU activation, due to the necessity to cope with the low dimensionality intrinsic to the UTS modality.

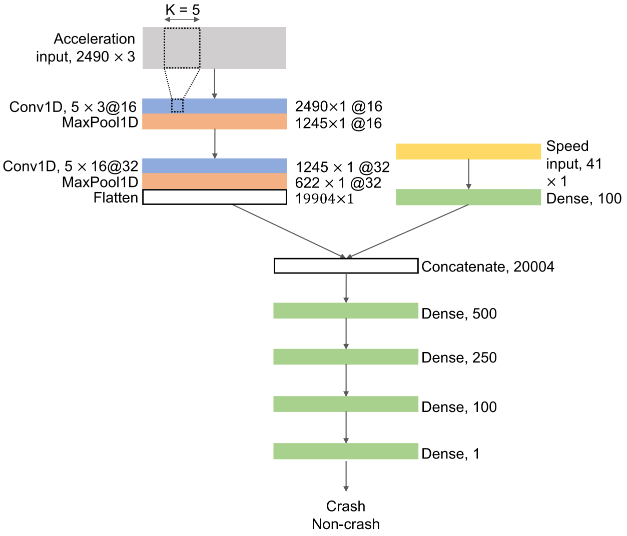


FIGURE 1. AI agent architecture for car crash detection.

The features that are extracted from the two modalities are then concatenated and fed to a multi-layer perceptron model in charge of performing the classification task. The multi-layer perceptron is composed of three dense layers with 500, 250, and 100 neurons, respectively, and ReLU activation. Eventually, the final layer includes one dense neuron layer with a sigmoid activation function, and returns the posterior probability of a crash event. To update the weights, the designers relied on the Adam algorithm with a 0.001 learning rate. The loss was measured using binary cross-entropy.

With the aim of maximising the recall, since one more call to assist the customer (i.e., false positive) is better than leaving a customer in need of assistance in a crash event (i.e., false negative), the threshold to make binary the continuous output of the network was set to 0.506, using the precision-recall curve measured on the validation set. The experiments on the test set get the values of recall and precision equal to 70% and 63%, respectively.

III. METHODS

FIGURE 2 shows an overview of the proposed approach for MTS to explain the CNN decisions and also to evaluate the explanation quality. FIGURE 2 can be read by following the arrow in bold at the bottom of the figure itself. More in detail, the proposed approach consists of four steps: namely, training, testing, explanation extraction and explanation assessment.

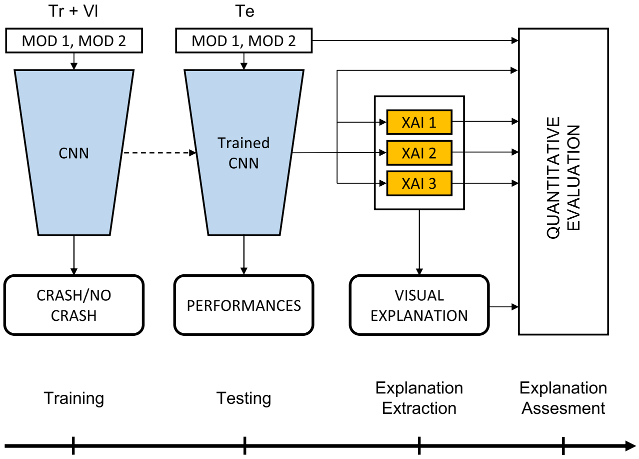


FIGURE 2. An overview of our approach.

Training, as already described, relies on the training and validation sets ('Tr+Vl' in FIGURE 2) related to both the speed signal (denoted with 'MOD 1') and the acceleration signal (denoted with 'MOD 2'), to train a CNN classifier aimed at solving the task of telling car crash events from non-crash ones. Validation is carried out in order to determine the best model that is ultimately adopted for testing as the so-called 'Trained CNN' (refer to the dashed arrow in the figure).

At the subsequent step (i.e., testing), the input speed and acceleration signals arranged in a suitable test set ('Te' in FIGURE 2) as described in Section II-B are fed to the trained CNN in order to evaluate the test performances. The resulting black box (or trained CNN) is, in turn, fed to the three explainability methods – that is, Grad-CAM denoted with 'XAI 1', IG denoted with 'XAI 2' and LIME denoted with 'XAI 3' – for the purpose of extracting visual explanations, one for each XAI method, capable of explaining what the black box has learnt so far.

Eventually, for the purpose of explanation assessment, a quantitative evaluation of the three XAI methods is run, based on the following inputs: the three XAI methods themselves, the test set and the trained CNN.

Below we will provide a more detailed discussion of the most relevant aspects of the proposed methodology: section III-A presents the MTS-to-image encoding proposed to leverage XAI techniques typically applied to image analysis; section III-B illustrates the three different XAI techniques employed, both agnostic and model specific, focusing on the methodological adaptations needed to fit the task at hand; finally, section III-C describes the evaluation strategy that is considered for the resulting explanation.

A. MTS-TO-IMAGE ENCODING

The strategy of transforming a time series into a bi-dimensional object that can be treated as an image is quite common in the time series classification research landscape [30]. This is the approach also employed by the AI agent provider to treat the multivariate acceleration input. We hereby propose a formalisation of such modelling that we take as a reference when retrieving the corresponding explanations.

We remap an MTS signal to a grayscale 2D image as follows. Let us denote by $M = [m_1, \ldots, m_Q]^T$ an MTS with N timestamps and Q attributes, such that each attribute $m_q = [m_{1q}, \ldots, m_{Nq}]$ is an UTS. It is straightforward to remap such an object into a $Q \times N$ image I_{2D-MTS} that has a number of rows equal to the number of attributes and a number of columns equal to the number of timestamps.

FIGURE 3 shows this procedure applied to the accelerations MTS of the dataset at hand, generating an image denoted as a_{2D-MTS}, with a size equal to 2490 × 3. After the MTS acceleration signal, denoted with 'MOD 2' in FIGURE 3, is acquired from the telemetry of vehicles, each UTS is combined to create a 2D image where each row indicates a UTS and each column the timestamp. Each pixel is also rescaled into the interval $[-1, 1]$ to cope with the negative peak values observed from the samples.

It is worth reminding that this encoding applies only to the acceleration MTS input, whereas we do not provide any specific modelling of the speed magnitude UTS. Indeed, as anticipated in section II-B and further proven in section IV-A, the speed UTS is not relevant with respect to the XAI aims, since it actually does not retain significant information for the classification task.

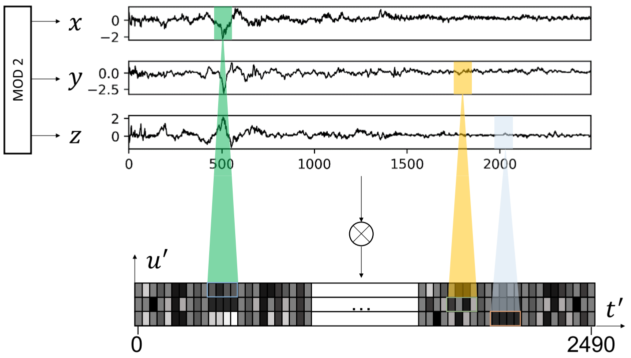


FIGURE 3. Summary of the procedure to build the image-like MTS.

B. XAI FOR MTS

As aforementioned, in this work we address the challenge of explaining a real-world black box. By leveraging the MTS-to-image encoding presented in the previous section as well as the convolutional nature of the AI agent, we now introduce three main XAI methods:

(i) a *model-specific solution*, namely Grad-CAM [31], which is intentionally designed for convolutional architectures, considering CNN activation on the specific input sample;

(ii) a *model-inspection approach*, namely IG [32], a method, designed for deep networks, that gains its explanation as a result of examining the internal model behaviour when varying the input sample;

(iii) for the sake of completeness, we include in this section an overview of LIME [23], an *agnostic solution* which generalises by definition to any model and outputs a comprehensible local predictor and is here employed for comparison purposes in the eventual evaluation of the explanators.

All the above-mentioned methods were originally designed for image explanation purposes and provide a Saliency Map (SM) as output. SMs are an efficient way of pointing out what causes a certain outcome, mainly when images are being treated. Indeed, a SM can be regarded as a "mask" that visually sheds light on the critical aspects of the analysed data, i.e., the very timestamps that are relevant to the crash-detection task. In other words, a SM can be considered as a visual representation of feature importance. This ultimately allows the end user to visually inspect the explanation.

We now present how we customise and employ Grad-CAM, IG and LIME to deal with a black box architecture working on multivariate telematic data.

1) GRAD-CAM

Grad-CAM [31] is a suitable method for explaining the decision made by the output layers of CNNs. Indeed, it relies on the gradient information that flows into the last convolutional layer, with the aim of understanding the importance of each neuron in the decision of interest. The reason why the last convolution layer is used is that this approach is able to capture the higher-level semantics extracted by the convolutional block as well as it retains any spatial information which is generally lost in fully-connected layers. This way, we can expect the last convolutional layers to eventually yield the best compromise between high-level semantics and detailed spatial information [31]. Grad-CAM returns an attribution map, that is, a SM with the same size as the input data, such that the colour intensity is correlated with the input features and, as a consequence, the activated areas highlight which timestamps and which attributes the CNN looks into within the synthetic image when it comes to making its predictions.

With the aim of obtaining the class-discriminative attribution map denoted with $SM^c_{\text{Grad-CAM}}$ for a class c, Grad-CAM computes, first, the gradient information of the prediction score for class c, y^c, with respect to feature map activations A^k of the last convolutional layer, i.e., $\frac{\partial y^c}{\partial A^k}$ with $k \in [1, \ldots, F]$ and F identifying each feature map and the total number of feature maps in the considered convolutional layer, respectively. These gradients that flow back are global-average-pooled over the image-like MTS signal dimensions with the aim of determining a weight w_k^c that represents the importance of each feature map k for the target class c:

$$w_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{i,j}^k}, \tag{1}$$

with Z denoting the number of pixels in the feature map k, and the pair (i,j) denoting each pixel. These weights are then exploited to compute a weighted combination of all the feature maps for the class under investigation; then, the use of a rectified linear activation function or Rectified Linear Unit (ReLU) [33] keeps only the features that exert a positive influence on the class of interest, that is, any pixels whose intensity should grow to increase y^c [31]. Thus, Grad-CAM returns a linear combination between weight values and feature maps and relies on a ReLU in order to keep only the positive attribution to the prediction, which results in a coarse heatmap of the same size as the filter (or feature map) appearing in the last convolutional layer:

$$SM^c_{\text{Grad-CAM}} = ReLU\left(\sum_{k=1}^{F} w_k^c A^k\right). \tag{2}$$

where F is the total number of feature maps in the last convolutional layer.

Although this approach can work on our data, it does not explicitly exploit the information given by the relationship between the attributes. This way, we modify the baseline Grad-CAM method by introducing a recombination method to provide a finer grade for the explanations of each MTS attribute. In detail, we face the problem of matching the localization map $SM^c_{\text{Grad-CAM}}$ with the input exploiting the last *Conv1D* layer of the image-like MTS branch (FIGURE 1). As the *Conv1D* layer comprises $k = 32$ feature maps, each of dimension 1245 × 1, we need to map the final 1245 × 1 heatmap ($SM^c_{\text{Grad-CAM}}$) onto the 2490 × 3 input signal. To this aim, we design both an *ablation-based perturbation* on the multivariate input a_{2D-MTS} and a recombination approach, thus finally adapting Grad-CAM to the multivariate nature of the telematics data received in input. The proposed approach works as follows (FIGURE 4).

1) According to classical Grad-CAM (denoted with 'XAI 1' in FIGURE 4), we calculate a 2490×1 heatmap SM_I (represented by the corresponding block in the left handside of FIGURE 4) and we compute the probability of crash (denoted with p_I), with respect to the original acceleration signal (denoted with 'MOD 2' in FIGURE 4), returned by the black-box model (represented by the 'Trained CNN' block in FIGURE 4).
2) We apply an ablation-based perturbation onto the MTS-to-image encoding by alternatively setting 2 out of 3 univariate signals to zero. This results in three perturbed MTS signals, named respectively $\tilde{x}$, $\tilde{y}$, $\tilde{z}$, each saving only the information of one attribute of the original MTS (in this respect, refer to the right handside of FIGURE 4).
3) We alternatively feed each perturbed signal to the black-box model by employing the classical Grad-CAM algorithm ('XAI 1'). This allows to extract three 2490 × 1 heatmaps $SM_{\tilde{x}}$, $SM_{\tilde{y}}$, $SM_{\tilde{z}}$ together with the prediction score $p_{\tilde{x}}, p_{\tilde{y}}, p_{\tilde{z}}$.
4) Finally, we aggregate the outputs of the previous steps (refer to the 'Recombination Rule' layer in the lower part of FIGURE 4), thus achieving a final heatmap $SM_{a_{2D-MTS}}$, according to the following recombination rule:

$$SM_{a_{2D-MTS_x}} = |p_I - p_{\tilde{x}}|\,(SM_I - SM_{\tilde{x}}) + SM_{\tilde{x}} \tag{3}$$

$$SM_{a_{2D-MTS_y}} = |p_I - p_{\tilde{y}}|\,(SM_I - SM_{\tilde{y}}) + SM_{\tilde{y}} \tag{4}$$

$$SM_{a_{2D-MTS_z}} = |p_I - p_{\tilde{z}}|\,(SM_I - SM_{\tilde{z}}) + SM_{\tilde{z}} \tag{5}$$

Instead of simply replicating the same heatmap SM_I on the three attributes of the multivariate input, the recombination rule allows to differentiate among the individual univariate contributions using the perturbed signals (blue ones or $SM_{\tilde{x}}$, $SM_{\tilde{y}}$, $SM_{\tilde{z}}$) and relying on the network prediction as a weight of relevance and on the linear combination of SM_I with $SM_{\tilde{x}}$, $SM_{\tilde{y}}$, $SM_{\tilde{z}}$.

Hence, $SM_{a_{2D-MTS}}$ is the concatenation of $SM_{a_{2D-MTS_x}}$, $SM_{a_{2D-MTS_y}}$ and $SM_{a_{2D-MTS_z}}$ to build a final 3×2049 heatmap.

Note that the classical Grad-CAM approach delivers heatmaps of the same dimension of the feature maps in the convolutional layer under consideration: so in our case the *Conv1D* layer with 1245 × 1 feature maps. Hence, in order to gain the 2490 × 1 heatmaps cited in steps 1 and 3, we duplicated each pixel intensity value of the calculated heatmap to match the largest dimension of the original input.

In this scheme, the contribution of the saliency map extracted using the original signal (orange ones or SM_I) is always retained in the linear combination as it represents the explanation of the full interaction among attributes, so the multivariate contribution.

To better understand this rationale, let us consider the following example: if the CNN prediction as a result of applying the perturbed input $\tilde{x}$ deviates considerably from p_I, so the value $|p_I - p_{\tilde{x}}|$ is high, which implies that all the information for the Grad-CAM is embedded in one or both of the ablated axes (set to zero). Therefore, the component x which did not get perturbed does not provide any relevant information about the correct class and, according to the recombination method, the corresponding heatmap $SM_{\tilde{x}}$ is weighed less.

Although this is presented as a task-specific solution, the recombination strategy can be potentially applied to all MTS-based CNNs to recover the importance information retained in each specific attribute, so offering a comprehensive and fine-graded explanation of the multivariate input.

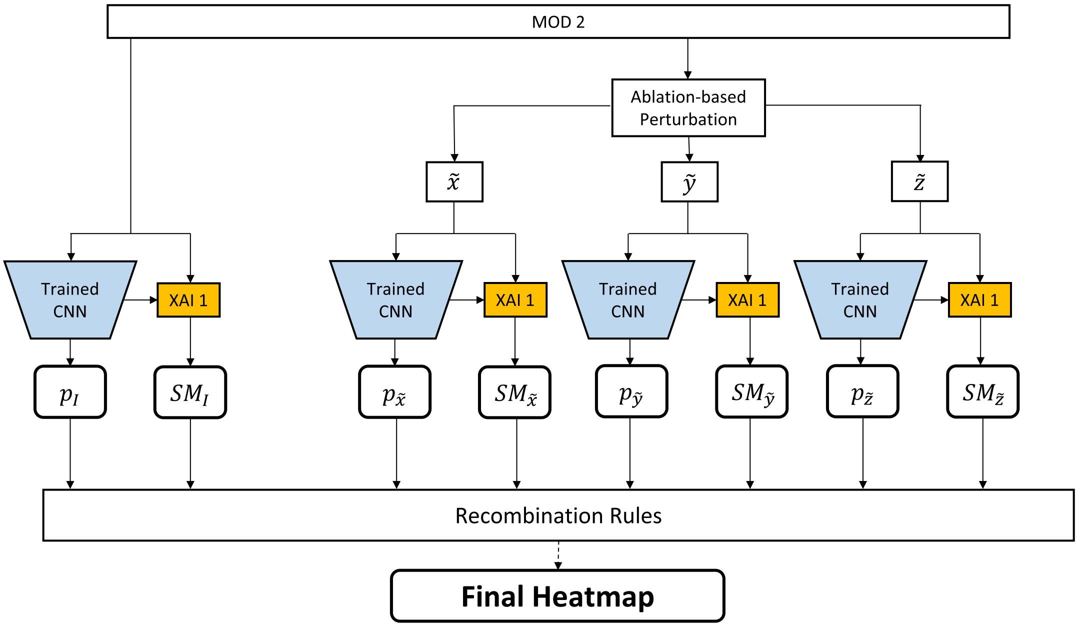


FIGURE 4. **Customized ablation-recombination methodology aimed at extracting a 2490 × 3 final heatmap $SM_{a_{2D-MTS}}$. Notation: MOD 2 acceleration signal, $\tilde{x}, \tilde{y}, \tilde{z}$ perturbed signals, $p_{\tilde{x}}, p_{\tilde{y}}, p_{\tilde{z}}$, prediction with perturbed signals, p_I prediction with original sample, SM_I explanation for original sample, $SM_{\tilde{x}}, SM_{\tilde{y}}, SM_{\tilde{z}}$ explanations for perturbed samples.**

2) INTEGRATED GRADIENTS

Integrated Gradients [32] is a path method measuring the changes in model prediction that result from changes to the intensity of input features. To this end, it realises a transformation between a baseline, on the one hand, – i.e., an object that generates no response through the network and that can be regarded as an instance of feature absence – and the input value, on the other hand. Such transformation is linear and incremental: namely, IG computes the variation in the model predictions with respect to the input feature at each step, averaging these incremental changes together. This method, too, was originally designed for images: IG exploits gradients with the aim of identifying, relative to the model's prediction function, which pixels (features) are responsible for strongly influencing the prediction at a given point. With respect to other gradient-based XAI methods, the intuition behind IG is that of accumulating the local gradients of each pixel and assigning their importance as a score that accounts for how much it adds or subtracts to the overall output class probability of the model. This aspect contributes to eventually avoiding the problem of gradient saturation. More formally, IG represents the path integral of the model gradient along a straight line path moving from the baseline to the input.

In our application, the image is our MTS-to-image encoding and the pixels, or features, are actually attribute values at specific time instants. We also would have the MTS-to-image encoding baseline denoted as a_{baseline}: it is a sample that does not provide any activation for the CNN and thus results in a nearly null posterior probability for the target class, i.e., the output of the model that has to be explained. This is an intrinsic challenge introduced by the considered task and a suitable approach for solving it is one of the main contributions of the paper. In detail, we proceed in the following way:

- we interpolate small steps along a straight line in the feature space, by considering an intensity step ε that varies between 0 (a baseline or starting point) and 1 (value of the input pixel);
- at each step we compute gradients between the model predictions and the current input;
- we approximate the integral between the selected baseline and the input by collecting (namely, according to cumulative average) the local gradients computed at the previous step.

This procedure can be formalised by the following IG-like mathematical notation:

$$\begin{aligned} &IG_i(a_{2D-MTS}) \\ &\quad = (a_{2D-MTS,i} - a_{baseline,i}) \\ &\qquad \times \int_{\varepsilon=0}^{1} \frac{\partial f(a_{baseline} + \varepsilon(a_{2D-MTS} - a_{baseline}))}{\partial a_{2D-MTS,i}} \, d\varepsilon \end{aligned} \quad (6)$$

where f represents the input-output function of the CNN to be explained, i the i-th feature (pixel) in the input and $\partial f(a_{2D-MTS})/\partial a_{2D-MTS,i}$ denotes the gradient of f along the i-th dimension of a_{2D-MTS}.

The Baseline Problem: A significant challenge of IG is the need to identify a baseline that is uninformative to the task. For most deep convolutional networks dealing with images, it is possible to choose a baseline such that the prediction at the baseline is near zero ($f(a_{baseline}) \approx 0$), e.g., a completely black image. Indeed, the baseline should convey a complete absence of signal, since when we assign attribution to a specific cause we implicitly consider the absence of the cause as a baseline for comparing outcomes [32]. In our case, a black image-like MTS would be a zero signal, which actually results informative for predicting the non-crash class, i.e. a low probability of a crash event. Thus, the aim is to find input signals where the feature values do not provide any activation in a CNN designed to work on time-series data. To this end, we design a mixed baseline which varies depending on the predicted class of the signal that we want to explain and we validate it with an exhaustive analysis.

For a signal predicted as crash event, we exploit as baseline the zero acceleration signal and the constant speed equal to 0.1 in a normalised scale. These two signals are supposed to be an input that does not produce activation for the crash class, as they model a vehicle moving at a constant very low speed. As a specular case, for a signal predicted as a non-crash event, we exploit as baseline the average of a subset of validation set signals predicted as a crash with a confidence larger than 0.95. This averaged high confidence crash input is supposed not to produce any activation in the network with respect to the non-crash class, so responding to the definition of baseline but only for the specific class we want to explain.

To analyse and prove the effectiveness of the proposed baselines, we perform an exhaustive analysis according to the following rationale. Given a specific class that we want to explain (either crash or non-crash), we consider the prediction of the AI agent for the corresponding baseline $p_{baseline}$ and for the original sample p_{input}: the more $p_{baseline}$ and p_{input} are different, ideally with $p_{baseline} \rightarrow 0$ and $p_{input} \rightarrow 1$, the more our baseline reflects the definition required by the IG method. Hence, for all inputs in the test set, we computed the difference $\delta = p_{input} - p_{baseline}$ between the prediction of the crash-alert system for the baseline ($p_{baseline}$) and for the current signal (p_{input}) and we provided a meaningful visualisation of the results on the test set (20, 293 samples) to comprehensively check if the rationale is respected. In other words, as we look for the probability of the target class being picked near 0 for the baseline ($p_{baseline} \sim 0$) and near 1 for the input signals ($p_{input} \sim 1$), we want large δ values for the instances correctly predicted by the CNN, encoded as a high difference of the posterior probability for input and baseline, i.e., high values of δ.

3) LIME

For completeness let us now briefly present another XAI method that we employed as a competitor. The Local Interpretable Model-agnostic Explanations (LIME) approach is suited to explain any black box since it does not inspect its internal parameters, but it relies on the intuition that the explanation can be locally derived from a surrogate model, starting from records that are randomly generated in the neighbourhood of the sample to be explained and are weighted according to their proximity to the sample itself [23]. We hereby consider its design for images, which has a straightforward application for working on our image-like MTS so it does not need any methodological adaptation. Given an image $I \in \mathbb{R}^d$, i.e., the original instance being explained, LIME uses the concept of super-pixels, i.e., regions or patches of similar pixels in the original sample, and defines $I' \in \{0, 1\}^d$ as a binary vector indicating the "presence" or "absence" of a super-pixel [23]. Thus, I' is equal to a vector containing only values equal to 1, with a length equal to the number of super-pixels captured in the image. In order to identify the super-pixels LIME exploits a segmentation algorithm.

To learn the local behaviour of the model f that we need to explain, LIME perturbs the vector I' with the aim of sampling new instances, named z', lying both in the proximity and far away from I'. This enables the creation of a binary dataset which is suitable for training the interpretable surrogate model. As a result, given a perturbed sample $z' \in \{0, 1\}^d$ that contains a fraction of the non-zero elements in I', we extract the corresponding segmented regions belonging to the original sample. This procedure creates a new image z exhibiting the original I pixel values in the segmented regions selected by z', and zero elsewhere. Then, we feed z to the black box, thus obtaining the prediction $f(z)$, that is, the posterior probability of a sample belonging to a class, which is subsequently used as a label for training the surrogate model together with z'. We repeat this process for all perturbed instances, and this way we create a dataset Z of perturbed samples together with the associated labels $(z', f(z))$.

Based on the surrogate model $g \in G$, where G is a class of potentially interpretable models (e.g., linear classifier, decision trees etc.), the explanation ξ_{LIME} is obtained according to the following:

$$\xi_{LIME}(I) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_{I'}) + \Omega(g) \tag{7}$$

with $\Omega(g)$ denoting a measure of complexity (by contrast with interpretability) of the explanation, $\mathcal{L}(f, g, \pi_{I'}) = \sum_{z,z' \in Z} \pi_{I'}(z)(f(z) - g(z'))^2$ a measure of how unfaithful g is at approximating f in the locality defined by $\pi_{I'}$, i.e., an exponential kernel attributing higher weights to instances similar to I' (if we consider the concept of *proximity*). Therefore, LIME is aimed at hitting a compromise between fidelity and interpretability, and, with the aim of ensuring both, it minimises $\mathcal{L}(f, g, \pi_{I'})$ while ascertaining that $\Omega(g)$ stays low enough to be interpretable by humans. Thus, even though the original model may be too complex for being explained globally, a locally faithful explanation (linear in this case) can still be gained.

C. QUANTITATIVE EVALUATION

With a fast-growing body of literature about XAI, the most important challenge lies in finding an evaluation methodology that is at the same time robust, cheap and effective. Moreover, there is still no consensus at all in the literature relative to the choice of the most appropriate method for evaluating explainability [34]. Doshi-Velez and Kim in [12] provide a first path toward the definition and rigorous evaluation of explainability, distinguishing between application-grounded evaluation, human-grounded evaluation, and functionally-grounded evaluation. In the first two approaches, the explanation is assessed by humans. The former involves domain experts/end users evaluating the XAI method on the specific task for which the machine learning model was trained, while the latter presents the explanation to ordinary people in a simple experimental task where the quality of the explanation is independent on the prediction accuracy of the learning model. Finally, the functionally-grounded evaluation does not require humans, resulting cheaper than human-level evaluation in terms of time, effort, and cost and involves a proxy task based on some formal definition of explainability to evaluate the quality of explanations [12].

Within this framework, a saliency maps-based explanation is feasible to our scope of increasing interpretability and user trust about a model working on MTS. This could be directly evaluated with a qualitative approach by visual inspection, however such an evaluation of the explanation is not suited to compare different XAI methods for three main issues: (i) it does not allow us to assess the effectiveness of the XAI methods on a large dataset; (ii) it does not evaluate if the XAI method is able to capture the temporal dependencies learned from the model; (iii) it relies on the subjectivity of the evaluator.

In this respect, here we adapted to MTS two perturbation-based strategies presented by Schlegel et al. [13] for UTS. The evaluation procedure can therefore be carried out as follows. First, we compute the explanation provided by each one of the XAI methods included in the comparison, i.e., Grad-CAM, IG and LIME. Second, we perform two types of perturbation to the input signal: (i) the *XAI perturbation*, that modifies the input time points identified by the XAI method as relevant; (ii) the *random perturbation*, which modifies random regions of the input signal. Third, we measure both the performance drop Δ of the AI agent for the samples given by the XAI perturbation (Δ_{XAI}), and by the random perturbation (Δ_{random}). This evaluation is based on the assumption that if relevant features (time points) get changed, the model performance should decrease more than in a scenario where only random regions undergo the modification.

Defining $\mathbf{M}$ as the original MTS sample, $\mathbf{M}_{XAI}$ and $\mathbf{M}_{random}$ as the samples having undergone a XAI-based and a random-based perturbation respectively, and using $F1$ score as quality metric for evaluating the CNN performance, the Δ drop can be computed as follows:

$$\Delta_{XAI} = F1_{\mathbf{M}} - F1_{\mathbf{M}_{XAI}}, \quad \Delta_{random} = F1_{\mathbf{M}} - F1_{\mathbf{M}_{random}} \tag{8}$$

If the XAI method returns correct explanations, we expect that:

$$\Delta_{XAI} > \Delta_{random}. \tag{9}$$

We perform two types of perturbation distinguished into two categories, namely *perturbation analysis* and *sequence evaluation*. The former locally exploits punctual perturbations to the signal pattern: in this respect, it defines a *zero* perturbation, that is, time series values corresponding to relevant regions are set to 0. The latter takes into account the signal trend over time by evaluating the importance of time series features such as slopes or minima defining *swap* and *mean* perturbations. The first one inverts the time ordering of the time series relevant/random values whereas the second one replaces them with the mean value over the selected window.

Finally, since we aim to carry out an exhaustive evaluation, we perform this procedure on all 20, 293 test set samples, using the trained CNN to obtain the predictions on the test data. Then, we retrieve an explanation with the selected XAI methods and, on the basis of the resulting time point relevance, test data are perturbed by the XAI and random perturbations. Each of the newly created test sets is therefore predicted by the model and the quality measure is evaluated for comparison purposes (eq. 8-9).

IV. EXPERIMENTS AND RESULTS

This section first presents the preliminary analysis of the dataset at hand (section IV-A), then section IV-B describes the specific settings employed to run the experiments, whereas the final results are reported and commented in section IV-C.

A. PRELIMINARY ANALYSIS

We conducted a preliminary analysis to study whether the two modalities (i.e. the MTS acceleration signal and the UTS speed signal) have the same importance in the model and are worthy to be contextually explained. To this end, we conducted an ablation test over the two modalities: we alternatively put to zero one of the inputs, and then we measured the drop in classification performance. As performance indicator of interest, we considered precision, as a metric that summarises both the contribution of the true positive (i.e., correctly recognised crash events) and false positive samples (i.e., non-crash instances recognised as crash events). A significant drop in this performance score would imply two possible and adverse scenarios: (i) the false positive samples have increased, so the AI agent would trigger a high number of emergency calls; (ii) the true positive samples have decreased, so the AI agent is missing true emergency situations.

To avoid any bias we considered the validation set for this analysis, finding that precision undergoes a 36.8% reduction when ablating the acceleration MTS, versus the almost absent reduction (1.5%) when ablating the UTS speed signal. This preliminary experiment suggested that the model considers the acceleration-based MTS as the most informative input for the task at hand. Consequently, we froze the speed signal, therefore ensuring that the explanations depend only on the acceleration.

B. EXPERIMENTAL SETUP

In this section we detail the experimental setup and the parameters chosen to extract the desired explanations and to evaluate them. We point out that the evaluation for the explainability uses the test dataset since we are interested in assessing explainability for new data with respect to the training set to provide insights on the internal representation learned from the network. The code is available at https://github.com/ltronchin/translating-xai-mts.git.

First, as stated in the previous section, we froze the speed signal by implementing a straightforward application of the Grad-CAM approach for MTS (described in section III-B). On the contrary, for IG and LIME we proceeded as follows: (i) to apply IG we only considered in the path integral the contribution of the gradient with respect to the varying acceleration MTS input, and consequently also the baseline; (ii) to apply LIME we did not perform any perturbation on the speed signal to ignore its contribution in training the surrogate model.

Regarding the XAI parameters, on the one hand Grad-CAM does not require any fine-tuning phase as it is directly applied to the last convolutional layer of the CNN. On the other hand, for IG and LIME we adopted the following settings.

1) IG

The integral of IG can be approximated via a Riemann summation: we sum the gradients at points occurring in sufficiently small intervals along the path from the baseline $a_{baseline}$ to the input a_{2D-MTS}. Following the recommendations reported in [32], we checked that the attributions approximately would add up to the difference between the score at the input and that at the baseline. According to this rationale, we chose 200 as the number of steps to faithfully approximate the IG integral.

Besides this, as reported in section III-B, the crucial point in applying IG is the baseline choice, which we inspect by adopting the visual representation shown in FIGURE 5. In the chart, each box depicts one test sample, where the green and red denote a hit or a miss, i.e., a sample that the model correctly or incorrectly recognises, respectively. The shapes represented in the boxes, i.e., × or ○, define the ground truth of the sample, crash and non-crash, respectively. Finally, the grey level in each cell describes the magnitude of the performance difference δ, from 0 (black) to 1 (white). Recalling the rationale for the baseline choice, we expect the difference δ between the prediction at the input (p_{input}) and at the baseline ($p_{baseline}$) to be high, ideally equal to 1. The full representation is available in the GitHub repository.

Hence, FIGURE 5 confirms the effectiveness of our mixed baseline approach since the difference δ between p_{input} and $p_{baseline}$ is 1 for the majority of the samples. Moreover, it is worth noticing that for the majority of wrongly classified samples (red ones) the grey-level intensity is lower than 1 (darker than those correctly classified): indeed, this is reasonable and expected since the AI agent is actually making a mistake, so it shows lower confidence in the prediction (p_{input}). From a preliminary analysis, we also observed that using the same baseline both for the crash and non-crash samples, i.e., considering a zero acceleration signal for the image-like MTS and a constant speed equal to 0.1 in a normalised scale for the velocity signal, results in uninformative SM for non-crash signals. These results are not reported here for the sake of brevity.

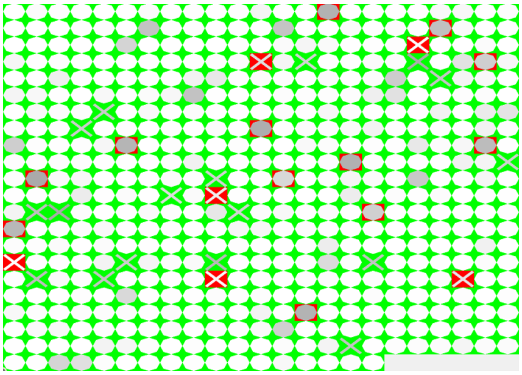


FIGURE 5. The figure shows the δ value for 500 randomly selected test samples.

2) LIME SETTINGS

LIME trains a transparent model to explain the black-box locally: hence, it requires a fine-tuning phase to fit the parameters and to create the perturbed training datasets of the surrogate model. As a segmentation function, we use the Felzenszwalb representation [35] that copes with the grayscale nature of the image-like MTS, and we fix other LIME's parameters coherently with the original implementation [23], except for the feature selection approach and the number of coefficients considered as important in the linear model. These two settings are worth studying since the former allows to select which crucial regions (superpixel) result in the best representation for the surrogate model to approximate the complex one, whereas the latter defines how many important regions are needed to assess the visual explainability on the considered sample. On the one hand, as feature selection strategy we tested both the highest weight and the forward selection. The former technique selects the highest features resulting from the product between the original data point and the absolute values of the coefficients learned from the linear model using all features. The latter iteratively adds features to the model evaluating the increase in R^2 score. On the other hand, to choose the number of coefficients considered as important we performed several experiments considering 5, 10, 15, 20, 25 among the highest features.

To evaluate the tuning procedure, we use the R^2 score and the Accordance Accuracy (AA) score as a measure of agreement between the CNN and linear model prediction on original samples. The former aims to estimate how well the linear regression fits the training data, whilst the latter aims to estimate how well the linear model locally resembles the complex model computing the formula:

$$AA = \frac{1}{n_{TEST}} \sum_{i=0}^{n_{TEST}-1} 1\,(g(a'_{2D-MTS,i}) = f(a_{2D-MTS,i})) \quad (10)$$

where $1(g(a'_{2D-MTS,i}) = f(a_{2D-MTS,i}))$ is the indicator function which is worth 1 when $g(a'_{2D-MTS,i}) = f(a_{2D-MTS,i})$, 0 otherwise, n_{TEST} indicates the number of test samples in the dataset and $a_{2D-MTS,i}$ and $a'_{2D-MTS,i}$ are the acceleration image-like MTS and the corresponding vector of superpixels, respectively. We find an R^2 and AA scores equal to 79% and 96% using forward selection and 94% and 99% using highest weight obtaining no difference by changing the number of considered parameters, i.e. from 5 to 25 with a step equal to 5. Thus, in order to investigate the interpretability ability of LIME heatmaps we visually inspect the explanation for crash and non-crash events finding that progressively incrementing the number of features does not increase the effectiveness of the explanation, achieving the opposite effect on the interpretability. FIGURE 6 clearly shows that as the number of selected regions rises, it is no longer possible to identify a clear pattern of features in the input sample that lead to the decision of interest. Summarising, we found that the best parameters that fit our application are: (i) the highest weights features selection procedure; (ii) the number of features extracted from the linear model equal to 5.

C. RESULTS

FIGURE 7 reports the saliency maps attained by the three XAI approaches on two instances of different classes as representative examples. The left column shows the explanations related to the crash sample, whereas the right column includes those related to the non-crash sample. Furthermore, the legend of the saliency maps colours is located in the top right corner of FIGURE 7 and it shows that we adopt two different colour coding. The first is applied to Grad-CAM and IG since they provide real valued importance scores: hence, they are represented by red shades which get darker for higher importance values. The second is used for LIME for which we adopt a binary importance map, where green areas indicate the relevant time-stamps for the given predicted class.

Regarding Grad-CAM saliency maps shown in the top row of FIGURE 7, we point out the following remarks. Considering the crash sample, the explanation highlights as important the signal trend following the crash event: after the shift in x acceleration values, there is a transition period that leads the car cinematic to a new stable state. On the contrary, saliency maps on y and z axes do not show significant activation, outlining two main findings: (i) the x component is the most significant UTS when it comes to predicting the event as a crash; (ii) y and z contributions are independent with respect to that of the x-axis. This finding is reasonable as, in general, the strongest fluctuations of the acceleration for a crash event occur along the vehicle's travel direction, i.e., the x-axis, proving the reliability of the explanations. For the non-crash example, we observe a less clear focus of the black-box model on a precise pattern. Grad-CAM heatmaps exhibit lower and more distributed intensity signals with respect to the crash sample saliency maps: long-term temporal relationships are detected as important for the non-crash prediction. Indeed, a bumpy road is characterised by less intense and longer disturbances with respect to those that emerge in a crash event. From FIGURE 7, we further notice that the Grad-CAM explanation presents spurious peaks in the heatmap along the three axes, especially for the crash class (activation on the y-axis). This effect is probably due to the ablation study described in section III-B. Indeed if, on the one hand, it allows us to determine which attribute of MTS is most important for the prediction, the recombination rule propagates the activation from one UTS to the others resulting in a saliency interval that does not solely depend on the explainability of the considered UTS. However, we alleviated this effect using the network prediction for re-weighting as stated in section III-B1.

Let us now turn the attention towards IG explanations, shown in the second row of FIGURE 7. In the case of the crash sample we notice that most of the information relevant for the prediction is in the x-axis, but in different time instants compared to those chosen by Grad-CAM: specifically the time instants of the negative peak and two small intervals immediately before it. For the y and z axes the results are similar to Grad-CAM confirming the x component as the most important UTS for predicting a crash. The considerations made for Grad-CAM explanations on the non-crash example also hold for IG: the saliency map shows almost a uniform colour with lower and distributed peaks highlighting the long-term temporal relationships. A final confirmation of these results is provided by LIME, shown in the last row of FIGURE 7. When we compare the results on the crash sample we observe that the relevant intervals comprehend time instants of the x-axis negative peak and two small intervals immediately before it. Less clear is the correspondence between the results on the other two axes or rather on the non-crash sample. This behaviour can be explained with the followings observations. First, LIME explanation does not present a clear evidence of the difference between each UTS, since it uses a surrogate model to approximate the CNN and it does not directly inspect the convolutional architecture inner workings. Second, IG and Grad-CAM offer a diverse kind of explanation than LIME: IG and Grad-CAM succeed in explaining the sample at time instant level, i.e., the importance score is defined for each point composing the signal; on the contrary, LIME is designed to explain the input sample at the region level, returning to the user the highest weight of the trained linear model. Third, since LIME relies on segmentation to define the regions to be perturbed over the image-like MTS, it captures the importance in a blended fashion: in fact, it explains the model considering at once both temporal (namely, the presence of slope and minima) and spatial features. As a result, the algorithm neglects the contribution of each UTS separately and fails to provide the importance of each component. This can be is observed from the LIME binary saliency map reported in FIGURE 7: each detected region is shared across the three axes with a few exceptions.

Overall, from a qualitative perspective we can state that IG and Grad-CAM are successfully able to exploit the cross-correlation in UTS learned from the CNN defining

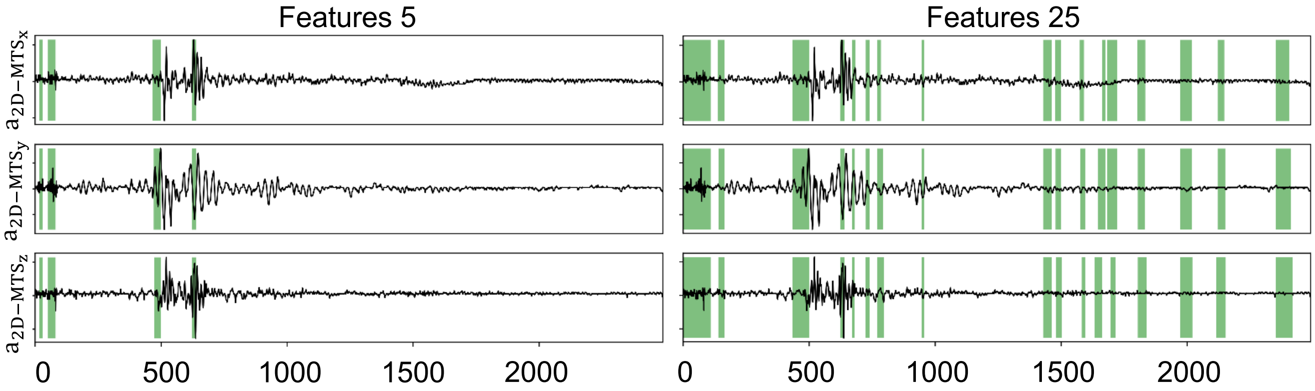


FIGURE 6. Visual evaluation of LIME fine-tuning. By increasing the number of valuable features, the explanation becomes less interpretable: as the number of selected regions rises, it is no longer possible to identify a pattern of features in the input sample that lead to the decision of interest.

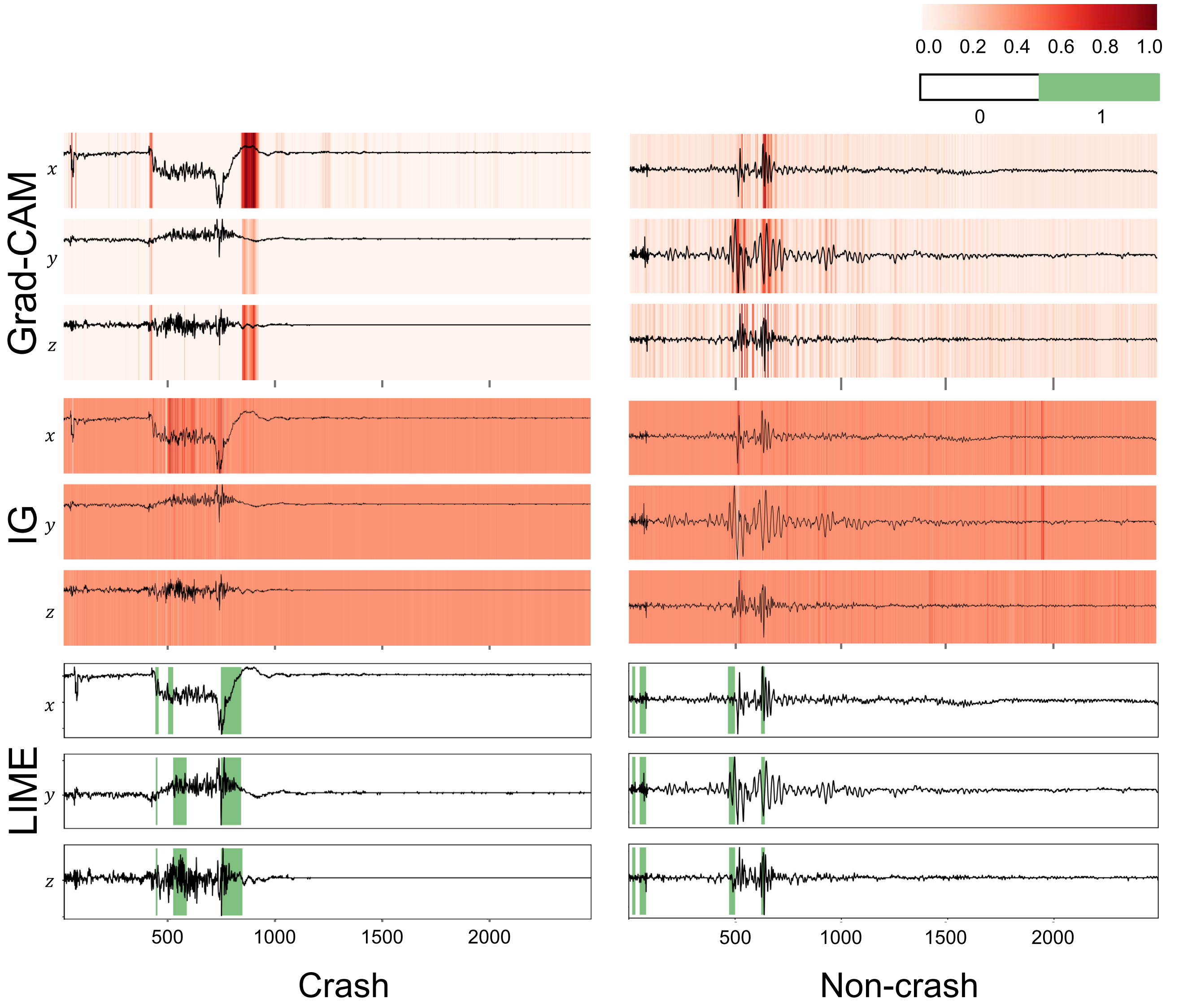


FIGURE 7. Explanation over the acceleration signal for a crash and a non-crash event using Grad-CAM, IG and LIME.

which UTS is most valuable for the prediction. The explanations result reasonable in finding the crucial features used from the model to perform the crash detection task, casting light on the black-box's decision process.

Turning the attention to the quantitative analysis, TABLE 2 shows the results emerging from the XAI evaluation described in section III-C. Results point out that IG-based perturbations in all cases exceed the random drop, yielding also the absolute highest drop in performance for the *zero* and *swap* perturbations (58.2% and 15.2%). Hence, the quantitative analysis suggests IG as the most informative explainability method among the competitors, for it is able to detect the time points that are the most valuable for the CNN to perform the prediction. Indeed, if the points identified as valuable by IG are perturbed, we always obtain a higher performance drop than in the random case. The opposite happens in the case of Grad-CAM: indeed, it is assessed as the least reliable algorithm for the considered task as two out of three times it is overtaken by random drop. As a further observation, we notice that all the *swap* and *mean* drop values are lower than the zero ones. This could imply that the temporal trend exerts less influence on the CNN when it comes to performing the classification task. We deem that the model mainly focuses on sudden or instantaneous changes in the acceleration values, i.e., the presence of spikes, since deleting them (*zero* XAI perturbation) results in an average performance drop of 44% by comparison with the average drops of 12% and 6% for swap and mean XAI perturbations, respectively.

TABLE 2. **Each box reports the performance drop per XAI method and perturbation type (Pert.). The results are in bold if $\Delta_{XAI} > \Delta_{random}$.**

Pert. XAI	Zero	Swap	Mean
Grad-CAM	**20.1 %**	6.5 %	2.7 %
IG	**58.2 %**	**15.2 %**	**5.5 %**
LIME	**54.3 %**	13.8 %	**10.0 %**
Random	0.7 %	14.3 %	3.6 %

Before discussing the external validation, a few remarks are in order. It is worth noting that given the temporal nature of the task it would indeed be possible to use a recursive architecture or networks based on its applications, but the choice to maintain a CNN approach was motivated by two main rationales: (i) for the purposes of explainability of network decisions, a CNN lends itself better to be interpreted with visualisation mechanisms such as those shown in the manuscript; indeed, recent studies have shown that XAI methods applied to networks generally used with Time Series, such as RNN, GRU and LSTM, are directly transferable into them only in some cases that are highly task-dependent [36]; hence, in order to favour the generalisability of the methods, we chose an approach that was more easily explainable even in possible different contexts; (ii) the performance that can be achieved with an LSTM network, even if implementing an Attention logic, is comparable with that of a CNN, by appropriately processing the input data [37]; however, the execution times are significantly lower with a CNN, thus making this choice the most adapt for real world deployment.

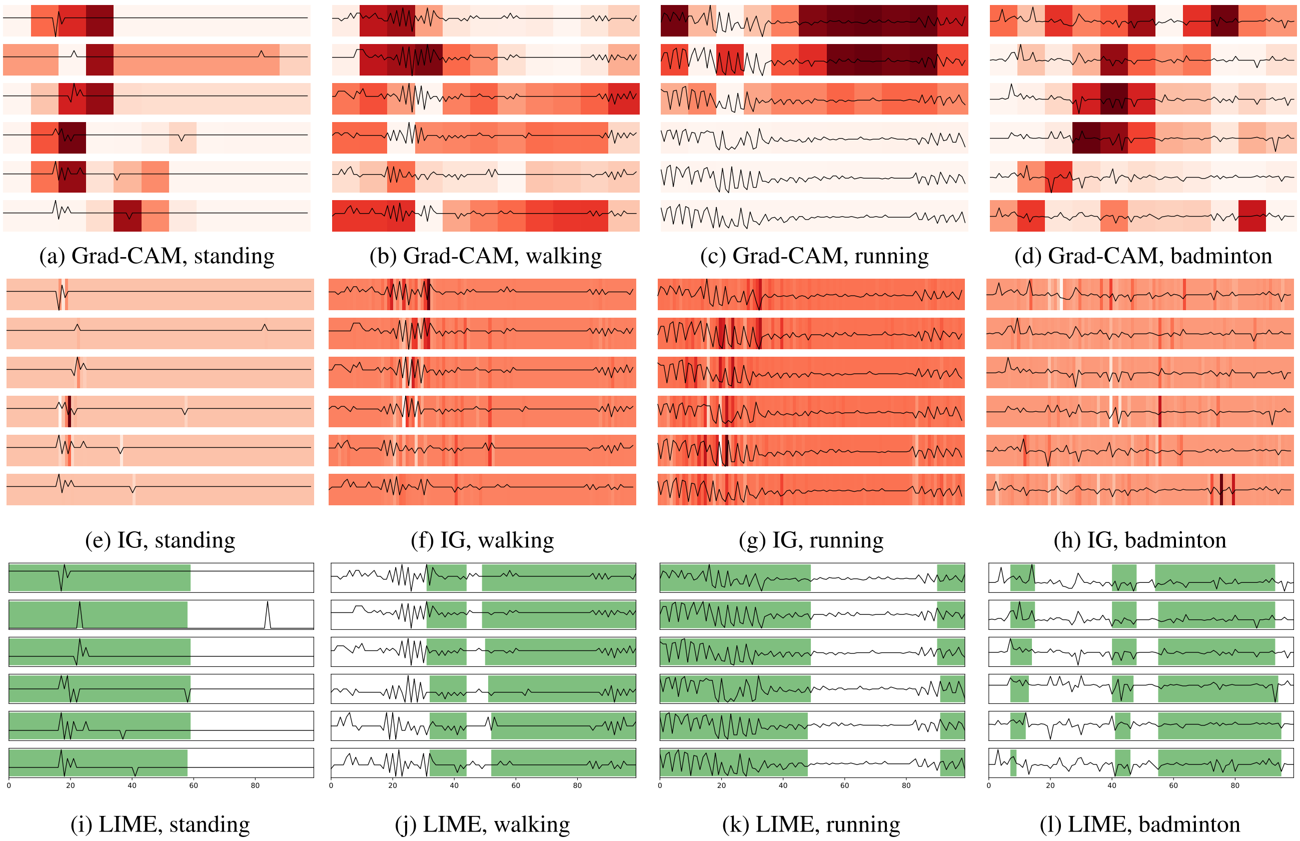
V. EXTERNAL VALIDATION

We validate the generalisability of our approach by including a different CNN, i.e., MTEX-CNN [3], applied to the BasicMotions dataset [38]. With reference to the network architecture, the MTEX-CNN is a state-of-the-art two-stage CNN network, so its design straightforwardly allows the application of our XAI methodology. In the first step, MTEX-CNN employs 2D convolution filters to extract features related to each attribute, whilst in the second step, it utilises 1D convolution filters to extract temporal information. It then processes the output feature map from the convolutional steps with fully connected layers for classification purposes. Interested readers can refer to [3] for a detailed description of the network architecture.

Turning our attention to the dataset, its use allows us to maintain the context of multivariate time series retrieved from motion sensors as accelerometers and gyroscopes. The BasicMotions is an MTS dataset from the UCR Time Series Classification Archive [38]; it comprises multivariate signals from 80 subjects acquired using the accelerometer and gyroscope of a smartwatch. For each acquisition, the data were sampled once every tenth of a second for ten seconds, resulting in a 100-length signal. The dataset contains data from people doing four activities: walking, resting, running, and badminton, which we use as classification labels in our analysis. We adhered to the original hold-out split of the dataset [38], resulting in 40 subjects for training and 40 for testing. We trained the MTEX-CNN on the training data for 100 epochs with a batch size of 32 and using the Adam optimiser with a learning rate of 0.001, using categorical cross-entropy as a loss function. We achieved a performance coherent with the state-of-the-art, with an accuracy of 92.5% and a F1-score of 92.1% on the test portion.

Given the trained model, we applied the customisation of Grad-CAM, IG and LIME to extract the SMs on the test set using the same experimental setup detailed in Section IV for the telematic dataset. We extract Grad-CAM SM using the last 1D convolutional layer of the MTEX-CNN architecture. For IG, we use the zero signal as a baseline for the accelerometer and gyroscope data. We use 500 steps to approximate the IG integral faithfully. Turning our attention to LIME, we use 5 as the maximum number of features for the explanation.

FIGURE 8 displays the SMs generated by the three XAI approaches for four different class instances: standing, walking, running, and badminton. We used the same colour map as in FIGURE 7. The first three signals in each plot represent accelerometer data, while the last three are from the gyroscope. From the first row of FIGURE 8 (Grad-CAM saliency maps), we observe that the network primarily focuses on the accelerometer signal for the walking and running classifications. For the standing class, the SM highlights the temporal periods following the movement-triggered shift



(a) Grad-CAM, standing (b) Grad-CAM, walking (c) Grad-CAM, running (d) Grad-CAM, badminton

(e) IG, standing (f) IG, walking (g) IG, running (h) IG, badminton

(i) LIME, standing (j) LIME, walking (k) LIME, running (l) LIME, badminton

FIGURE 8. Explanation over the MTS from basic motion dataset for standing, walking, running and badminton samples using Grad-CAM, IG and LIME.

from sitting to standing in each subject. It is reasonable for classification purposes, as the network looks for the absence of a signal post-trigger rather than the trigger itself, which, on the contrary, holds for the other classes. In the badminton class, the network considers the variations in the gyroscope signal, aligning with the classification of a less repetitive sports activity.

Moving to the second-row of FIGURE 8 we notice that IG saliency maps provide less information than Grad-CAMs. This could be due to the limited temporal length of the signals, which restricts IG from providing detailed point-wise information.

The last row of FIGURE 8 shows LIME's saliency maps. From these SMs, it is challenging to discern a clear distinction in importance between accelerometer and gyroscope signals. Thus, we encounter similar limitations for LIME as observed in our crash/non-crash application.

Overall, combining the three methods offers a comprehensive view of the temporal period in which the network predominantly focuses on performing the classification task. The insights from the Grad-CAM, IG, and LIME saliency maps provide an understanding of network reasoning, proving the effectiveness of our approach in enhancing the interpretability of CNN models working on MTS data.

VI. CONCLUSION

In this paper, we faced the challenge of explaining a deployed multimodal architecture applied to a multivariate time series real-world dataset. To this end we proposed the adaptation of two state-of-the-art XAI algorithms and investigated their effectiveness with a quantitative evaluation analysis.

The targeted problem consists in optimising the automatic assistance service by which an insurance company triggers a call to the EMS whenever a vehicle mounting a suitable black box is involved in an accident. Introducing a XAI layer into the considered application scenario aims at increasing the trustworthiness of the AI agent that performs the crash detection task and enhances the emergency team's effectiveness. Indeed, if the EMS were provided with complementary information about the crash event given by the explanation itself – e.g., the forces applied to the vehicle due to a collision exploiting the explanation heatmap – the EMS dispatching teams could accurately allocate their resources according to the severity of damages. As a result, prior knowledge of the severity of the accident not only has the potential to improve EMS teams' key performance indicator of time-to-the-scene, but it can also save critically scarce resources via data-driven precision dispatching.

More specifically, we showed how to adapt different methodologies, originally designed for images, to the challenge of explaining multivariate time series.

In our comparative analysis, we found that Integrated Gradients is the most effective approach, fitting with the complex nature of the data, and providing the best explanations.

In general, this study provides insight into the quality of explanation and sheds light on the most significant features that are exploited by CNN when it performs the crash detection task.

Although this work shows promising results in a rather challenging real-world scenario, some aspects can benefit from further investigation. As a first direction for future work, we plan to study XAI methods able to provide a more human-interpretable representation, since the saliency maps provided by all the XAI methods can be hard to interpret depending on the skills of the specific end-users. A second direction could focus on developing a XAI method able to explain both signals available in the telematics data at hand (i.e., acceleration and speed), fully exploiting the multimodal nature of the task at hand.

Also, whilst this study makes significant contributions to the field of explainable AI, it is important to acknowledge its limitations.

- *Dataset specificity*. The study focuses on specific datasets of multivariate time series data generated by motion sensors (i.e. accelerometers and gyroscopes). Although we show that our findings and methodologies generalise to another dataset, including data from the same sensors in a different domain, there is the need to further investigate other domains and different data sources.
- *Lack of human evaluation*. The study primarily relies on quantitative evaluation measures to assess the effectiveness of the XAI algorithms. Whilst these measures provide valuable insights, a more comprehensive evaluation involving human feedback and interpretation could enhance the understanding of the explanations' quality and usefulness.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence (bias) the work reported in this article.

AUTHOR CONTRIBUTIONS

Lorenzo Tronchin: conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing–original draft, writing-review and editing, and visualization. **Ermanno Cordelli**: conceptualization, methodology, visualization, and writing–review and editing. **Lorenzo Ricciardi Celsi**: writing–original draft and writing–review and editing. **Daniele Maccagnola**: writing–review and editing. **Massimo Natale**: writing–review and editing. **Paolo Soda**: conceptualization, methodology, writing–review and editing, supervision, project administration, and funding acquisition. **Rosa Sicilia**: conceptualization, methodology, writing–original draft writing–review and editing, supervision, and project administration.

REFERENCES

[1] N. Jin, Y. Zeng, K. Yan, and Z. Ji, "Multivariate air quality forecasting with nested long short term memory neural network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8514–8522, Dec. 2021, doi: 10.1109/TII.2021.3065425.

[2] Z. Ji, Y. Wang, K. Yan, X. Xie, Y. Xiang, and J. Huang, "A space-embedding strategy for anomaly detection in multivariate time series," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117892.

[3] R. Assaf and A. Schumann, "Explainable deep neural networks for multivariate time series predictions," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 6488–6490.

[4] P. Soda, R. Sicilia, L. Acciai, and G. Iannello, "Grasping inter-attribute and temporal variability in multivariate time series," *IEEE Trans. Big Data*, vol. 7, no. 5, pp. 885–892, Nov. 2021.

[5] T.-C. Fu, "A review on time series data mining," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 164–181, Feb. 2011, doi: 10.1016/j.engappai.2010.09.007.

[6] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Min. Knowl. Disc.*, vol. 33, pp. 917–963, Mar. 2019.

[7] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.

[8] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Aug. 2018.

[9] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, Dec. 2019.

[10] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2018, pp. 80–89.

[11] L. Tronchin, R. Sicilia, E. Cordelli, L. R. Celsi, D. Maccagnola, M. Natale, and P. Soda, "Explainable AI for car crash detection using multivariate time series," in *Proc. IEEE 20th Int. Conf. Cognit. Informat. Cognit. Comput.*, Oct. 2021, pp. 30–38.

[12] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608*.

[13] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of XAI methods on time series," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 4197–4201.

[14] A. H. Gee, D. Garcia-Olano, J. Ghosh, and D. Paydarfar, "Explaining deep classification of time-series data with learned prototypes," in *Proc. CEUR Workshop*, vol. 2429. NIH Public Access, 2019, p. 15.

[15] I. Karlsson, J. Rebane, P. Papapetrou, and A. Gionis, "Explainable time series tweaking via irreversible and reversible temporal transformations," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 207–216.

[16] E. Ates, B. Aksar, V. J. Leung, and A. K. Coskun, "Counterfactual explanations for multivariate time series," in *Proc. Int. Conf. Appl. Artif. Intell. (ICAPAI)*, May 2021, pp. 1–8.

[17] K. Fauvel, T. Lin, V. Masson, É. Fromont, and A. Termier, "XCM: An explainable convolutional neural network for multivariate time series classification," *Mathematics*, vol. 9, no. 23, p. 3137, 2021.

[18] S. A. Siddiqui, D. Mercier, M. Munir, A. Dengel, and S. Ahmed, "TSViz: Demystification of deep learning models for time-series analysis," *IEEE Access*, vol. 7, pp. 67027–67040, 2019.

[19] S. Cho, G. Lee, W. Chang, and J. Choi, "Interpretation of deep temporal representations by selective visualization of internally activated nodes," 2020, *arXiv:2004.12538*.

[20] R. Saluja, A. Malhi, S. Knapič, K. Främling, and C. Cavdar, "Towards a rigorous evaluation of explainability for multivariate time series," 2021, *arXiv:2104.04075*.

[21] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jun. 2009, pp. 947–956.

[22] S. Athey, J. Tibshirani, and S. Wager, "Generalized random forests," *Ann. Statist.*, vol. 47, no. 2, pp. 1148–1178, Apr. 2019.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[24] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[25] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *J. Syst. Eng. Electron.*, vol. 28, no. 1, pp. 162–169, Feb. 2017, doi: 10.21629/JSEE.2017.01.18.

[26] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 96–112, Feb. 2016.

[27] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Netw.*, vol. 116, pp. 237–245, Aug. 2019.

[28] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2017, pp. 6541–6549.

[29] F. van den Boom, "Regulating telematics insurance," in *AIDA Europe Research Series on Insurance Law and Regulation*. Cham, Switzerland: Springer, 2021, pp. 293–325.

[30] J. Debayle, N. Hatami, and Y. Gavet, "Classification of time-series images using deep convolutional neural networks," in *Proc. 10th Int. Conf. Mach. Vis. (ICMV)*, Apr. 2018, pp. 242–249.

[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[32] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3319–3328.

[33] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.

[34] C. Molnar, *Interpretable Machine Learning*. Morrisville, NC, USA: Lulu.com, 2020.

[35] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[36] W. Freeborough and T. van Zyl, "Investigating explainability methods in recurrent neural network architectures for financial time series data," *Appl. Sci.*, vol. 12, no. 3, p. 1427, Jan. 2022.

[37] H. Weytjens and J. De Weerdt, "Process outcome prediction: CNN vs. LSTM (with attention)," in *Proc. Bus. Process Manage. Workshops (BPM)*. Seville, Spain: Springer, Sep. 2020, pp. 321–333.

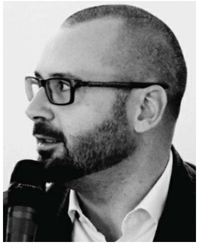
[38] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The UCR time series archive," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 6, pp. 1293–1305, Nov. 2019.

LORENZO TRONCHIN (Student Member, IEEE) received the degree (Hons.) in biomedical engineering from the University Campus Bio-Medico of Rome (UCBM), in 2020, where he is currently pursuing the Ph.D. degree in science and engineering for humans and the environment. His research activities focus on machine learning, explainable artificial intelligence, and generative models applied to multimodal data.

ERMANNO CORDELLI received the master's degree in biomedical engineering and the Ph.D. degree in biomedical engineering (computer science area) from the University Campus Bio-Medico of Rome (UCBM), in 2014 and 2017, respectively. After the master's degree, he worked for one year with a collaboration contract with the Computer Systems and Bioinformatics (CoSBi) Research Laboratory, Departmental Faculty of Engineering, UCBM, where he is currently an Assistant Professor with the CoSBi Laboratory. His main research topics are artificial intelligence and its applications in the health sector, federated learning, computer vision, radiomics, and the IoT, working within a project on the creation of an intelligent pen for diabetes treatment.

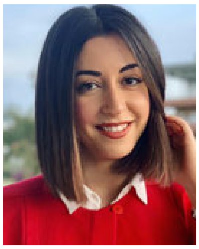
LORENZO RICCIARDI CELSI (Senior Member, IEEE) has been the Head of AI with Whoosnap, since 2022. He is responsible for the development of Insoore AI and property of Whoosnap. He received the Italian National Habilitation as an Associate Professor of automatic control, in 2023. Previously, he was the Manager with ELIS Innovation Hub, from 2018 to 2022; a Researcher with Sapienza University of Rome, in 2018; and a Contract Professor with eCampus University, from 2015 to 2018. His research interests include multi-agent systems and multimedia forensics applications of AI.

DANIELE MACCAGNOLA received the Ph.D. degree in computer science. He is currently a Data Scientist with Generali Italia S.p.A. Previously, he was a Lead Data Scientist with Clinical Insights Ltd., and a Research Fellow with Università di Milano-Bicocca and Brunel University London.



MASSIMO NATALE is the Head of data and platform governance with Generali Italia S.p.A. He is responsible for leading a team of 30 data scientists that will leverage data and AI/ML algorithms to drive business results. His passion for exploring new ways of human–machine communication drives him to find different and more effective AI/ML solutions and integrate data science applications into business processes. He was with large companies and start-ups, and brings a broad set of experience in design and management of innovative ICT projects. He was the author of more than 20 scientific papers published on international peer-reviewed journals, patents, and conference presentations.

PAOLO SODA (Member, IEEE) received the degree (Hons.) in biomedical engineering and the Ph.D. degree in biomedical engineering (computer science area) from the University Campus Bio-Medico of Rome (UCBM), in 2004 and 2008, respectively. Currently, he is a Full Professor of computer science and computer engineering with UCBM. His research interests include artificial intelligence, pattern recognition, machine learning, big data analytics, and data mining applied to data, signal, 2-D and 3-D image and video processing, and analysis.



ROSA SICILIA (Member, IEEE) was born in Cosenza, Italy, in 1993. She received the degree in biomedical engineering and the Ph.D. degree in biomedical engineering (computer science area) from the University Campus Bio-Medico (UCBM), Rome, in 2016 and 2020, respectively. After one year as a Postdoctoral Researcher, she is currently an Assistant Professor (RTDA) with UCBM, a position co-funded by Regione Lazio to work on the project "We-ease-it: A Smart and Intelligent Outpatient Clinic for Hospital 4.0." Her main research interests include machine learning and multimodal data mining.

Open Access funding provided by 'Università ''Campus Bio-Medico'' di Roma' within the CRUI CARE Agreement